XD/XL series PLC

User manual［Hardware］

WUXI XINJE ELECTRIC CO., LTD.

No.PD01 20181203 3.5

XINJE

XD/XL series PLC

User manual［hardware］

General descriptions

- Thank you for purchasing Xinje XD/XL series PLC.
- This manual mainly introduces XD/XL series PLC hardware features etc.
- Please read this manual carefully before using and wire after understanding the content.
- About software and programming instructions, please refer to related manuals.
- Please hand this manual over to operation users.

Notices for users

- Only experienced operator can wire the plc. If any problem, please contact our technical department.
- The listed examples are used to help users to understand, so it may not act.
- Please conform that PLC specifications and principles are suitable when connect PLC to other products.
- Please conform safety of PLC and machines by yourself when use the PLC. Machines may be damaged by PLC errors.

Responsibility state

- The manual content has been checked carefully, however, mistakes may happen.
- We often check the manual and will correct the problems in subsequent version. Welcome to offer advices to us.
- Excuse us that we will not inform you if manual is changed.

Contact information

If you have any problem about products, please contact the agent or Xinje company.

- Tel: 0086 510-85134136 85123803
- Fax: 0086 510-85111290
- Address: Building 7 fourth floor, No.100, Dicui Rd, Wuxi, China.
- Code : 214072

Safety notes

Please read this part carefully before using and operate after understanding the usage, safety and notices. Pay attention to safety and wire correctly.

We have summarized possible problems that may happen and classify them by warning and caution. About other matters, please operate in basic working order.

⚠ Caution	Incorrect use may lead to danger, such as moderate and slight injury, property loss.
⚠ Warning	Critical miss may lead to serious danger, such as death or serious injury, serious loss of property.

- **Conform about products**

⚠ Caution
Do not install the controller which is damaged, lack parts or type unfit. Otherwise, injury may occur.

- **Product design**

⚠ Warning
Please make safety circuit outside controller to make sure the system can run in safety when controller errors. Otherwise, incorrect action or fault may occur.

⚠ Caution
Do not put control wiring or power wiring together, separate them at least 10cm in principle. Otherwise, incorrect action or damage may occur.

- **Product installation**

⚠ Warning
Cut off all external power before installing controller. Otherwise, an electric shock may occur.

Caution
1. Please install and use the PLC in the environment condition that specified in general specifications in this manual. Do not use in wet, high temperature, smog, conductive dust, corrosive gas, combustible gas, vibration, shock occasion. Otherwise, electric shock, fire disaster, incorrect action, damage etc. 2. Do not touch conductive parts of PLC. Otherwise, incorrect action or fault may occur. 3. Please install the product by DIN46277 or M3screw and install them on flat surface. Otherwise, incorrect action or damage may occur. 4. Avoid ablation powder or clastic wires into product shell when processing screw holes. Otherwise, incorrect action or fault may occur. 5. Make sure connection compact and good when using expansion cables to connect expansion modules. Otherwise, bad communication or incorrect action may occur. 6. Cut off power when connecting external devices, expansion devices and battery etc. Otherwise, incorrect action or default may occur.

- **Product wiring**

Warning
1. Cut off external power before wiring. Otherwise, an electric shock may occur. 2. Connect AC or DC power to special power terminal correctly. Otherwise, may burn the controller. 3. Close the panel cover plate before controller powering on and running. Otherwise, an electric shock may occur.

Caution
1. Do not connect external 24V power to controllers' or expansion modules' 24V and 0V terminals , products damage may occur. 2. Use 2mm^2 cable to ground the ground terminals of expansion modules and controllers, never common ground to high voltage system. Otherwise, products fault or damage may occur. 3. Do not wiring between idle terminals. Otherwise, incorrect action or damage may occur. 4. Avoid ablation powder or clastic wires into product shell when processing screw holes. Otherwise, incorrect action or fault may occur.

5. Tighten up wiring terminals and separate conductive parts. Otherwise, incorrect action or product damage may occur.

- **Run and maintenance**

Warning
1. Do not touch terminals after power on. Otherwise, an electric shock may occur. 2. Do not connect or move the wires when power on. Otherwise, an electric shock may occur. 3. Make sure to stop the PLC before changing the controller program. Otherwise, malfunction may occur.

Caution
1. Do not disassemble and assemble product arbitrarily. Damage to product may occur. 2. Plug and connect cables on the condition of power off. Otherwise, cable damage or malfunction may occur. 3. Do not wire the idle terminals. Otherwise, malfunction or damage may occur. 4. Cut off the power when disassemble expansion modules, external devices and batteries. Otherwise, malfunction and fault may occur. 5. Dispose them as industrial waste when out of use.

Catalog

Preface

We will introduce constitution of content, application, convention, relevant manuals and how to get data in this part.

Content Components

This manual includes XD/XL series PLC types and system constitutions. It mainly introduces XD/XL series PLC basic units' specification, I/O wiring, run and maintenance, and XD/XL series PLC expansion modules' parameters, appearance and features etc.

This manual has 9 chapters, an overview of each chapter are as follows:

1．Summary

This chapter mainly introduces XD/XL series PLC specifications, types and descriptions.

2. Specifications

This chapter mainly introduces XD/XL series PLC basic units' common specifications, performance specifications, terminal placement, product dimensions, interface descriptions etc.

3. System constitutions

This chapter mainly introduces XD/XL series PLC system constitutions, peripheral devices, expansion devices, CPU and expansion devices connection principles, products installation, I/O point calculation, I/O address number distribution etc.

4．Power specifications and wiring

This chapter mainly introduces XD/XL series PLC power specifications, wiring methods.

5．Input specifications and wiring

This chapter mainly introduces XD/XL series PLC input specifications, input wiring, high speed counting etc.

6．Output specifications and wiring

This chapter mainly introduces XD/XL series PLC output specifications, relay output and transistor output etc.

7．Run, debug, maintenance

This chapter mainly introduces XD/XL series PLC run, debug steps, daily maintenance etc.

8．Expansion devices

This chapter mainly introduces I/O expansion modules, analog temperature modules' specifications, dimensions and terminal placements.

9．Switch between soft elements

This chapter mainly introduces XD/XL series PLC special function that free switch between input and output points.

Appendix 1．Special soft elements schedule

This chapter mainly introduces XD/XL series PLC special function soft elements, registers and expansion module address distribution etc.

Appendix 2．Instruction schedule

This chapter mainly introduces basic instructions, application instructions and special instructions that XD/XL series PLC support.

Appendix 3．PLC function configuration schedule

This chapter mainly introduces XD/XL series PLC main function of each type for lectotype.

Appendix 4．Common questions A&Q

This chapter mainly introduces XD/XL series PLC problems and solutions that may occur when using.

Manual scope of application

This manual is hardware manual of XD/XL series PLC, contents are as follows:

1．XD series PLC basic units

XD1-16R/T-E
XD1-32R/T-E

XD2-16R/T/RT-E/C
XD2-24R/T/RT-E/C
XD2-32R/T/RT-E/C
XD2-48R/T/RT-E/C
XD2-60R/T/RT-E/C

XD3-24R/T/RT-E/C, XD3-24PR/T/RT-E/C
XD3-32R/T/RT-E/C, XD3-32PR/T/RT-E/C
XD3-48R/T/RT-E/C, XD3-48PT-E/C
XD3-60R/T/RT-E/C, XD3-60PT-E/C

XD5-16R/T-E/C
XD5-24R/T/RT-E/C, XD5-24T4-E/C
XD5-32R/T/RT-E/C, XD5-32T4-E/C
XD5-48R/T/RT-E/C
XD5-60R/T/RT-E/C
XD5-48T4-E/C
XD5-48T6-E/C

XD5-60T4-E/C
XD5-60T6-E/C
XD5-60T10-E/C

XDM-24T4-E/C, XDM-24PT4-E/C
XDM-32T4-E/C, XDM-32PT4-E/C
XDM-60T4-E/C, XDM-60T4L-E/C
XDM-60T10-E/C, XDM-60PT10-E/C

XDC-24T-E/C
XDC-32T-E/C
XDC-48T-E/C
XDC-60T-E/C

XD5E-30T4-E
XD5E-60T10-E

XDME-60T10-E

2． XD series PLC expansion modules

- I/O expansion

XD-E8X8YR, XD-E8PX8YR, XD-E8X8YT, XD-E8PX8YT, XD-E16X, XD-E16PX, XD-E16YR, XD-E16YT, XD-E16X16YR, XD-E16PX16YR, XD-E16X16YT, XD-E16PX16YT, XD-E32X, XD-E32PX, XD-E32YR, XD-E32YT

- Analog expansion modules

AD: XD-E4AD, XD-E8AD, XD-E8AD-A, XD-E8AD-V
DA: XD-E2DA, XD-E4DA
AD/DA: XD-E4AD2DA, XD-E4AD2DA-B

- Temperature measurement

XD-E6PT-P, XD-E6TC-P, XD-E2TC-P

- Pressure meansurement

XD-E1WT-A, XD-E2WT-A, XD-E4WT-A
XD-E2WT-B
XD-E1WT-C, XD-E2WT-C, XD-E4WT-C

3. XD series expansion board

- XD series expansion BD board

XD-NE-BD, XD-NO-BD, XD-NS-BD

- XD series left expansion ED board

XD-WBOX-ED, XD-SBOXT-ED, XD-4GBOX-ED, XD-NES-ED
XD-4AD-A-ED, XD-4AD-V-ED
XD-4DA-A-ED, XD-4DA-V-ED
XD-2AD2DA-A-ED, XD-2AD2DA-V-ED

XD-2AD2PT-A-ED, XD-2AD2PT-V-ED
XD-2PT2DA-A-ED, XD-2PT2DA-V-ED

4．XL series PLC basic units
XL3-16T, XL3-16R, XL1-16T
XL3-16T, XL3-16R, XL3-16PR
XL5-32T4
XL5E-32T4
XLME-32T4

5．XL series PLC expansion module

- I/O expansion
  XL-E8X8YR, XL-E8X8YT
  XL-E16X
  XL-E16YR, XL-E16YT
  XL-E16X16YT, XL-E32X, XL-E32YT

- Analog expansion
  XL-E4AD2DA, XL-E8AD-A, XL-E8AD-V, XL-E4DA, XL-E4PT3-P, XL-E4TC-P

6．XL series ED expansion module

- XL communication expansion ED module
  XL-NES-ED
- XL analog expansion ED module
  XL-2AD2DA-A-ED, XL-2AD2DA-V-ED
  XL-2AD2PT-A-ED, XL-2AD2PT-V-ED
  XL-2PT2DA-A-ED, XL-2PT2DA-V-ED
  XL-4AD-A-ED, XL-4AD-V-ED
  XL-4DA-A-ED, XL-4DA-V-ED

7．XL power supply module
XL-P50-E

Manual conventions

We use some short names to replace the original names in the manual. The possible names have been listed in the table below to compare.

Short name	Explanation
XC series PLC	General name of XC series programmable logic controllers

XL series PLC	General name of XL series programmable logic controllers
XD series PLC	General name of XD series programmable logic controllers
Basic units or noumenon	Short name of XD series PLC basic units
Expansion devices or expansion units	General name of XD series PLC expansion modules and BD cards
Expansion modules	General name of XD series PLC all expansion modules.
Input and output expansion or I/O expansion	Short name of XD series PLC all input and output expansion modules
Analog expansions	Short name of XD series PLC all analog expansion modules
Peripheral units	General name of programming software, HMI and network modules
Programming software	General name of XD series PLC programming software XDPPro
HMI	General name of TG, TH, TP, OP, MP series products
TG series	General name of TG series touch screen
TH series	General name of TH series touch screen
TP series	General name of TP series touch screen
OP series	General name of OP series text panel
MP series	General name of MP series touch display

Relevant manual

This manual includes XD/XL series PLC hardware, about more application such as programming and instructions, please refer to relevant manuals.

Manual name	Manual introduction	Notes
Installation manual		
XD/XL series PLC installation manual	Descript XD/XL series basic units' specification, dimensions, installation, wiring etc.	Electronic version Need additional request
Programming software		
XD/XL series PLC users' manual【software】	Introduce XD/XL series PLC software XDPPro usage and skill etc.	Electronic version Need additional request
Instruction programming manual		
XD/XL series PLC users' manual【instructions】	Introduce XD/XL series PLC basic instructions, application instructions, communication, PID, C language,	Electronic version Need additional

	BLOCK etc.	request
Expansion manual		
XD/XL series analog temperature expansion manual	Introduce XD/XL series analog, temperature expansion module feature, parameters, ID, dimension, terminals and wiring etc.	Electronic version need additional request
X-NET manual		
X-NET fieldbus communication manual	Introduce X-NET fieldbus using method	Electronic version need additional request

Manual Acquisition

Users can get manual above in the following ways:

1. Paper manual
   Please ask product vendor, agent or agency to supply.

2. Electronic version
   Please ask product vendor, agent or agency to supply CD.

1 Summary of XD/XL Series PLC

XD/XL series PLC have diverse CPU units and expansions with powerful functions. In this chapter, we mainly introduce the XD/XL series PLC performance, program summary and product different parts.

1-1. Product Specifications

1-1-1. XD series CPU units

1 Models

XD series PLC CPU unit have rich product types.

- I/O Points 16, 24, 30, 32, 48, 60 points
- Output Type transistor, relay, transistor and relay mixed.
- Input Type PNP, NPN
- Power Type AC220V, DC24V

Series	Description
XD1(economic type)	Include 16, 32 points. cannot support right expansion module, left expansion ED module, expansion BD.
XD2(basic)	Include 16, 24, 32, 48, 60 points. cannot support right expansion module, can connect left expansion ED module, expansion BD (except 16 points model).
XD3(standard)	Include 16, 24, 32, 48, 60 points. Can connect expansion module, ED module, expansion BD (except 16 points model).
XD5(enhanced)	Include 16, 24, 32, 48, 60 points. With all the XD3 functions, the speed is 12 times of XC series, larger capacity. Support 2~6 axes pulse output, can connect expansion module, ED and BD.
XDM (motion control)	Include 24, 32, 48, 60 points. With all the XD3 functions, support 4~10 axes high speed pulse output, support 2-axis linkage motion, interpolation, follow-cutting, can connect expansion module, ED and BD.
XDC (motion fieldbus)	With all the functions of XD3. Support 2~4 axes pulse output, 20-axis fieldbus motion control, special model supports 6-axis fieldbus motion control (4~6 axes interpolation), can connect expansion module, ED, BD.
XDE (Ethernet model)	Include 30 points model. With all the functions of XD3. Support Ethernet communication, support 4-axis high speed pulse output, support 2-axis

	linkage motion, interpolation, follow-cutting, can connect expansion module, ED and BD.
XDME(motion control, Ethernet)	Contains 60 points functions. It is compatible with most functions of XDM, supports Ethernet communication, supports motion control commands such as interpolation and servo, supports 10-axis high-speed pulse output, connects expansion module, expands ED and expands BD.

※1: About non-cpu function of products, please refer to appendix 3.

2 Powerful functions

XD series PLC have rich basic functions and many special functions. Different type is fit for different application.

Abundant basic function

- **High speed operation**
  Basic processing instruction: 0.02~0.05us. Scanning time: 10,000 per 1ms. Program capacity is up to 384KB.
- **Abundant expansions**
  The CPU units support 10~16 different expansion modules and 1~2 expansion boards, 1 left expansion ED module.
- **Multiple communication ports**
  CPU units have 1~4 communication ports, support RS232, RS485, and can work with many external devices, such as frequency inverters, instruments, printers.
- **Abundant software capacity**
  Up to 1024 processes S, 128 retention processes HS, 8000 intermediate relays M, 960 retention relays HM, 1280 input relays X, 1280 output relays Y, 576 normal timers T, 96 latched timers HT, 576 counters C, 96 retention counters HC, 8000 data registers D, 1000 retention data registers HD, 6144 registers FD.
- **Two programming types**
  XD series PLC support two programming types, instruction list and ladder chart which can switch to each other.
- **Rich instructions**
  Include order control, data move and compare, arithmetic, data circulate and shift, pulse output, HSC, interruption, PID etc.

- **Real time clock**
  XD series PLC has built-in clock to control time.
- **Compact size, convenient to install**
  XD series PLC has DIN and screw two installation modes.

Enhanced special function

- **X-NET fieldbus**
  XD2, XD3, XD5, XDM, XDE series PLC support X-NET fieldbus, which can fast communicate with XD series PLC and TG/TN series HMI. XDC series PLC supports X-NET fieldbus function, can control 20 motors at the same time. Refer to X-NET fieldbus manual for details.
- **Ethernet Communication**
  Ethernet PLC has RJ45 port and supports TCP/IP protocol. It can realize MODBUS-TCP communication and free format communication based on Ethernet. Supports program download, online monitoring, remote monitoring, and communication with other TCP/IP devices.
- **High-speed pulse counter, frequency up to 80KHz**
  XD series PLC CPU units have 2~10 channels two-phase high-speed counter and high-speed counting comparer, can realize single-phase and AB-phase counting, frequency up to 80 KHz.
- **High-speed pulse output, frequency up to 100 KHz.**
  XD series PLC[※1] usually have 2~10 pulse output terminals, pulse frequency up to 100KHz.
- **Interruption function**
  XD series PLC interruption functions include external interruption, timing interruption and high-speed counting interruption to meet different interruption demands.
- **I/O points switch freely**
  XD series PLC unique function. Do not need to change program when terminals are damaged.
- **C language function block**
  C language block makes the program more secured. C language rich operation function can realize many functions, which saves internal space and improves programming efficiency.
- **PID function on CPU units**
  XD series PLC[※1] CPU units have PID control function and auto-tuning control function.
- **Sequence BLOCK**
  Sequence block makes instructions carry out in sequence, especially suitable for pulse output, motion control, module read and write etc, and largely simplifys the program writing.
- **100 segments high speed counting interruption**

XD series PLC※1 high speed counter have 100 segments 32 bits preset value. Each segment can generate interruption with good real-time, high reliability, low cost.

- **PWM(pulse width modulation)**
  XD series PLC※1 PWM function can be used to control DC motor.
- **Frequency measure**
  XD series PLC※1 can measure frequency.
- **Precise time**
  XD series PLC※1can realize 1ms and 32bit precise timing.

※1: Here XD series PLC means the PLC that can realize the related function, not all XD series can realize the all above functions. Please refer to appendix 3 about PLC specific functions.

※2: PLC can output 100KHz to 200KHz high speed pulse, but cannot ensure all the servo can work well. Please connect 500Ω resistor between output terminal and 24V power supply.

3 Easy to program

XD/E series also use XDPPro program software. Improved aspects:

- Ladder and instruction can be switched at any time.
- Add Software annotation, ladder annotation, instruction hints etc.
- Offer many editing panel of special instructions.
- Perfect monitor modes: ladder monitor, free monitor, data monitor.
- Mutely-windows display, convenient to manage.

※1: More about XDPPro application, please refer to XD series PLC user manual (software).

1-1-2. XL series CPU units

1 Models

XL series ultra-thin PLC, the basic unit has one sub-series product.

- I/O Points 16 points, 32 points
- Output Type transistor, relay
- Input Type NPN, PNP
- Power Type DC24V

Series	Description
XL1(economic type)	Contains 16 points. Compatible with all functions of XD1 series PLC, the speed is 12 times faster than XC series. It does not support special functions such as pulse output, high-speed counting, X-NET field bus, right expansion module and left expansion ED module, and can meet the simple use needs of users.
XL3(basic)	Include 16 points. With all the functions of XD3 series PLC, the processing speed is 12 times of XC series PLC. Support right expansion module and left expansion ED module.
XL5(enhanced)	Contains 32-point. Compatible with all functions of XD5 series PLC, the speed is 12 times that of XC series, supporting four pulse output, supporting right expansion module and left expansion ED module, which can meet the needs of most users.
XL5E(Ethernet)	Contains 32-point. Compatible with all functions of XD5 series PLC, the speed is 12 times faster than XC series. It supports Ethernet communication, 4-channel pulse output, right expansion module and left expansion ED module. It can meet the needs of most users.
XLME(motion control, Ethernet)	Contains 32-point. Compatible with all functions of XDM series PLC, the speed is 12 times faster than XC series. It supports Ethernet communication, motion control instructions such as interpolation and servo, 4-channel pulse output, right expansion module and left expansion ED module. It can meet the needs of most users.

2 Powerful functions

XL series PLC have rich basic functions and many special functions.

Abundant basic function

- **High speed operation**
  Basic processing instruction: 0.02~0.05us. Scanning time: 10,000 per 1ms. Program capacity is up to 256KB.

- **Abundant expansions**
  The CPU units support 10 different right expansion modules and 1 left expansion ED module.
- **Multiple communication ports**
  CPU units have 1~3 communication ports, support RS232, RS485, and can work with many external devices, such as frequency inverters, instruments, printers.
- **Abundant software capacity**
  Up to 1024 processes S, 128 retention processes HS, 8000 intermediate relays M, 960 retention relays HM, 1280 input relays X, 1280 output relays Y, 576 normal timers T, 96 latched timers HT, 576 counters C, 96 retention counters HC, 8000 data registers D, 1000 retention data registers HD, 5120 registers FD.
- **Two programming types**
  XL series PLC support two programming types, instruction list and ladder chart which can switch to each other.
- **Rich instructions**
  Include order control, data move and compare, arithmetic, data circulate and shift, pulse output, HSC, interruption, PID etc.
- **Real time clock**
  XL series PLC has built-in clock to control time.
- **Compact size, convenient to install**
  XL series PLC has mini size and is easy to install on the DIN rail.

Enhanced special function

- **X-NET fieldbus**
  XL series PLC support X-NET fieldbus, which can fast communicate with XD/XL series PLC and TG/TN series HMI. Refer to X-NET fieldbus manual for details.
- **Ethernet Communication**
  Ethernet PLC has RJ45 port and supports TCP/IP protocol. It can realize MODBUS-TCP communication and free format communication based on Ethernet. Support program download, on-line monitoring, remote monitoring, and communication with other TCP/IP devices. Specific applications can be referred to "TCP/IP Communication User Manual Based on Ethernet Communication".
- **High-speed pulse counter, frequency up to 80KHz**
  XL series PLC CPU units have 3 channels two-phase high-speed counter and high-speed counting comparer, can realize single-phase and AB-phase counting, frequency up to 80 KHz.
- **High-speed pulse output, frequency up to 100 KHz.**
  XL series PLC[※1] usually have 2 pulse output terminals, pulse frequency up to 100KHz.
- **Interruption function**

XL series PLC interruption functions include external interruption, timing interruption and high-speed counting interruption to meet different interruption demands.

- **I/O points switch freely**
  XL series PLC unique function. Do not need to change program when terminals are damaged.
- **C language function block**
  C language block makes the program more secured. C language rich operation function can realize many functions, which saves internal space and improves programming efficiency.
- **PID function on CPU units**
  XL series PLC CPU units have PID control function and auto-tuning control function.
- **Sequence BLOCK**
  Sequence block makes instructions carry out in sequence, especially suitable for pulse output, motion control, module read and write etc, and largely simplifys the program writing.
- **100 segments high speed counting interruption**
  XL series PLC high speed counter have 100 segments 32 bits preset value. Each segment can generate interruption with good real-time, high reliability, low cost.
- **PWM(pulse width modulation)**
  XL series PLC PWM function can be used to control DC motor.
- **Frequency measure**
  XL series PLC can measure frequency.
- **Precise time**
  XL series PLC can realize 1ms and 32 bits precise timing.

3 Easy to program

XL series PLC also use XDPPro program software.

1-1-3. XD Expansions

1 Expansion Modules

To meet control requirement better, XD series PLC can work with expansions, XD1, XD2 cannot connect expansion modules, and XD3 can link 10 expansion modules, XD5, XDM, XDC, XD5E, XDME can connect 16 modules.

- Diverse types: I/O module, analog module.
- Compact size
- DC24V power

I/O module	Analog module	Temperature control
Power : DC24V Input points: 8-32 Output points: 8-32 Output type: Transistor Relay	Power: DC24V Type: DA, AD AD/DA DA channel No.: 2-4 AD channel No.: 4-8	Power: DC24V Input: PT100 thermocouple Channel: 6 PID control: built-in relay

2 Expansion BD

XD series can connect expansion BD board, 24~32 points can connect 1 BD, 48~60 points type can connect 2 BD boards. (16 points cannot connect BD)

- RS485 communication BD: X-NET interface, filedbus communication function, XD-NE-BD
- Optical fiber BD: X-NET optical fiber interface, filedbus communication function, XD-NO-BD
- RS232 communication BD: XD-NS-BD

3 Expansion ED

XD series left expansion ED board is for wireless communication. It can connect 1 ED board.

- Wifi communication ED: XD-WBOX-ED, support PLC program upload and download, remote monitoring.
- Wireless transparent transmission ED: XD-SBOXT-ED, support communication between PLC, HMI, PC.
- 4GBOX communication module: XD-4GBOX-ED, support remote wireless monitoring, PLC program upload and download, mobile phone message exchange, support 4G network.
- Communication expansion module: XD-NES-ED, support RS232 or RS485 (high-speed, support X-NET fieldbus), the two ports cannot use at the same time.
- Analog I/O:
  XD-2AD2DA-A-ED, support current I/O

XD-2AD2DA-V-ED, support voltage I/O
XD-4AD-A-ED, support current input
XD-4AD-V-ED, support voltage input
XD-4DA-A-ED, support current output
XD-4DA-V-ED, support voltage output

- Analog and temperature mixed type:

XD-2AD2PT-A-ED, support 2 channels current input, 2 channels PT100 temperature input.
XD-2AD2PT-V-ED, support 2 channels voltage input, 2 channels PT100 temperature input.
XD-2PT2DA-A-ED, support 2 channels PT100 temperature input, 2 channels current output.
XD-2PT2DA-V-ED, support 2 channels PT100 temperature input, 2 channels voltage output.

1-1-4. XL Expansions

1 Expansion Modules

To meet control requirement better, XL series PLC can work with expansions, XL3 can link 10 expansion modules, XL5/XL5E/XLME can link 16 expansion modules, XL1 cannot support expansion modules.

- Diverse types: I/O module, analog module.
- Compact size
- DC24V power

I/O module

Power : DC24V
Input points: 8~32
Output points: 8~32
Output type: Transistor
Relay

Analog module

Power: DC24V
DA channel No.: 2-4
AD channel No.: 4-8
Analog type: current
voltage

Temperature control module

Power: DC24V
Temperature input: PT100
Thermocouple
Temperature channel: 4
PID control: built-in
relay

2 Expansion ED

XL series PLC can connect one ED module on the left side.

- Communication expansion module: XL-NES-ED, support RS232 or RS485 (high-speed, support X-NET fieldbus), the two ports cannot use at the same time.
- Analog I/O:
  XL-2AD2DA-A-ED, support current I/O
  XL-2AD2DA-V-ED, support voltage I/O
  XL-4AD-A-ED, support current input
  XL-4AD-V-ED, support voltage input
  XL-4DA-A-ED, support current output
  XL-4DA-V-ED, support voltage output
- Analog and temperature mixed type:
  XL-2AD2PT-A-ED, support 2 channels current input, 2 channels PT100 temperature input.
  XL-2AD2PT-V-ED, support 2 channels voltage input, 2 channels PT100 temperature input.
  XL-2PT2DA-A-ED, support 2 channels PT100 temperature input, 2 channels current output.
  XL-2PT2DA-V-ED, support 2 channels PT100 temperature input, 2 channels voltage output.

1-2. Model list

1-2-1. XD series basic unit model and list

1 Basic unit model

XD series PLC basic unit model constitute:

XD□—○○□□○—□

① ② ③ ④ ⑤ ⑥ ⑦

1	Series name	XD
2	Series type	1: XD1 series economic type 2: XD2 series basic type 3: XD3 sereis standard type 5: XD5 series enhanced type M: XDM series motion control type C: XDC series X-NET motion fieldbus control type E: XDE Ethernet communication type
3	I/O points	16: 8 input/ 8 output 24: 14 input/ 10 output 30: 18 input/12 output 32: 18 input/ 14 output 48: 28 input/ 20 output 60: 36 input/ 24 output
4	Input point type	Nothing: NPN type P: PNP type
5	Output point type	R: Relay output T: Transistor output RT: Relay/Transistor mixed
6	Pulse channels	Nothing: item 5 is T/RT means 2 pulse channels 4: 4 channels 6: 6 channels 10: 10 channels
7	Power supply	E: AC power supply (220V) C: DC power supply (24V)

2 Basic unit model list

XD1 series List

Type							Input points (DC24V)	Output points (R, T)
AC power				DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	XD1-16R-E	XD1-16T-E	-	-	-	-	8	8
	XD1-32R-E	XD1-32T-E	-	-	-	-	16	16

XD2 series List

Type							**Input points (DC24V)**	**Output points (R, T)**
AC power				DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	XD2-16R-E	XD2-16T-E	XD2-16RT-E	XD2-16R-C	XD2-16T-C	XD2-16RT-C	8	8
	XD2-24R-E	XD2-24T-E	XD2-24RT-E	XD2-24R-C	XD2-24T-C	XD2-24RT-C	14	10
	XD2-32R-E	XD2-32T-E	XD2-32RT-E	XD2-32R-C	XD2-32T-C	XD2-32RT-C	18	14
	XD2-48R-E	XD2-48T-E	XD2-48RT-E	XD2-48R-C	XD2-48T-C	XD2-48RT-C	28	20
	XD2-60R-E	XD2-60T-E	XD2-60RT-E	XD2-60R-C	XD2-60T-C	XD2-60RT-C	36	24

XD3 series List

Type							**Input points (DC24V)**	**Output points (R, T)**
AC power				DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	XD3-16R-E	XD3-16T-E	XD3-16RT-E	XD3-16R-C	XD3-16T-C	XD3-16RT-C	8	8
	XD3-24R-E	XD3-24T-E	XD3-24RT-E	XD3-24R-C	XD3-24T-C	XD3-24RT-C	14	10
	XD3-32R-E	XD3-32T-E	XD3-32RT-E	XD3-32R-C	XD3-32T-C	XD3-32RT-C	18	14
	XD3-48R-E	XD3-48T-E	XD3-48RT-E	XD3-48R-C	XD3-48T-C	XD3-48RT-C	28	20
	XD3-60R-E	XD3-60T-E	XD3-60RT-E	XD3-60R-C	XD3-60T-C	XD3-60RT-C	36	24
PNP	XD3-16PR-E	XD3-16PT-E	-	XD3-16PR-C	XD3-16PT-C	-	8	8
	XD3-24PR-E	XD3-24PT-E	XD3-24PRT-E	XD3-24PR-C	XD3-24PT-C	XD3-24PRT-C	14	10
	XD3-32PR-E	XD3-32PT-E	XD3-32PRT-E	XD3-32PR-C	XD3-32PT-C	XD3-32PRT-C	18	14
	XD3-48PR-E	XD3-48PT-E	XD3-48PRT-E	XD3-32PR-C	XD3-32PT-C	XD3-32PRT-C	28	20
	XD3-60PR-E	XD3-60PT-E	XD3-60PRT-E	XD3-48PR-C	XD3-48PT-C	XD3-48PRT-C	36	24

XD5 series list

Type							Input points（DC24V）	Output points（R, T）
	AC power			DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN 型	XD5-16R-E	XD5-16T-E	-	XD5-16R-C	XD5-16T-C	-	8	8
	XD5-24R-E	XD5-24T-E	XD5-24RT-E	XD5-24R-C	XD5-24T-C	XD5-24RT-C	14	10
	-	XD5-24T4-E	-	-	XD5-24T4-C	-	14	10
	XD5-32R-E	XD5-32T-E	XD5-32RT-E	XD5-32R-C	XD5-32T-C	XD5-32RT-C	18	14
	-	XD5-32T4-E	-	-	XD5-32T4-C	-	18	14
	XD5-48R-E	XD5-48T-E	XD5-48RT-E	XD5-48R-C	XD5-48T-C	XD5-48RT-C	28	20
	-	XD5-48T4-E	-	-	XD5-48T4-C	-	28	20
	-	XD5-48T6-E	-	-	XD5-48T6-C	-	28	20
	XD5-60R-E	XD5-60T-E	XD5-60RT-E	XD5-60R-C	XD5-60T-C	XD5-60RT-C	36	24
	-	XD5-60T4-E	-	-	XD5-60T4-C	-	36	24
	-	XD5-60T6-E	-	-	XD5-60T6-C	-	36	24
	-	XD5-60T10-E	-	-	XD5-60T10-C	-	36	24
PNP 型	XD5-24PR-E	XD5-24PT-E	XD5-24PRT-E	XD5-24PR-C	XD5-24PT-C	XD5-24PRT-C	14	10
	XD5-32PR-E	XD5-32PT-E	XD5-32PRT-E	XD5-32PR-C	XD5-32PT-C	XD5-32PRT-C	18	14
	XD5-48PR-E	XD5-48PT-E	XD5-48PRT-E	XD5-48PR-C	XD5-48PT-C	XD5-48PRT-C	28	20
	XD5-60PR-E	XD5-60PT-E	XD5-60PRT-E	XD5-60PR-C	XD5-60PT-C	XD5-60PRT-C	36	24
	-	XD5-48PT6-E	-	-	XD5-48PT6-C	-	28	20

XDM series list

Type							Input points (DC24V)	Output points (R, T)
	AC power supply			DC power supply				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	-	XDM-24T4-E	-	-	XDM-24T4-C	-	14	10
	-	XDM-32T4-E	-	-	XDM-32T4-C	-	18	14
	-	XDM-60T4-E	-	-	XDM-60T4-C	-	36	24
	-	XDM-60T10-E	-	-	XDM-60T10-C	-	36	24
	-	XDM-60T4L-E	-	-	XDM-60T4L-C	-	36	24
PNP	-	XDM-24PT4-E	-	-	XDM-24PT4-C	-	14	10
	-	XDM-32PT4-E	-	-	XDM-32PT4-C	-	18	14
	-	XDM-60PT10-E	-	-	XDM-60PT10-C	-	36	24

XDC series list

Type							Input points (DC24V)	Output points (R, T)
	AC power			DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		

NPN	-	XDC-24T-E	-	-	XDC-24T-C	-	14	10
	-	XDC-32T-E	-	-	XDC-32T-C	-	18	14
	-	XDC-48T-E	-	-	XDC-48T-C	-	28	20
	-	XDC-60T-E	-	-	XDC-60T-C	-	36	24
	-	XDC-60C4-E	-	-	XDC-60C4-C	-	36	24
	-	XDC-60C6-E	-	-	XDC-60C6-C	-	36	24

XD5E series list

Type							**Input points (DC24V)**	**Output points (R, T)**
AC power				DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	-	XDE-30T4-E	-	-	-	-	18	12
	-	XD5E-60T10-E	-	-	-	-	36	24

XDME series list

Type							**Input points (DC24V)**	**Output points (R, T)**
AC power				DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	-	XDME-60T10-E	-	-	-	-	36	24

1-2-2. XL series basic unit model and list

1 Basic unit model

XL series PLC basic unit model constitute:

XL □ □ — ○○ □ □ ○

① ② ③ ④ ⑤ ⑥ ⑦

①: series	XL: XL series ultra-thin PLC
②: type	1: XL1 economic type
	3: XL3 series standard type
	5: XL5 enhanced type
	M: XLM series motion control type
③: Ethernet	E: Ethernet
	-: normal
④: I/O points	16: 8 input /8 output

	32：16 input /16 output
⑤：input type	-：NPN
	P：PNP
⑥：output type	T：transistor output
	R: relay output
⑦：pulse output channel	-: output type is T, 2 channels
	4：4 channels

2 Basic unit model list

XL3 series List

Type							Input points（DC24V）	Output points（R, T）
	AC power			DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	-	-	-	XL3-16R	XL3-16T	-	8	8

Type							Input points（DC24V）	Output points（R, T）
	AC power			DC power				
	Relay output	Transistor output	Relay/transistor mixed	Relay output	Transistor output	Relay/transistor mixed		
NPN	-	-	-	-	XL1-16T	-	8	8
	-	-	-	XL3-16R	XL3-16T	-	8	8
	-	-	-	-	XL5-32T4	-	16	16
	-	-	-	-	XL5E-32T4	-	16	16
	-	-	-	-	XLME-32T4	-	16	16
PNP	-	-	-	XL3-16PR	-	-	8	8

1-2-3. XD expansion module list

1 I/O expansion

I/O expansion modules name constitute:

XD — E ○ □ ○ □ — □

1 2 3 4 5 6 7

1	Series name	XD

2	Expansion module	E
3	Input points	8 or 16 or 32
4	Special for input	When input is NPN: X When input is PNP: PX
5	Output points	8 or 16 or 32
6	Output mode	YR: relay output YT: transistor output
7	Power supply type	E: AC220V C: DC24V

- I/O expansion module type list

Model				I/O points	Input points （DC24V）	Output points （R，T）
type	Input	Output				
		Relay output	Transistor output			
NPN	XD-E8X	-	-	8	8	-
	-	XD-E8YR	XD-E8YT	8	-	8
	-	XD-E8X8YR	XD-E8X8YT	16	8	8
	XD-E16X	-	-	16	16	-
		XD-E16YR	XD-E16YT	16	-	16
	-	XD-E16X16YR-E	XD-E16X16YT-E	32	16	16
	-	XD-E16X16YR-C	XD-E16X16YT-C	32	16	16
	XD-E32X-E	-	-	32	32	-
	XD-E32X-C	-	-	32	32	-
	-	XD-E32YR-E	XD-E32YT-E	32	-	32
	-	XD-E32YR-C	XD-E32YT-C	32	-	32
PNP	XD-E8PX	-	-	8	8 点	-
	-	XD-E8PX8YR	XD-E8PX8YT	16	8 点	8
	XD-E16PX	-	-	16	16 点	-
	-	XD-E16PX16YR-E	XD-E16PX16YT-E	32	16 点	16
	-	XD-E16PX16YR-C	XD-E16PX16YT-C	32	16 点	16
	XD-E32PX-E	-	-	32	32 点	-
	XD-E32PX-C	-	-	32	32 点	-

2 Analog temperature modules

Analog, temperature model constitute:

XD－E 4AD 2DA 6PT 6TC 1WT – P

① ② ③ ④ ⑤ ⑥ ⑦

1	Expansion module	E
2	Analog input	4AD: 4 channels analog input 8AD: 8 channels analog input
3	Analog output	2DA: 2 channels analog output
4, 5	Temperature input	6PT: 6 channels PT100 sensor input 6TC: 6 channels thermocouple sensor input
6	Pressure measurement	1WT: 1 channel pressure measurement 2WT: 2 channels pressure measurement 4WT: 4 channels pressure measurement
7	Type	P: PID control A: hardware is new version B: analog voltage output -5~5V or -10~10V C: hardware difference (only for WT model) V: input is voltage type

Analog, temperature expansion module type schedule

Type		**Function**
Analog input	XD-E4AD	4 channels analog input
	XD-E8AD	8 channels analog input
	XD-E8AD-A	8 channels analog input, current input type
	XD-E8AD-V	8 channels analog input, voltage input type
Analog input and output	XD-E4AD2DA	4 channels analog input, 2 channels analog output
	XD-E4AD2DA-B	4 channels analog input, 2 channels analog output
Analog output	XD-E2DA	2 channels analog output
	XD-E4DA	4 channels analog output
Temperature measurement	XD-E6PT-P	6 channels PT100 temperature measurement, with PID control
	XD-E6TC-P	6 channels K-type thermocouple temperature measurement, with PID control
Pressure measurement	XD-E1WT-A	1 channel pressure measurement, -39.06mV～39.06mV
	XD-E2WT-A	2 channels pressure measurement, -39.06mV～39.06mV
	XD-E4WT-A	4 channels pressure measurement, -39.06mV～39.06mV
	XD-E2WT-B	2 channels pressure measurement, 0～10mV
	XD-E1WT-C	1 channels pressure measurement, 0～10mV
	XD-E2WT-C	2 channels pressure measurement, 0～10mV
	XD-E4WT-C	4 channels pressure measurement, 0～10mV

1-2-4. XL expansion module list

1 I/O expansion

I/O expansion modules name constitute:

XL－E ○ □ ○ □
1 2 3 4 5 6

1: Series XL series expansion module
2: Expansion module E: expansion module
3: Input points 8 or 16 or 32
4: Input type X: NPN type input
PX: PNP type input
5: Output points 8 or 16 or 32
6: Output mode YT: transistor output
YR: relay output

- I/O expansion module type list

Model				I/O points	Input points (DC24V)	Output points (R, T)
type	Input	Output				
		Relay output	Transistor output			
NPN	-	XL-E8X8YR	XL-E8X8YT	16	8	8
	XL-E16X	-	-	16	16	-
		XL-E16YR	XL-E16YT	16	-	16
	-		XL-E16X16YT	32	16	16
	XL-E32X	-	-	32	32	-
	-	-	XL-E32YT	32	-	32

2 Analog expansion modules

Analog model constitute:

XL－E ○ □ ○ □－□
1 2 3 4 5 6 7

1: Series XL series expansion module
2: Expansion module E: expansion module
3: Input channel 2 or 4 or 8
4: Analog input AD: analog voltage, current input

5： Output channel 2 or 4

6： Analog output DA: analog voltage, current output

7： Analog type A: current mode

V: voltage mode

P: PID function

Analog expansion module type list

Type		Description
Analog I/O	XL-E4AD2DA	4 channels analog input, 2 channels analog output
	XL-E4DA	4 channels analog output, current/voltage mode
	XL-E8AD-A	8 channels analog input, current mode
	XL-E8AD-V	8 channels analog input, voltage mode
Temperature control	XL-E4PT3-P	4 channels PT100 temperature measuring, built-in PID function
	XL-E4TC-P	4 channels themocouple temperature measuring, built-in PID function

3 Left expansion ED module

Analog module model constitute:

XL－2AD 2DA 2PT NES－A－ED

1 2 3 4 5 6

1： Analog input 2AD：2 channels analog input

2： Analog output 2DA：2 channels analog output

3： Temperature measurement 2PT：2 channels PT100 input

4： Communication NES：RS232 or RS458 communication

5： Analog type A：I/O is current mode

V：I/O is voltage mode

6： Left expansion ED：left expansion ED module

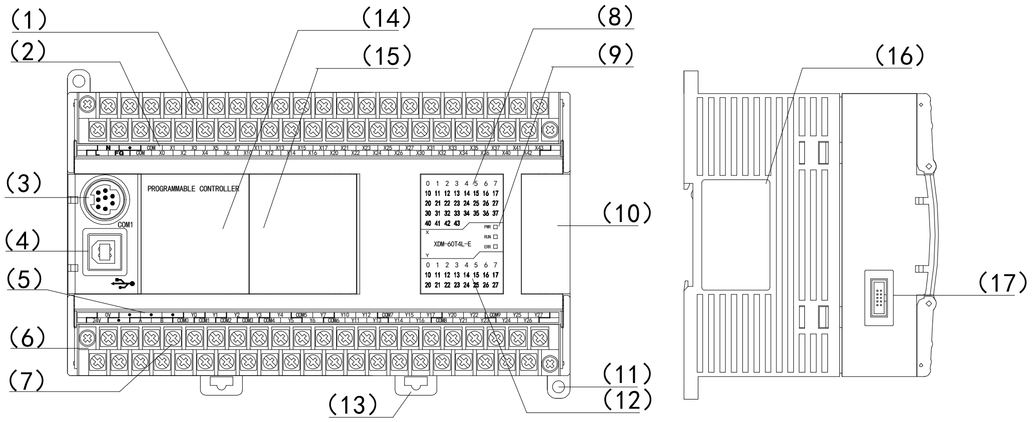
Left expansion ED module list:

Model		Description
Analog input	XL-E4AD-A-ED	4 channels analog current input
	XL-E4AD-V-ED	4 channels analog voltage input
Analog output	XL-E4DA-A-ED	4 channels analog current output
	XL-E4DA-V-ED	4 channels analog voltage output
Analog I/O	XL-E2AD2DA-A-ED	2 channels analog current input, 2 channels analog current output
	XL-E2AD2DA-V-ED	2 channels analog voltage input, 2 channels analog voltage output
Analog	XL-E2AD2PT-A-ED	2 channels analog current input, 2 channels PT100 temperature input

temperature mixed type	XL-E2AD2PT-V-ED	2 channels analog voltage input, 2 channels PT100 temperature input
	XL-E2PT2DA-A-ED	2 channels PT100 temperature input, 2 channels analog current output
	XL-E2PT2DA-V-ED	2 channels PT100 temperature input, 2 channels analog voltage output
Communication	XL-NES-ED	One RS232 port, one RS485 port, cannot use at the same time

1-3. Each Part's Description

1 XD series structure



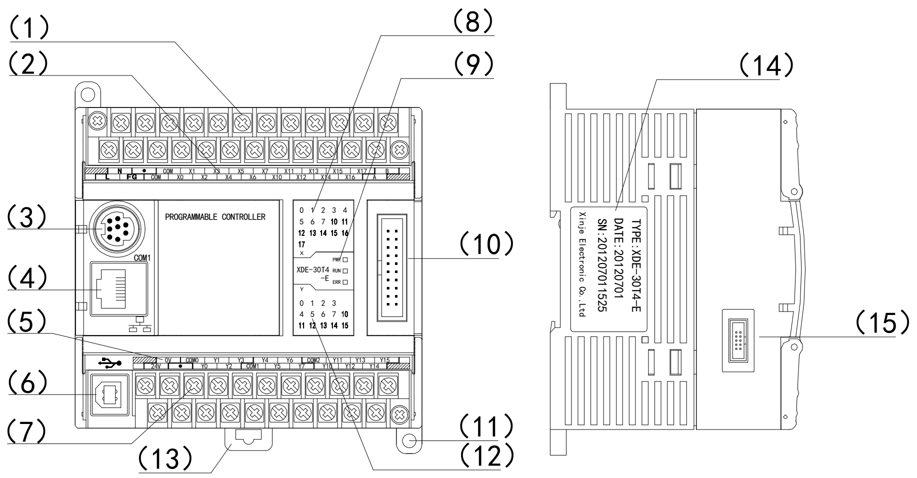
Each part's name is listed below:

① :Input & power supply terminals
② :Input terminal label
③ :COM1
④ :USB port
⑤ :Output terminal label
⑥ :Output & 24V power terminals
⑦ :output terminal, RS485 port(COM2)
⑧ :Input action display
⑨ :system LED
PWR: power supply
RUN: working
ERR: error

⑩ : expansion module connection port
⑪ : installation hole (2 holes)
⑫ : output action display
⑬ : rail mounting hook (2 hooks)
⑭ : expansion BD (COM4)
⑮ : expansion BD (COM5)
⑯ : product label
⑰ : expansion ED (COM3)

Note: (1) for the PLC hardware version below 3.2, position 4 is RS232 port.
(2) for XD1, XD2, XDC series PLC, position 4 is RS232 port.
(3) for XDC series PLC, position 4 RS232 port and terminal A and B (RS485 port) is the same port, they cannot be used at the same time.

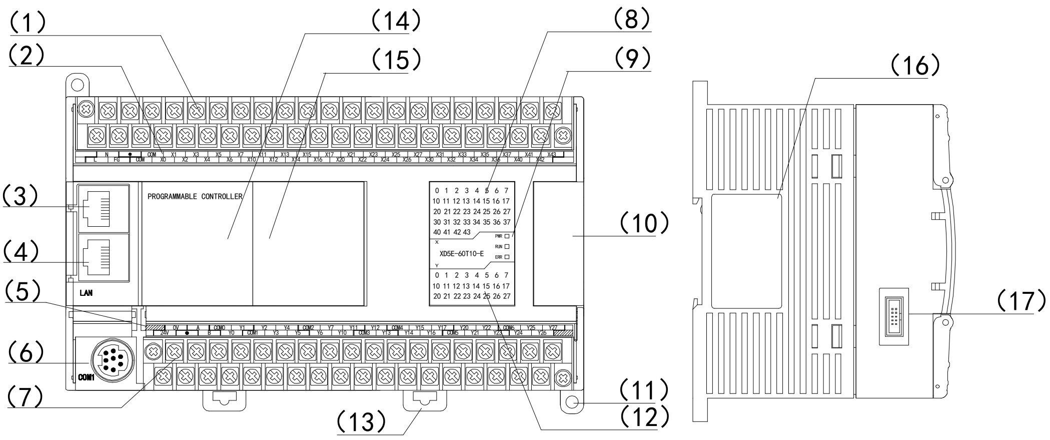
2 XD5E-30T4 structure



Each part’s name is listed below:

（1）：input terminal, power supply input, COM2
（2）：input label
（3）：COM1
（4）：Ethernet port RJ45
（5）：output label
（6）：USB port
（7）：output terminal, 24V output terminal
（8）：input indicator light
（9）：system indicator light
PWR：power
RUN：run
ERR：error
（10）：expansion module access
（11）：installation hole (2 holes)
（12）：output indicator light
（13）：rail installation hook
（14）：product label
（15）：ED module access

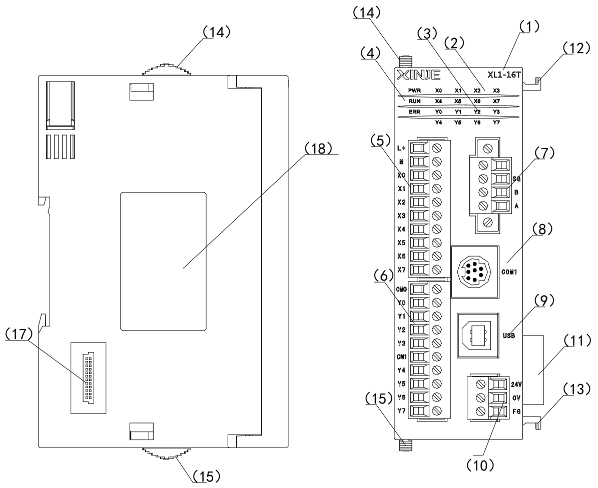
3 XD5E-60T10 XDME-60T10 structure



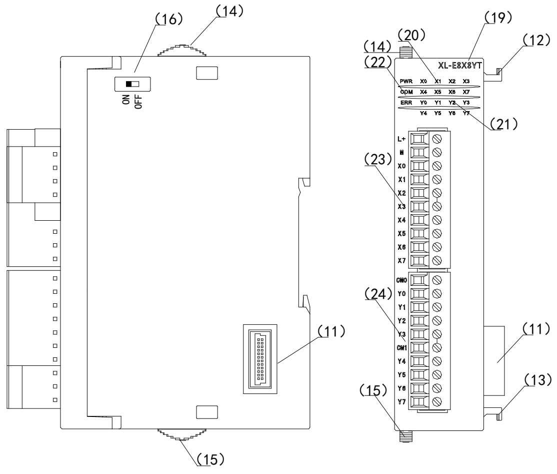
Each part’s name is listed below:

1: Input & power supply terminals
2: Input terminal label
3: RJ45 port 1
4: RJ45 port 2
5: Output terminal label
6: RS232 port (COM1)
7: output terminal, RS485 port(COM2)
8: Input action display
9: system LED
  PWR: power supply
  RUN: working
  ERR: error
10: expansion module connection port
11: installation hole (2 holes)
12: output action display
13: rail mounting hook (2 hooks)
14: expansion BD (COM4)
15: expansion BD (COM5)
16: product label
17: expansion ED (COM3)

4	XL series structure



CPU unit



Right expansion module

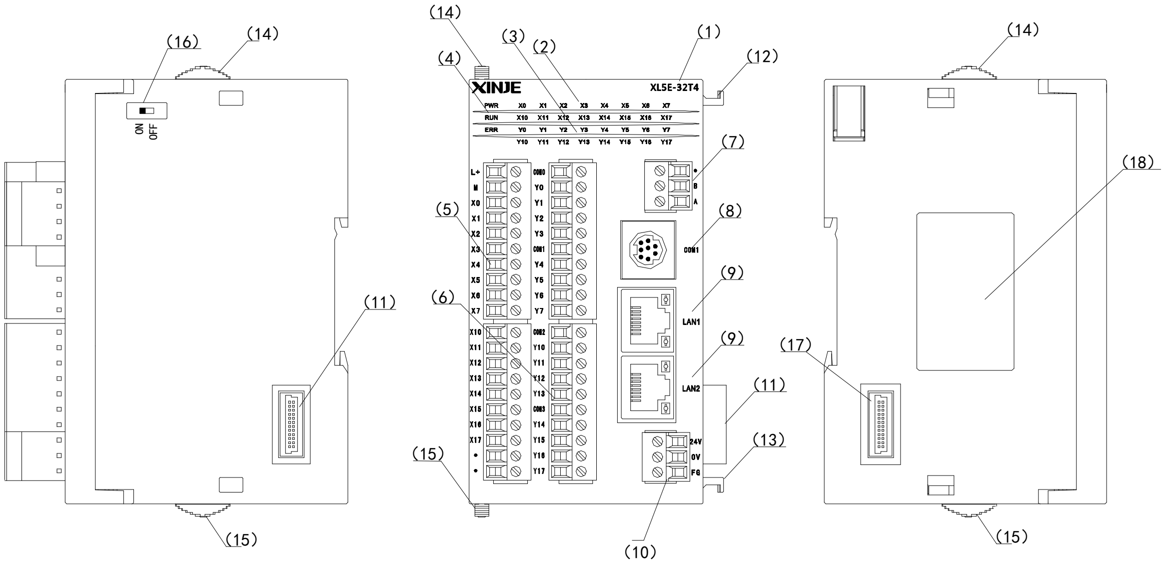
Each part’s name is listed below:

（1）：PLC model
（2）：input label and indicator light
（3）：output label and indicator light
（4）：system indicator light
PWR: power
RUN: run
ERR: error
（5）：input terminal
（6）：output terminal
（7）：RS485 port（PORT2）
（8）：RS232 port（PORT1）
（9）：USB port
（10）：power input terminal
（11）：right expansion module access
（12）：module fixed hook (up)
（13）：module fixed hook (down)
（14）：slide lock (up)
（15）：slide lock (down)
（16）：DIP switch
（17）：left expansion Ed module access
（18）：product label
（19）expansion module model
（20）：expansion module input label and indicator light
（21）：expansion module output label and indicator light
（22）：expansion module system indicator light
PWR：power
COM：communication
ERR：error
（23）：expansion module input terminal
（24）：expansion module output terminal

Note:
(1) XL3/XL5 series USB communication ports are only for download and monitoring of programs. (XL1 series does not have USB ports.)
(2) When the dial switch on the side of XL body is used for RS485 port communication, whether the PLC is the terminal? When the PLC is at the beginning or end of the bus, please turn the dial switch to ON.

(3) RS485 port of XL1 series does not have isolation, so it does not support X-NET Fieldbus function.

5	**XL5E-32T4 XLME-32T4 structure**



Each part's name is listed below:

(1) : PLC model
(2) : input label and indicator
(3) : output label and indicator
(4) : system LED
PWR: power supply
RUN: working
ERR: error
(5) : input terminals
(6) : output terminals
(7) : RS485 port (PORT2)
(8) : RS232 port (PORT1)
(9) : Ethernet port 1, 2
(10) : Power supply input terminal
(11) : right expansion module access port
(12) : module fixing hook(up)
(13): module fixing hook(down)
(14) : sliding lock (up)
(15) : sliding lock (down)
(16) : dial switch
(17) : left expansion module access port
(18) : product label

When the dial switch on the side of PLC body is used for RS485 port communication, whether the PLC is the terminal? When the PLC is at the beginning or end of the bus, please turn the dial switch to ON.

2 Specifications and parameters of CPU

This chapter mainly introduces XD/XL CPU’s general specifications, performance, dimensions, terminals arrangement and communication interfaces.
The Expansions’ description, please refer to XD series expansion module manual.

2-1． Specifications and Parameters

2-1-1． General Specifications

This specification is fit for XD and XL series PLC.

Items	Specifications
Isolation voltage	Above DC 500V 2MΩ
Anti-noise	Noise voltage 1000Vp-p 1us pulse per 1minute
Atmosphere	No corrosive, flammable gas
Ambient temperature	0℃~60℃
Ambient humidity	5%~95% (NO condensation)
USB port	USB download port, connect PC to upload/download/online monitoring
Port 0	RS-232, to connect upper computer, HMI for program or debug.
Port 1	RS-232, to connect upper computer, HMI for program or debug.
Port 2	RS-485, to connect intelligent instruments or inverters.
Ethernet port	RJ45, connect to upper device, monitoring, connect to other devices in the LAN
Installation	Use M3screws or DIN to fix
Grounding (FG)	The third type grounding (do not grounding with strong power system)

※1: XD1 series, XD2 series, XDC series, XL1 series, XDME-60, XD5E-60 models without USB port.

※2: PORT0 port only has XD1, XD2 series PLC, other models do not have this port.

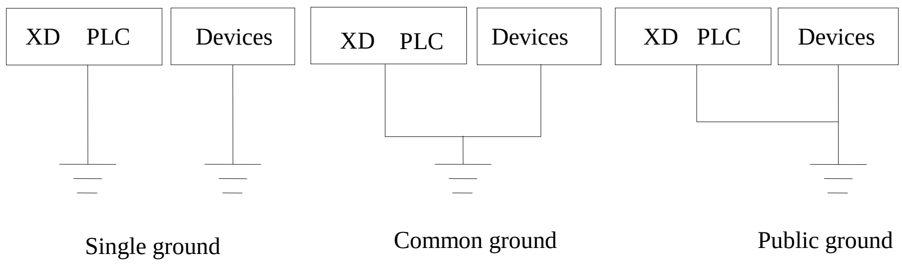
※3: XD1-16 without PORT2 is RS485 port.

※4: For XDC series PLC, PORT2 port is divided into RS232 and RS485 two communication interfaces, two communication ports can not be used at the same time.

※5: Ethernet port only has XD5E, XDME, XL5E, XLME series PLC.

※6: The DIN type should be DIN46277, with width 35 mm.

※7: The grounding should use type1 and 2, not 3.



2-1-2. Performance and Specifications

XD series PLC specifications:

<table>
<tr><th colspan="2">Items</th><th colspan="6">Specifications</th></tr>
<tr><td colspan="2">Program execution mode</td><td colspan="6">Loop scan mode</td></tr>
<tr><td colspan="2">Program mode</td><td colspan="6">Instructions and ladder</td></tr>
<tr><td colspan="2">Processing speed</td><td colspan="6">0.05us</td></tr>
<tr><td colspan="2">Power off retentive</td><td colspan="6">FlashROM and Li-battery(3V button battery)</td></tr>
<tr><td colspan="2">Users' program capacity [※1]</td><td colspan="6">XD1/XD2/XD3: 256KB,
XD5/XDM/XDC: 384KB
XD5E/XDME: 1MB
XDM-60T4L: 1.5MB</td></tr>
<tr><td rowspan="3">I/O points [※2]</td><td>Total I/O</td><td>16</td><td>24</td><td>30</td><td>32</td><td>48</td><td>60</td></tr>
<tr><td>Input</td><td>8</td><td>14</td><td>18</td><td>18</td><td>28</td><td>36</td></tr>
<tr><td>Output</td><td>8</td><td>10</td><td>12</td><td>14</td><td>20</td><td>24</td></tr>
<tr><td colspan="2">Internal Coils(X)[※3]</td><td colspan="6">1280 points: X0~X77, X10000~X11777, X20000~X20277</td></tr>
<tr><td colspan="2">Internal Coils(Y)[※4]</td><td colspan="6">1280 points: Y0~Y77, Y10000~Y11777, Y20000~Y20277</td></tr>
<tr><td colspan="2" rowspan="2">Internal Coils(M, HM)</td><td rowspan="2">11008/
87000</td><td colspan="5">XD1/XD2/XD3: M0~M7999【HM0~HM959】[※5]
XD5/XDM/XDC/XD5E/XDME: M0~M69999
【HM0~HM11999】</td></tr>
<tr><td colspan="5">For Special Use[※6]
XD1/XD2/XD3: SM0~SM2047
XD5/XDM/XDC/XD5E/XDME: SM0~SM4999</td></tr>
<tr><td colspan="2">Procedure(S)</td><td>1152/9000</td><td colspan="5">XD1/XD2/XD3: S0~S1023【HS0~HS127】
XD5/XDM/XDC/XD5E/XDME: S0~S7999
【HS0~HS999】</td></tr>
<tr><td rowspan="2">Timer(T)</td><td>points</td><td>672/7000</td><td colspan="5">XD1/XD2/XD3: T0~T575 【HT0~HT95】
XD5/XDM/XDC/XD5E/XDME: T0~T4999
【HT0~HT1999】</td></tr>
<tr><td>Spec.</td><td colspan="6">100mS timer: set time 0.1~3276.7sec.
10mS timer: set time 0.01~327.67sec.
1mS timer: set time 0.001~32.767sec.</td></tr>
<tr><td rowspan="2">Counter (C)</td><td>points</td><td>672/7000</td><td colspan="5">XD1/XD2/XD3: C0~C575【HC0~HC95】
XD5/XDM/XDC/XD5E/XDME: C0~C4999
【HC0~HC1999】</td></tr>
<tr><td>Spec.</td><td colspan="6">16 bits counter: set value K0~32,767
32 bits counter: set value -2147483648~＋2147483647</td></tr>
</table>

Data Register(D)	11048 words/90000 words/ 100000 words	XD1/XD2/XD3：D0~D7999【HD0~HD999】※5 XD5：D0~D69999※7【HD0~HD24999】 XDM/XDC/XD5E/XDME：D0~D69999【HD0~HD24999】
		For Special Use※6 XD1/XD2/XD3：SD0~SD2047 XD5/XDM/XDC/XD5E/XDME：SD0~SD4999
FlashROM Register (FD)	8144 words/ 14192 words	XD1/XD2/XD3：FD0~FD6143 XD5/XDM/XDC/XD5E/XDME：FD0~FD8191
		For Special Use※6 XD1/XD2/XD3: SFD0~SFD1999 XD5/XDM/XDC/XD5E/XDME：SFD0~SFD5999
High Speed Dispose Ability	High speed counter, pulse output, external interruption	
Password Protection	6 bits ASCII	
Self-diagnose Function	Power on self-check, monitor timer, grammar check	

XL3 series PLC specifications:

Items		**Specifications**	
Program execution mode		Loop scan mode	
Program mode		Instructions and ladder	
Processing speed		0.05us	
Power off retentive		FlashROM and Li-battery(3V button battery)	
Users' program capacity ※1		XL1/XL3: 256KB XL5: 384KB XL5E/XLME: 1MB	
I/O points ※2	Total I/O	16	32
	Input	8	16
	Output	8	16
Internal Coils(X)※3		896 points	XL1/XL3: X0~X77, X10000~X11177, X20000~X20177, X30000~X30077
		1280 points	XL5/XL5E/XLME: X0~X77, X10000~X11777, X20000~X20177, X30000~X30077
Internal Coils(Y)※4		896 points	XL1/XL3: Y0~Y77, Y10000~Y11177, Y20000~Y20177, Y30000~Y30077

		1280 points	XL5/XL5E/XLME: Y0~Y77, Y10000~Y11777, Y20000~Y20177, Y30000~Y30077
Internal Coils(M, HM)		11008/92000 points	XL1/XL3: M0~M7999【HM0~HM959】※5 XL5/XL5E/XLME: M0~M69999【HM0~HM11999】
			Special※6 XL1/XL3: SM0~SM2047 XL5/XL5E/XLME: SM0~SM4999
Procedure(S)		1152/9000 points	XL1/XL3：S0~S1023【HS0~HS127】 XL5/XL5E/XLME：S0~S7999【HS0~HS999】
Timer(T)	points	672/7000 points	XL1/XL3: T0~T575【HT0~HT95】 XL5/XL5E/XLME: T0~T4999【HT0~HT1999】
	Spec.	100mS timer: set time 0.1~3276.7sec. 10mS timer: set time 0.01~327.67sec. 1mS timer: set time 0.001~32.767sec.	
Counter (C)	points	672/7000 points	XL1/XL3: C0~C575【HC0~HC95】 XL5/XL5E/XLME: C0~C4999【HC0~HC1999】
	Spec.	16 bits counter: set value K0~32,767 32 bits counter: set value -2147483648~＋2147483647	
Data Register(D)		11048/100000 words	XL1/XL3: D0~D7999【HD0~HD999】※5 XL5/XL5E/XLME: D0~D69999【HD0~HD24999】
			Special※6 XL1/XL3: SD0~SD2047 XL5/XL5E/XLME: SD0~SD4999
FlashROM Register (FD)		7120/14192 words	XL1/XL3: FD0~FD5119 XL5/XL5E/XLME: FD0~FD8191
			Special※6 XL1/XL3: SFD0~SFD1999 XL5/XL5E/XLME: SFD0~SFD5999
High Speed Dispose Ability		High speed counter, pulse output, external interruption	
Password Protection		6 bits ASCII	
Self-diagnose Function		Power on self-check, monitor timer, grammar check	

Note:

※1: The users' program capacity means the maximum program capacity when download in secret.

※2: I/O points mean terminal number that users can connect from outside.

※3: X stands for the internal input relays and can be used as middle relay when input points are exceeded.

※4: Y stands for the internal output relays and can be used as middle relay when output points are exceeded.

※5:【】marks the default power off retentive area, this area can't be changed.

※6: For special use means special usage registers that are occupied by system, can’t be applied for other usage. For details, please refer to Appendix 1.

※7: The XD5 series data registers for firmware versions V3.5.3 and above range from D0 to D69999, and the XD5 series data registers for firmware versions V3.5.2 and below range from D0 to D59999.

※8: Input and output coils no. is octal, other coils and registers are decimal.

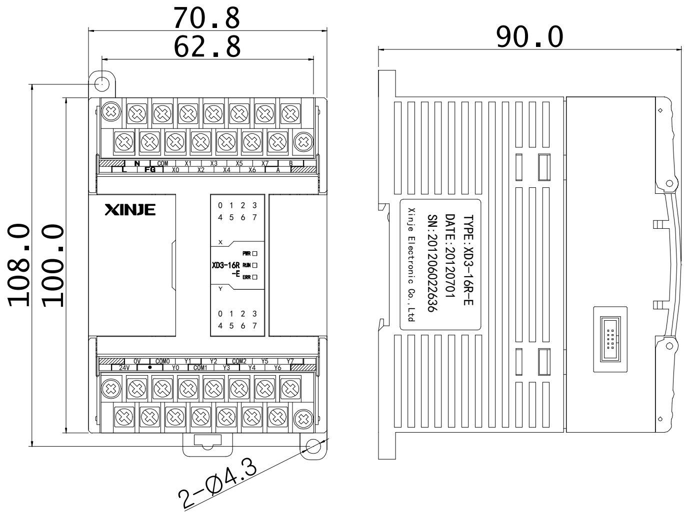
※9: The I/O which is not connected to other device can be used to internal coil.

2-2. Dimensions

Note: the height is 79.9mm for PLC hardware version v3.4 and below.

1 Picture 1

(Unit: mm)

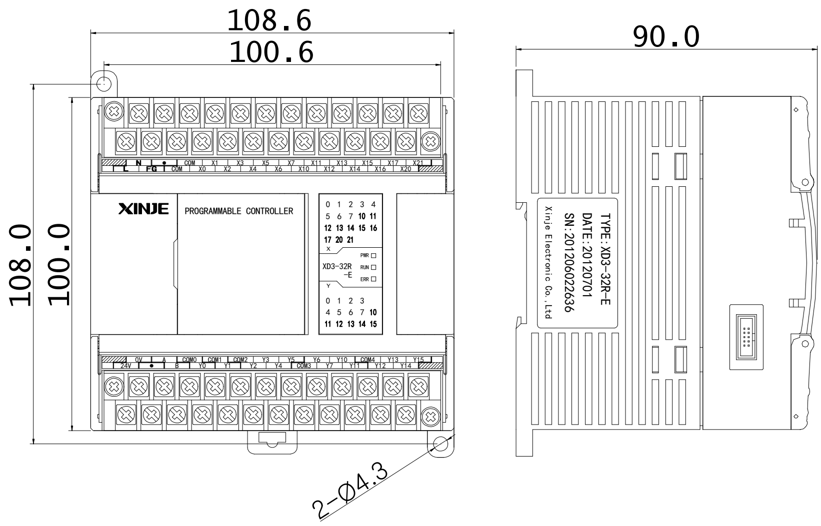


Suitable Model:

Series	Points
XD1	16
XD2	
XD3	
XD5	

2 Picture 2

(Unit: mm)

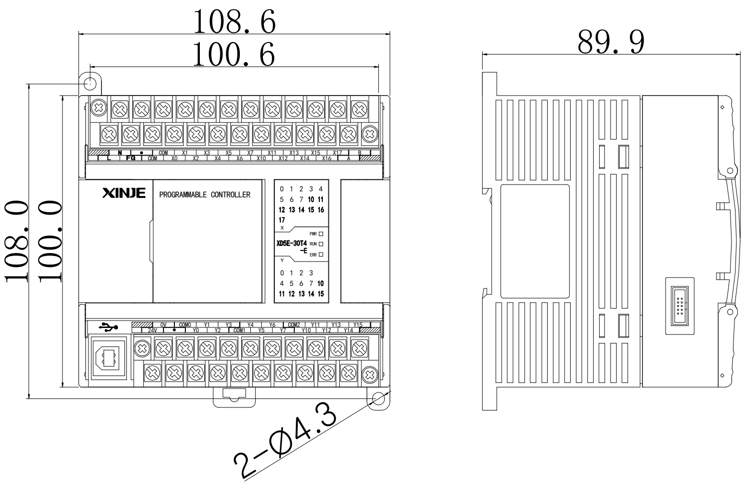


Suitable Model:

Series	Points
XD1	24/32
XD2	
XD3	
XD5	
XDM	
XDC	

3	Picture 3

(Unit: mm)

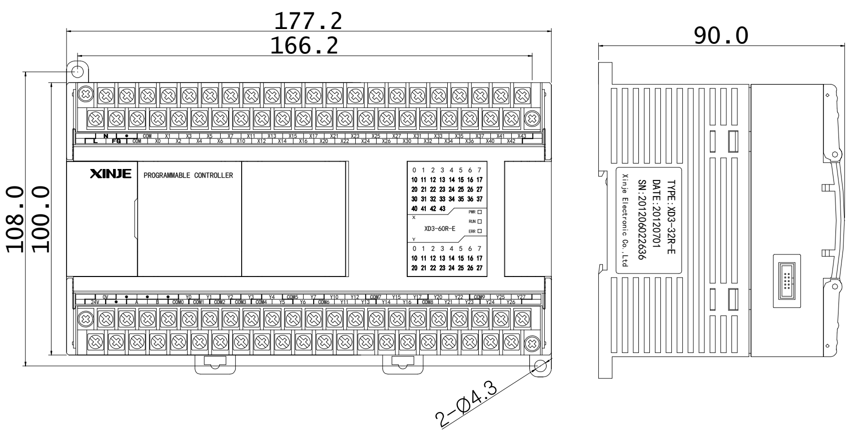


Suitable Model:

Series	Points
XD5E	30

4	Picture 4

(Unit: mm)

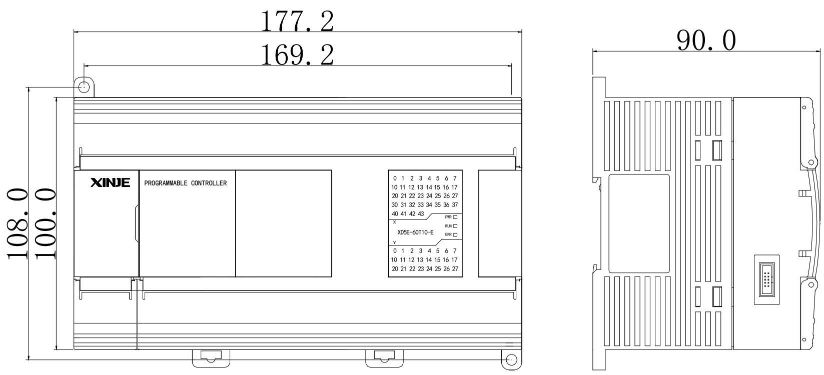


Suitable Model:

Series	Points
XD2	48/60
XD3	
XD5	
XDM	
XDC	

5	Picture 5

(Unit: mm)

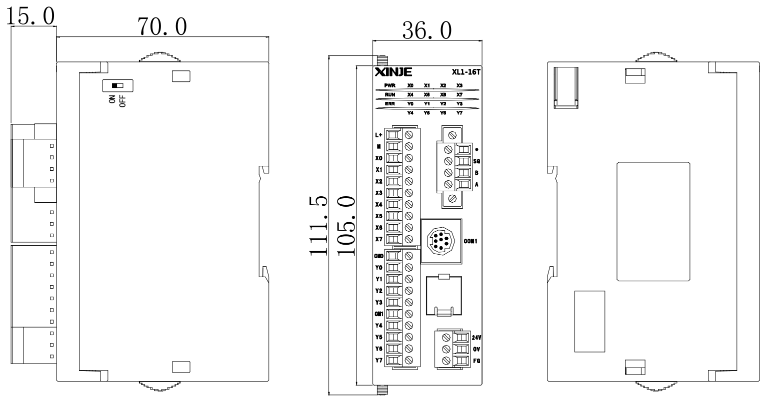


Suitable model:

Series	Points
XD5E	60
XDME	60

6	Picture 6

(Unit: mm)

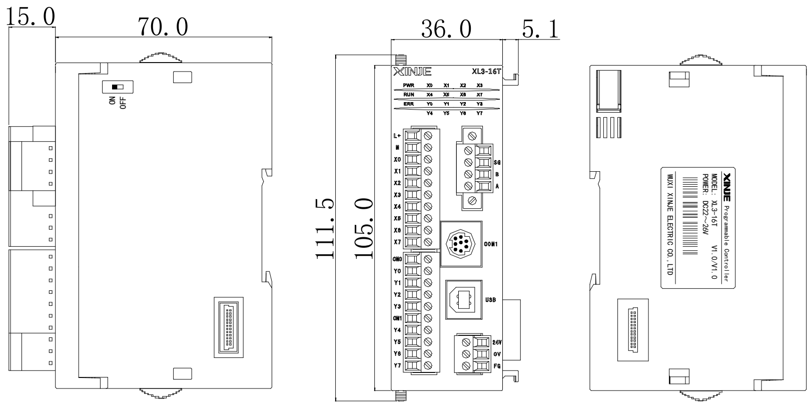


Suitable model:

Series	Points
XL1	16

7	Picture 7

(Unit: mm)

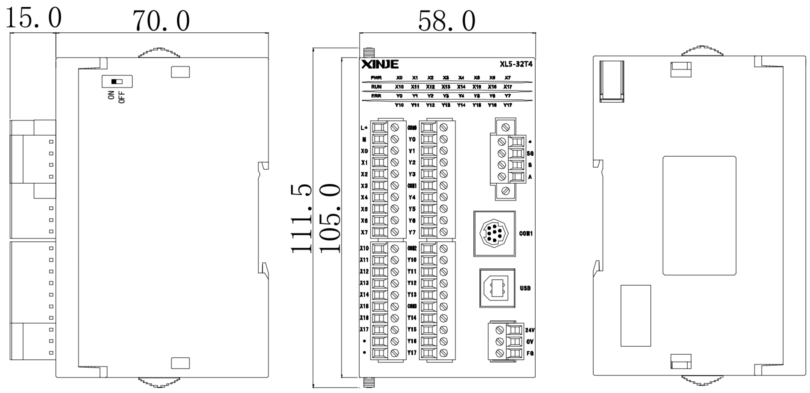


Suitable model:

Series	Points
XL3	16

8	Picture 8

(Unit: mm)

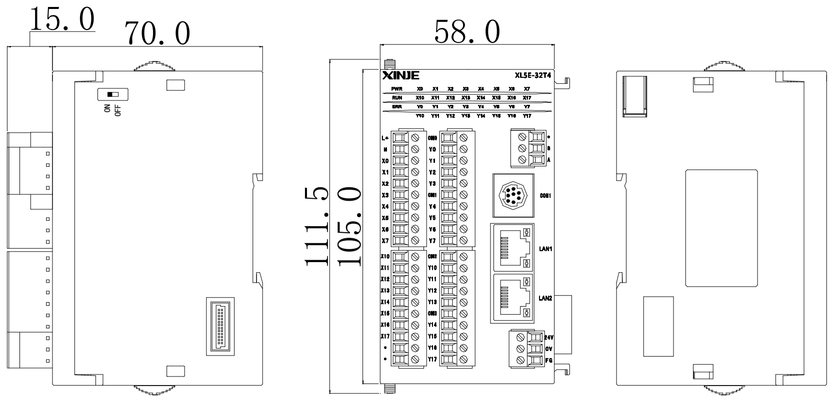


Suitable model:

Series	Points
XL5	32

9	Picture 9

(Unit: mm)



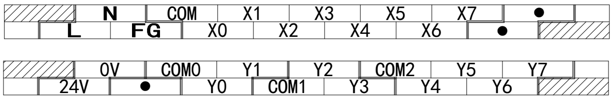
Suitable model:

Series	Points
XL5E	32
XLME	

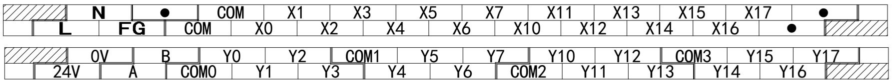
2-3. Terminal arrangement

2-3-1. XD series terminal arrangement

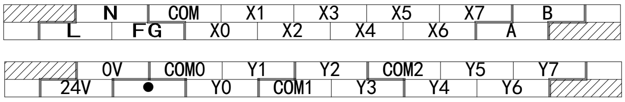
- Graph A



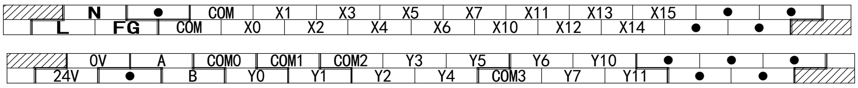
- Graph B



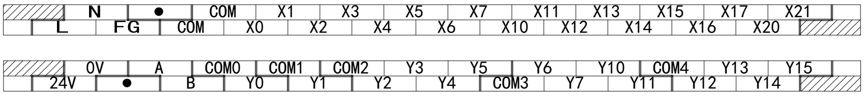
- Graph C



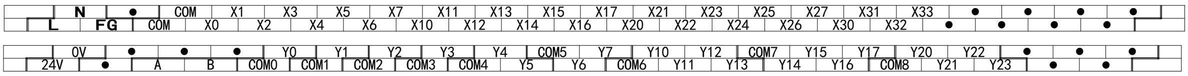
- Graph D



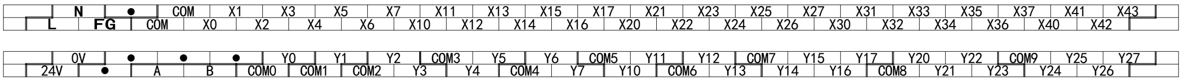
- Graph E



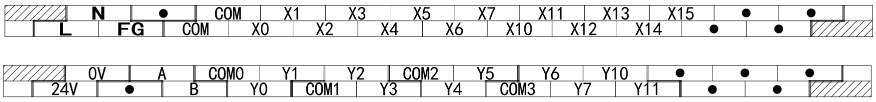
- Graph F



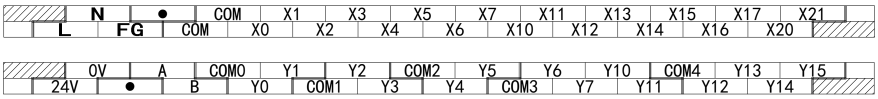
- Graph G



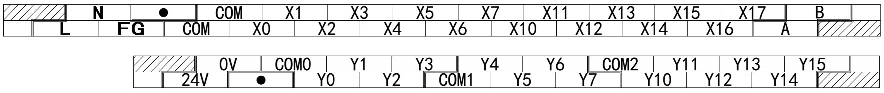
- Graph H



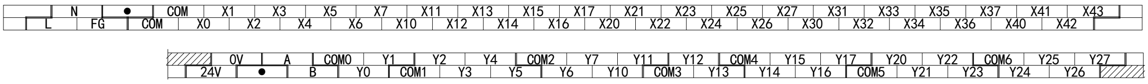
- Graph I



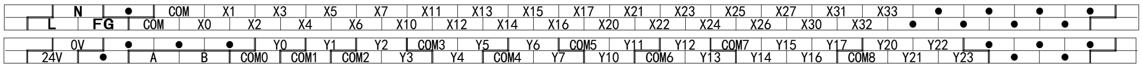
- Graph J



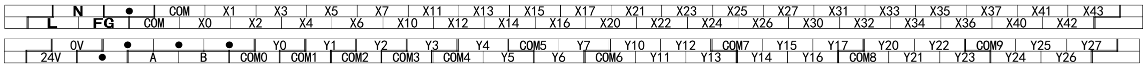
- Graph K



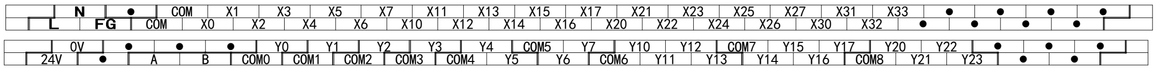
- Graph L



- Graph M



- Graph N



The graph for the model:

graph	Suitable model	Note
A	XD1-16	8 input /8 output
B	XD1-32	16 input /16 output
C	XD2-16、XD3-16、XD5-16	8 input /8 output
D	XD2-24、XD3-24、XD5-24、XDM-24	14 input /10 output
E	XD2-32、XD3-32、XD5-32、XDM-32	18 input /14 output
F	XD2-48、XD3-48、XD5-48、XDC-48	28 input /20 output
G	XD5-60T6、XD5-60T10、XDM-60T10	36 input /24 output
H	XD5-24T4、XDM-24T4、XDC-24T	14 input /10 output
I	XD5-32T4、XDM-32T4、XDC-32T	18 input /14 output
J	XD5E-30T4	16 input /14 output
K	XD5E-60T10、XDME-60T10	36 input /24 output
L	XD5-48T6	28 input /20 output
M	XD2-60、XD3-60、XD5-60、XD5-60T4、XDC-60、XDM-60T4	36 input /24 output
N	XD5-48T4	28 input /20 output

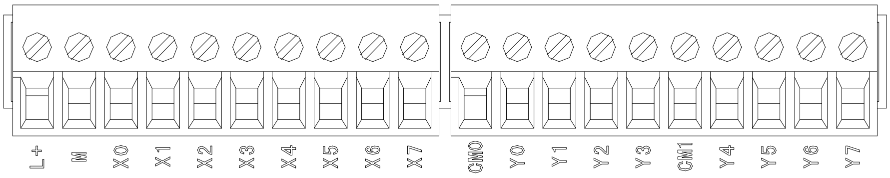
Note:

1. Transistor and relay mixed type, only the first two channels are transistor output, others are relay output.
2. E type PLC power supply terminal is L, N; C type PLC power supply terminal is 24V+, 24V-.

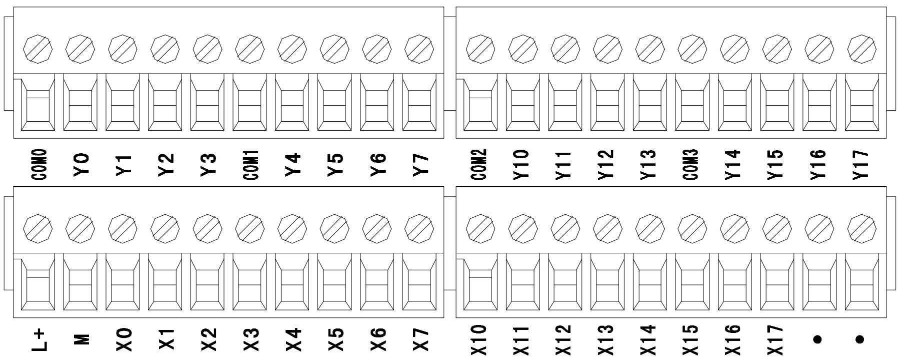
3. The 24V, 0V terminal is external output terminal, it can supply power for module and sensor. Do not over the max output current when using, please refer to chapter 4-1.
4. FG ground terminal can shield the interference, it can single connect to the ground.
5. The com terminal of input corresponding to all the input points; the com terminal of output corresponding to different output points. Please connect the wire as the division on the terminal label.
6. The terminals A and B on the terminal row are RS485 communication interfaces, A is RS485+, B is RS485-.

2-3-2. XL series terminal arrangement

XL series I/O terminals:



XL1-16, XL3-16



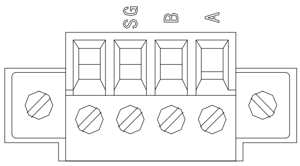
XL5-32T4, XL5E-32T4, XLME-32T4

Note:

(1) XL series PLC has no built-in 24V power supply, it needs external DC24V power supply. L+ connects to 24V+, M connects to 24V-.

(2) The common terminal of input terminal X0-X7 is M, the common terminal of output terminal Y0-Y3 is CM0. The common terminal of output terminal Y4-Y7 is CM1.

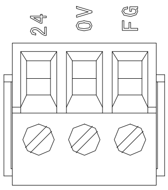
XL series RS485 terminals:



Note:

(1) A is RS485+, B is RS485-. Please connect A to A, B to B when communicating.
(2) SG is communication ground terminal, it can connect to SG terminal of servo drive in general.
(3) RS485 port of XL1 series does not have isolation, so it does not support X-NET Fieldbus function.

XL series PLC power supply terminals:



Note:

(1) PLC power supply input terminals are 24V, 0V.
(2) FG is ground terminal for shield interference, please connect to ground separately.

Connection head specifications of terminal

When wiring XL series PLC, its wiring head should meet the following requirements:

(1) The stripping length is 9 mm;
(2) Flexible conductors with bare tubular ends are 0.25-1.5 square.
(3) Flexible conductors with tubular pre-insulated end is 0.25-0.5 square.

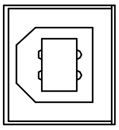
2-4. Communication Ports

XD series PLC have USB port, port0 (RS232), port0 (RS232, only XD1/XD2 support), port1(RS232), port2 (RS485, XDC is RS485/RS232), Ethernet port (XD5E/XDME series support). USB port can high-speed upload, download and monitor program, port0, port1 and port2 can communicate and download program, Ethernet port can download and monitor program, communicate with other equipment in the LAN.

XL series PLC have USB port(XL1/XL5E/XLME without this port), port1 (RS232), port2 (RS485), Ethernet port(XL5E/XLME support). USB port can high-speed upload, download and monitor program, port1 and port2 can communicate and download program. Ethernet port can download and monitor program, communicate with other equipment in the LAN.

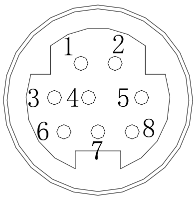
1 USB port

USB port only can download program but cannot communicate with other device. Please use printer USB cable or XINJE USB cable to download.



2 RS232 port

RS232 port can upload, download program and communication. Port0 only supports X-NET mode, port 1 supports Modbus and X-NET mode. The pin diagram of port0, port1, port2 are shown as below:



4: RxD
5: TxD
8: GND

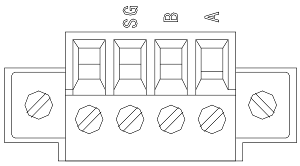
Mini Din 8-core plug-in (holes)

3 RS485 port

The port2 of XD series PLC are the terminal A and B. A is RS485+, B is RS485-.(

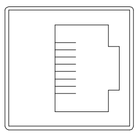
XD1-16 does not have RS485 port.

The port2 of XL series PLC is separately, they are terminal A, B and SG(signal ground).



4	**Ethernet port**

The Ethernet port is RJ45 access, can upload, download program, online monitoring, remote monitoring, communicate with other device in the LAN.



5	**Program Cable**

download program via RS232 port must use XINJE XVP cable.

Program cables are as below:

Mini Din 8-pin (pins) DB9-pin (holes)

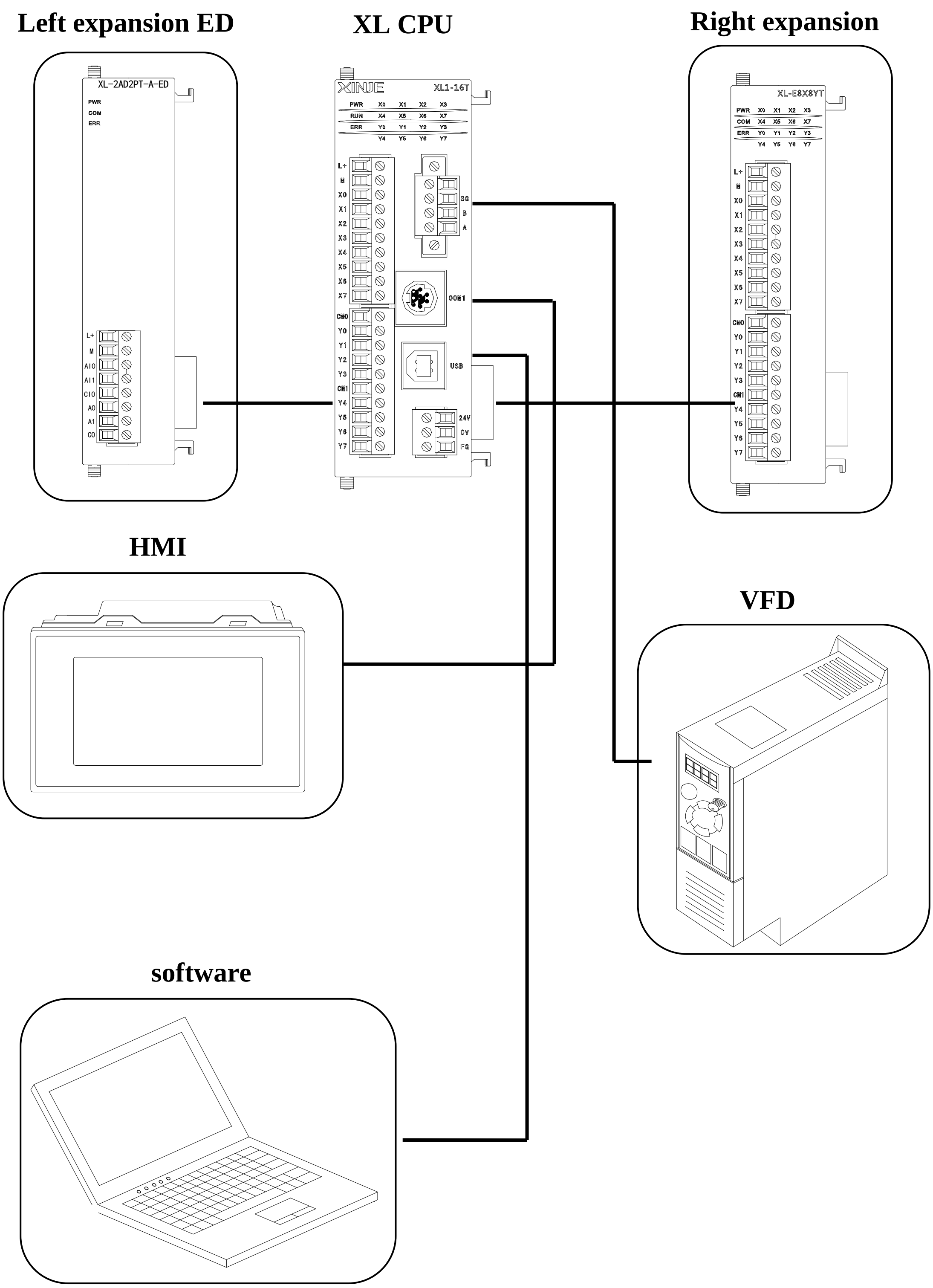
Note: above diagram is for DVP cable. If it is XVP cable, please connect pin1 of Mini Din8 and pin7 of DB9 based on above diagram.

3 System Structure

As the controllers, XD/XL series PLC can connect with many kinds of peripheral devices, expansion devices. In this chapter, we mainly introduce PLC basic units, peripheral devices and expansion devices connection. And also introduce the connection principle of PLC with expansions, products installation, points calculation, address number distribution etc.
For the introduction of expansions, please refer to chapter 8.

3-1. System Structure

According to XD/XL series PLC basic configuration, we build the system structure chart as below. We can know the general connection among PLC, peripheral equipments and expansions from the chart; also classic applications of PLC’s each COM port, connection and expansions etc.



※1: In the above chart, the communication devices connected to the COM port are only samples for your reference. Each COM port can connect with many devices in real applications.

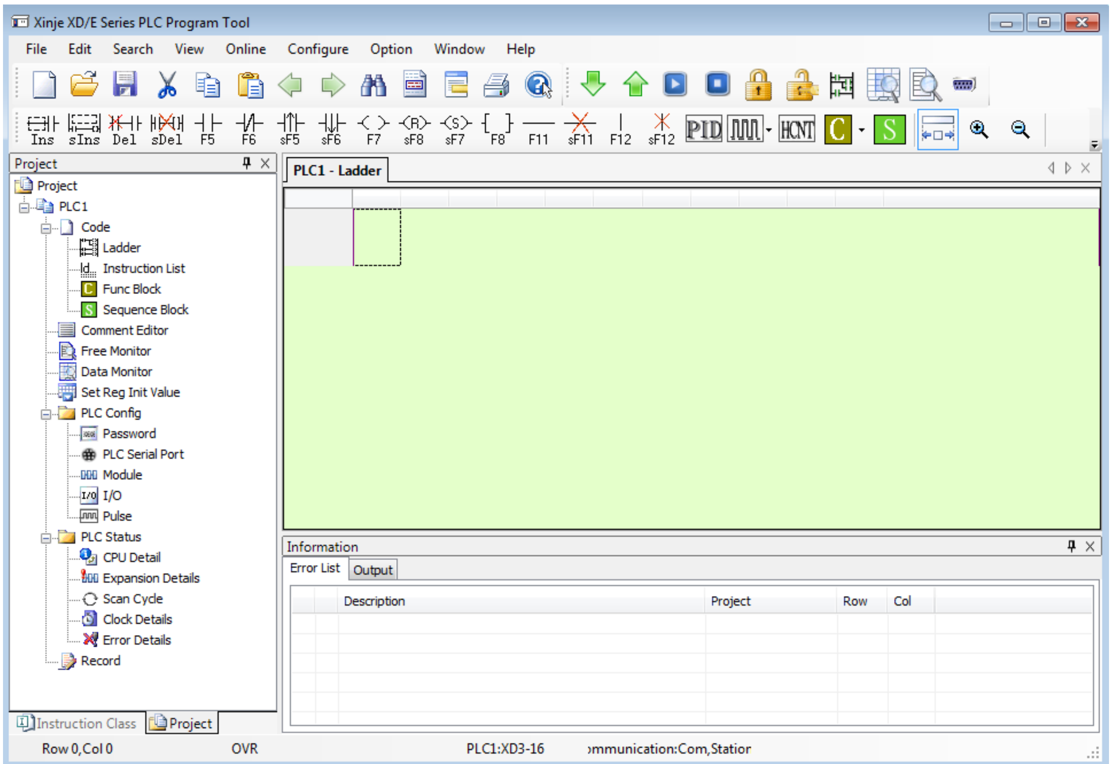
3-2. Peripheral Devices

XD/XL series PLC basic units can work with many kinds of peripheral devices.

3-2-1. Program Software

Users can write to or upload program from PLC, real time monitor PLC, configure PLC etc; After installing XDPPro on your PC, use the program cable, via port1 or USB port on PLC(CPU Units), to link PLC with XDPPro.

- Program Interface



※1: Please use the download cable offered by XINJE Company or make the cable by yourself. Connecting method, please refer to chapter 2-4.

3-2-2 Human Machine Interface (HMI)

The HMI link PLC to the operators. The HMI can send the commands from operators to PLC, and then PLC executes the commands.

XD/XL series PLC support diverse brands of HMI; the connection is based on the communication protocol. Generally communicate via Modbus protocol, the detailed parameters setting depends on the HMI.

The Xinje HMI can work with PLC directly (the communication parameters are set in accordance already). Presently Xinje HMI has TG, TH, TP, OP, MP series.

1 TG,TH series

- Size 4.3", 7", 8", 10.1", 10.4", 15.6''
- Display 16 million color,65536 color
- Operation touch screen
- Interface RS232, RS422, RS485, USB, Internet port
- Communication Work with many PLC brands, inverters, instruments etc.
  Drive panel printer directly, support multiple printer.
  Dual COM ports make it possible that work with 2 different devices at the same time.
  Support free format protocol, users can write the driver program freely
- Recipe input different group of data in the table
- Picture Rich stereoscopic 3D gallery, font effects, data collect, data backup etc.
- Password nine-level setting
- Advanced function animation design and so on

2 OP Series

- Size 3.7''
- Display Blue LCD, 256 true color
- Buttons Nr. 7, 20 not touch screen
- Interface RS232, RS422, RS485
- Communication work with many PLC brands.
  Communicate with Xinje Inverters
- RTC Built-in

3 MP Series

- Size 3.7”
- Display STN-LCD
- Buttons Nr.: 26, 20, the LCD is touch screen
- Interface RS232, RS422, RS485
- Communication work with many PLC brands.
  Communicate with Xinje Inverters
- RTC: Built-in

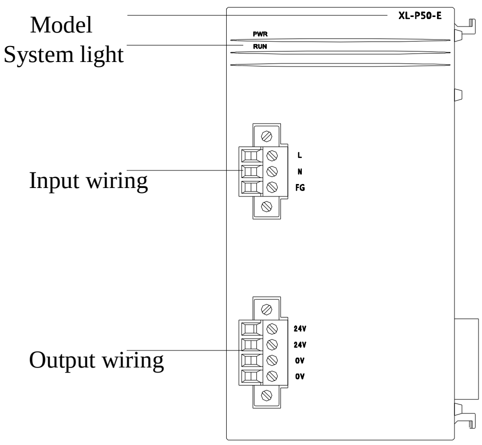
3-2-3 XL adapter power supply

XL series PLC can use external power supply or XL special power supply module XL-P50-E.

1 Basic specification

Item	Specification
Power supply	AC85-265V
Output voltage	DC24V
Output current	2A
Air	No corrosive and glammable gas
Ambient temperature	0℃~60℃
Ambient humidity	5%RH~95%RH (no condensation)
Installation	Install on the rail directly
Ground	The third ground (cannot connect to ground with strong power system)

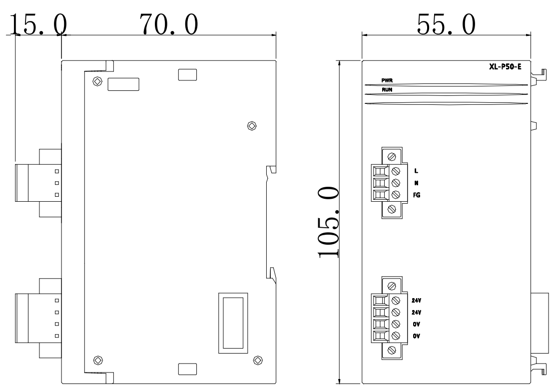
2 Structure



Structure name	Function
Model	The model of the product
System light	PWR: power light, always ON when the module is energized RUN: run light, always ON when the module is running well
Input wiring	L, N: power supply input terminal FG: ground terminal
Output wiring	Can output two groups of 24V, 0V power supply

3 Dimensions

Unit: mm



3-3. Configuration Principle

COM port

- XD/XL series PLC (CPU units) are usually equipped with port1 and port2.
- In principle, both ports can be used to program, download, communication; but please make sure not change the parameters of two ports at one time, otherwise the ports can't be used to program and download any more.
- Port1 is equipped with RS232. Port2 is RS485. The two ports are independent.

About Expansion Devices

- Generally, one CPU unit can work with different types of expansions, can expand digital I/O, analog I/O, temperature control etc.
- XD1/XD2 cannot support expansion module, XD3 can work with 10 expansions and XD5/XDM/XDC/XD5E /XDME can connect 16 modules.

- XL1 does not support extension modules, XL3 series can expand up to 10 modules, XL5/XL5E/XLME series can expand up to 16 modules.
- After connecting the CPU unit with the expansion, if the "PWR" LED of expansion ON, then the expansion can work properly; after installing the BD card to CPU unit, users need to configure it before using;

How to calculate the I/O

- I/O points include actual input and output points.
- After connect with the expansions, the total I/O points=I/O on basic unit + I/O on expansions.
- Digital I/O is octal.
- Analog I/O is decimal.
- After expansion, the total I/O can up to 572 points.

How to calculate the I/O

Basic Unit XD3-32R-E (18I/14O) connect with 5 XD-E8X8Y expansions, then the total I/O points should be:

Input Points: 18 + 8 *5 = 58

Output points: 14 + 8 *5= 54

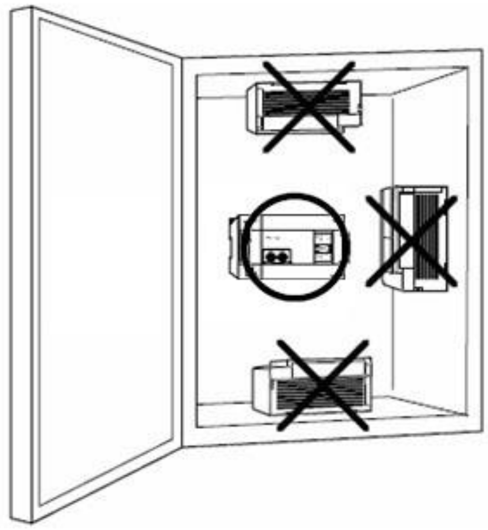
Total points: Input+ Output = 58+54=112

3-4．ID Assignment of Expansions

Sign	Name	Range		points
X	Input points	XD XL	X10000～X10077（#1 extension module） …… X11100～X11177（#10 extension module） …… X11700～X11777（#16 extension module）	1024
		XD	X20000～X20077（#1 extension BD） X20100～X20177（#2 extension BD）	128
		XD XL	X30000～X30077（#1 extension ED）	64
Y	Output points	XD XL	Y10000～Y10077（#1 extension module） …… Y11100～Y11177（#10 extension module） …… Y11700～Y11777（#16 extension module）	1024
		XD	Y20000～Y20077（#1 extension BD） Y20100～Y20177（#2 extension BD）	128
		XD XL	Y30000～Y30077（#1 extension ED）	64
ID	extension module	XD XL	ID10000～ID10099（#1 extension module） …… ID10900～ID10999（#10 extension module） …… ID11500～ID11599（#16 extension module）	1600
	extension BD	XD	ID20000～ID20099（#1 extension BD） ID20100～ID20199（#2 extension BD）	200
	extension ED	XD XL	ID30000～ID30099（#1 extension ED）	100
QD	extension module	XD XL	QD10000～QD10099（#1 extension module） …… QD10900～QD10999（#10 extension module） …… QD11500～QD11599（#16 extension module）	1600
	extension BD	XD	QD20000～QD20099（#1 extension BD） QD20100～QD20199（#2 extension BD）	200
	extension ED	XD XL	QD30000～QD30099（#1 extension ED）	100

3-5. Install The Products

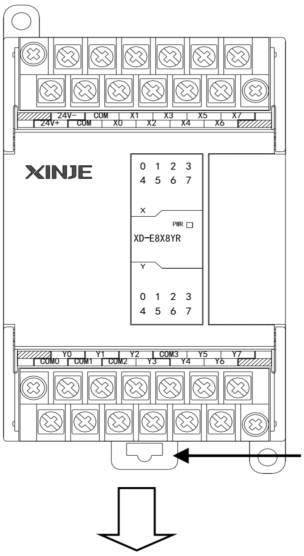
1 Installation Position



2 Installation Method

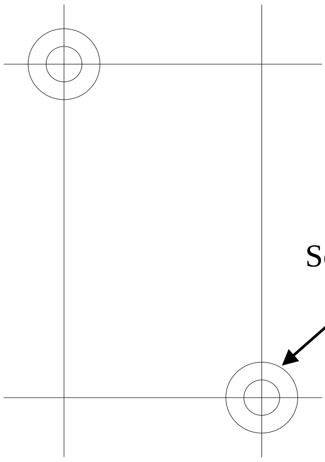
Use DIN or screws to install the CPU units and expansions.

- DIN46277



DIN installation

- Directly install by screws



Screw M3

Basic units or expansion modules install on DIN46277 rail (width 35mm). Pull down the hook on DIN rail and take down the product.

3 Installation Environment

Please install the products according to chapter 2-1-1.

4 Power Supply Specification and Wiring Method

In this chapter, we tell the structure, specification and external wiring of XD/XL series PLC. The wiring method differs due to different models, and the main difference is the terminals' position. About terminals arrangement, please refer to chapter 2-3.

4-1. Power Supply Specifications

The power supply specifications of XD series PLC (Type with '-E' is AC power, type with '-C' is DC power).
XL series PLC power supply only supports DC type.

1 AC power

Items	Content
Rated Voltage	AC100V~240V
Allowed Voltage Range	AC100V~240V
Rated Frequency	50/60Hz
Allow momentary power off time	Interruption Time≤0.5 AC cycle, interval≥1second
Impulse Current	Max 40A below 5mS/AC100V max 60A below 5mS/AC200V
Maximum Power Consumption	12W
Power Supply for Sensor	24VDC±10% 16 points max is 200mA, 32 points max is 400mA

※1: Please use the wire cable more than 2mm^2 to avoid the decrease of voltage.

※2: Even power off in 10ms, the PLC can still keep working. But when power is off for long time or voltage abnormally decrease, the PLC will stop working, output will be OFF. When power is on again, the PLC will run automatically.

※3: The grounding terminals on basic units and expansions connect together, and use the third type grounding.

2 DC Power

Items	Content
Rated Voltage	DC24V
Allowed Voltage Range	DC21.6V~26.4V
Input Current (Only for basic unit)	120mA DC24V
Allow momentary power off time	10ms DC24V
Impulse Current	10A DC26.4V
Maximum Power Consumption	12W
Power Supply for Sensor	24VDC±10% 16 points max is 200 mA, 32 points max is 400mA

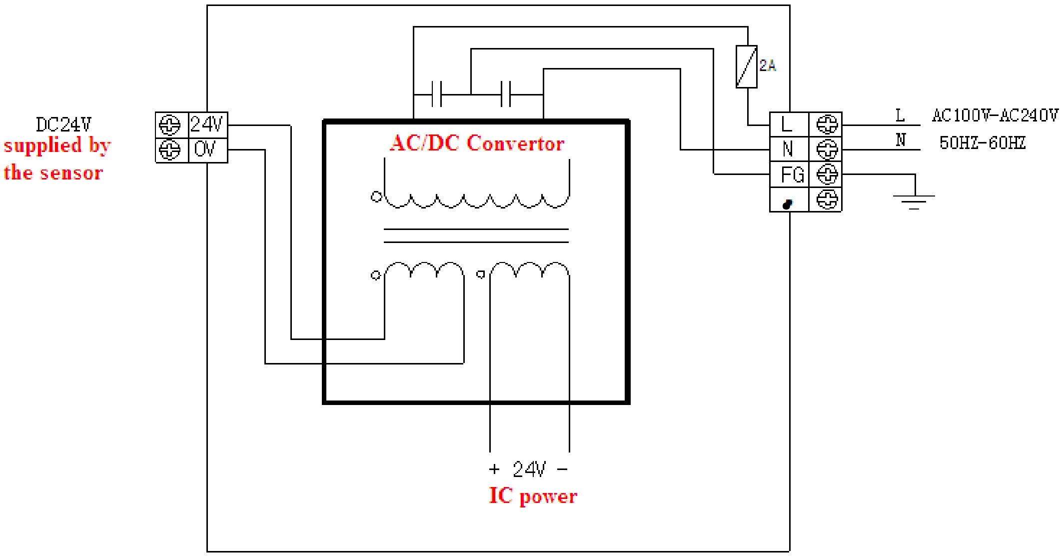
※1: XD series PLC provides DC24V power supply (terminal 24V, 0V), it can be power supply for sensor, 16 points PLC DC24V is 200mA, 24/32/48/60 points PLC DC24V is 400mA. This terminal cannot connect to external power supply.

※2: ● is empty terminal, do not use it.

※3: Please connect the com terminal for basic unit and expansion module.

4-2． AC Power Supply and DC Input

1 Connection



※1: Connect the power supply to L, N terminals.

※2: 24V, 0V terminals can supply power 200mA/DC24V for 16 points, and power 400 mA/DC24V for 32 points by sensor. Besides, the terminals power can not be supplied by outside power.

※3: ● terminal is idle, do not wire outside or work as middle relay terminals.

※4: Please connect the COM terminals on basic units and expansions together.

5 Input Specifications and Wiring Methods

In this chapter we will introduce the input specification and external wiring methods of XD/XL series PLC. The connection methods differ due to different models and the main difference is the terminals’ arrangement. Each model’s terminal arrangement, please refer to chapter 2-3.

5-1. Input Specification

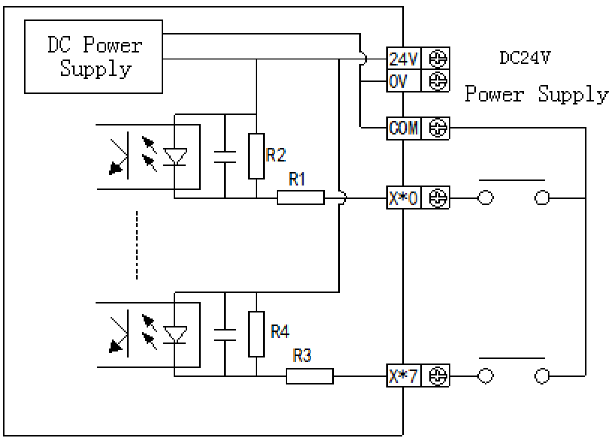
5-1-1. XD series input specification

XD series PLC input specification has NPN and PNP two modes, we will introduce the internal structure and wiring methods of the two modes as below:

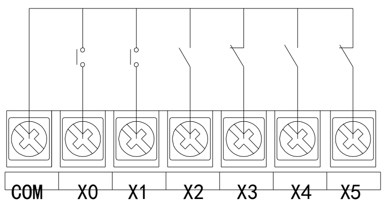
1	Basic Units

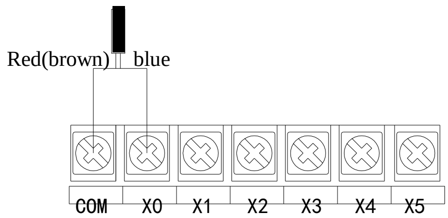
- NPN mode

Input signal's voltage	DC24V ±10%
Input signal's current	7mA/DC24V
Input ON current	Above 4.5mA
Input OFF current	Under 1.5mA
Input response time	About 10ms
Input signal's form	Contact input or NPN open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON

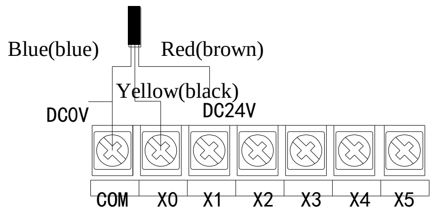


NPN wiring example





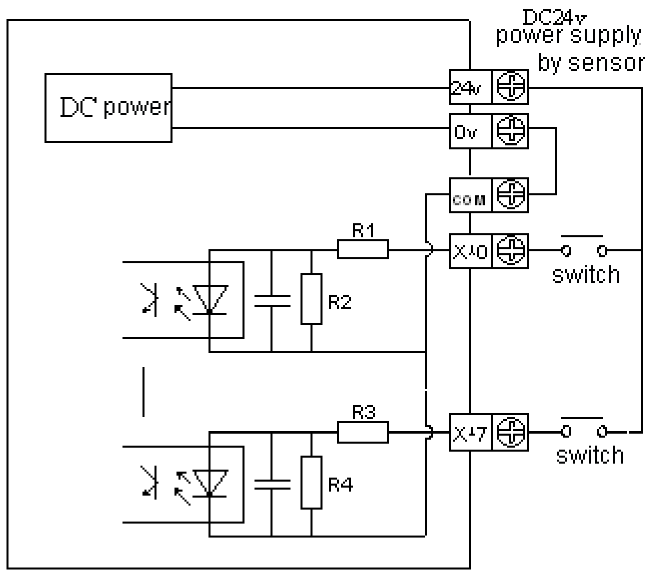
Switch **two-wire (NO, NC) proximity switch**



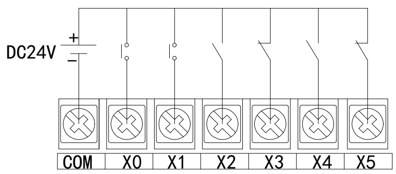
Three-wire(NPN) proximity switch

- PNP mode

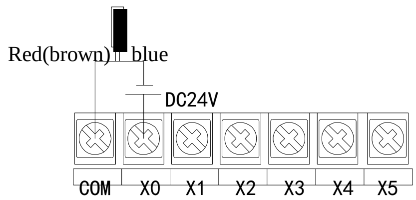
Input signal's voltage	DC24V±10%
Input signal's current	7mA/DC24V
Input ON current	Above 4.5mA
Input OFF current	Under 1.5mA
Input response time	About 10ms
Input signal's form	Contact input or PNP open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON



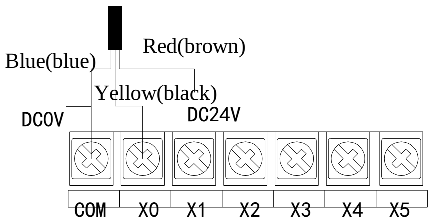
PNP wiring example:



Switch



two-wire (NO, NC) proximity switch



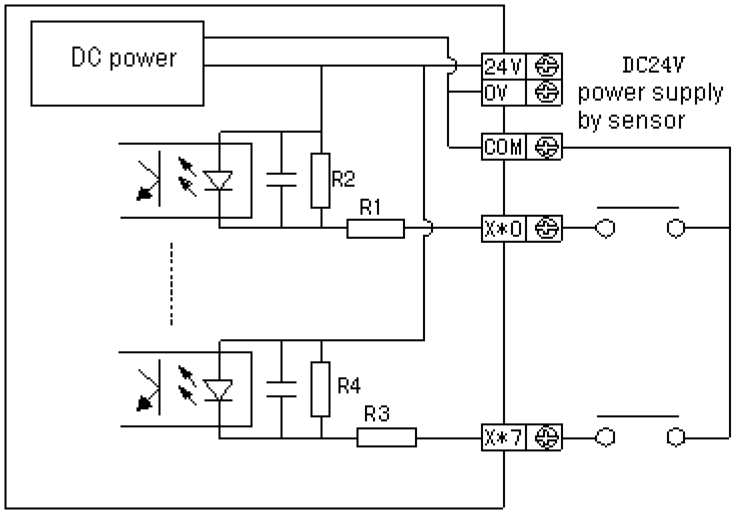
three-wire (PNP) proximity switch

note: the DC24V is provided by the PLC, no need to cnonect DC0V to com of input terminal. If using external power supply, it needs to connect it.

2 Expansion modules

- NPN mode

Input signal's voltage	DC24V ±10%
Input signal's current	7mA/DC24V
Input ON current	Above 4.5mA
Input OFF current	Under 1.5mA
Input response time	About 10ms
Input signal's form	Contact input or NPN open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON



- PNP mode

Input signal's voltage	DC24V ±10%
Input signal's current	7mA/DC24V
Input ON current	Above 4.5mA
Input OFF current	Under 1.5mA
Input response time	About 10ms
Input signal's form	Contact input or PNP open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON

5-1-2. XL series input specification

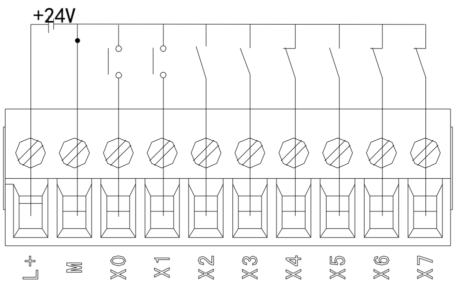
XL series PLC input is NPN mode, below is input specification and wiring method.

- **Input specification of CPU unit and expansion module (NPN mode)**

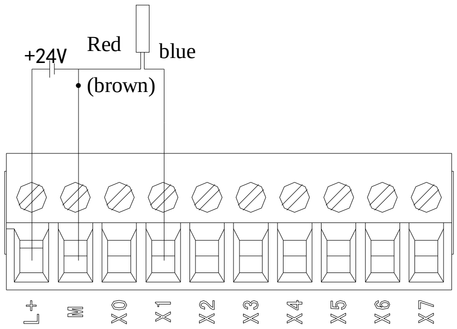
Input signal's voltage	DC24V ±10%
Input signal's current	7mA/DC24V
Input ON current	Above 4.5mA
Input OFF current	Under 1.5mA
Input response time	About 10ms

Input signal's form	Contact input or NPN open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON

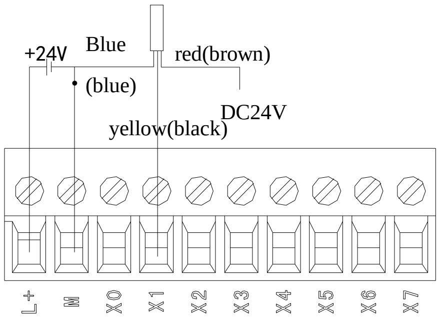
- **Wiring method of CPU unit and expansion module(NPN mode)**



Switch wiring



2-wire(NO or NC) proximity switch wiring



3-wire(NPN mode) proximity switch wiring

➢ Input terminal

It need to connect external DC24V power supply for PLC. Please connect 24V to L+, 0V to M. The input is ON when the input terminal and M pass through by connecting no voltage contactor or NPN open collector transistor, the related input light is ON.

➢ Input circuit

The first circuit and secondary circuit is isolated by optical coupler, the C-R filter is installed in secondary circuit. It can prevent from error operation caused by input vibration or noise. For input ON to OFF or OFF to ON, the response time is about 6ms inside PLC. The input terminal has internal digital filter.

➢ Input sensitivity

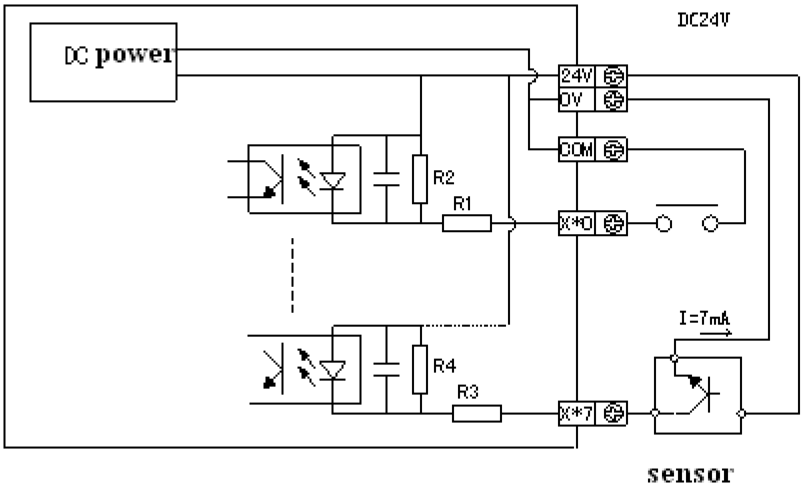
The input current is 7mA, but for reliable action, the input ON current must be above 4.5mA, the input OFF current is below 1.5mA.

5-2. DC Input Signal (AC power supply)

Below contents are only fit for XD series PLC.

1	DC Input Signal

- NPN mode

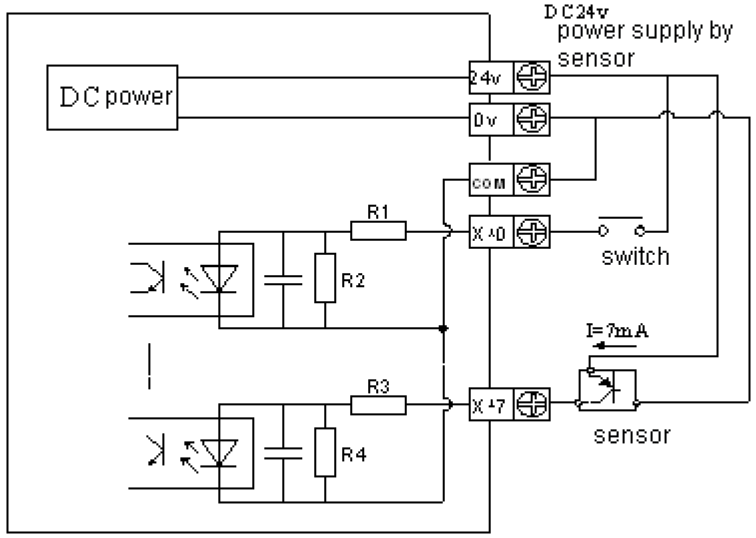


➢ Input terminals
When connect input terminals and terminal COM with contact without voltage or NPN open collector transistor, if input is ON, LED lamp will light which indicates input is ON. There are many input terminals COM to connect in PLC.

➢ Input circuits
Photo-electricity coupling is used to insulate between primary load circuit and secondary circuit. The secondary circuit with C-R filter is to avoid wrong operation caused by vibration of input contacts or noise along with input signal. For above-mentioned reasons, if input ON→OFF，OFF→ON, the response time delays about 6ms in PLC. There is a digital filter inside the input terminal.

➢ Input sensitivity
The PLC input current is DC24V 7mA, but to act correctly, the current should be above 4.5mA when input is ON and under 1.5mA when input is OFF.

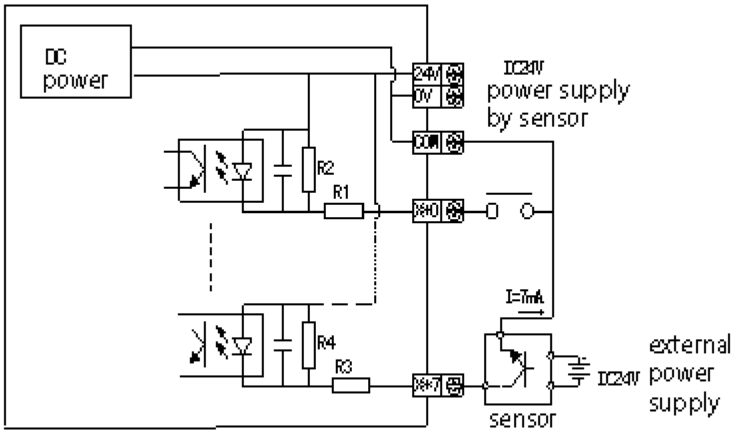
- PNP mode



- Input terminals
  When connect input terminals and terminal COM with DC24V contact or NPN open collector transistor, if input is ON, LED lamp will light which indicates input is ON. There are many input terminals COM to connect in PLC.
- Input circuits
  Photo-electricity coupling is used to insulate between primary load circuit and secondary circuit. The secondary circuit with C-R filter is to avoid wrong operation caused by vibration of input contacts or noise along with input signal. For above-mentioned reasons, if input ON→OFF， OFF→ON, the response time delays about 10ms in PLC. There is a digital filter inside the input terminal.
- Input sensitivity
- The PLC input current is DC24V 7mA, but to act correctly, the current should be above 4.5mA when input is ON and under 1.5mA when input is OFF.

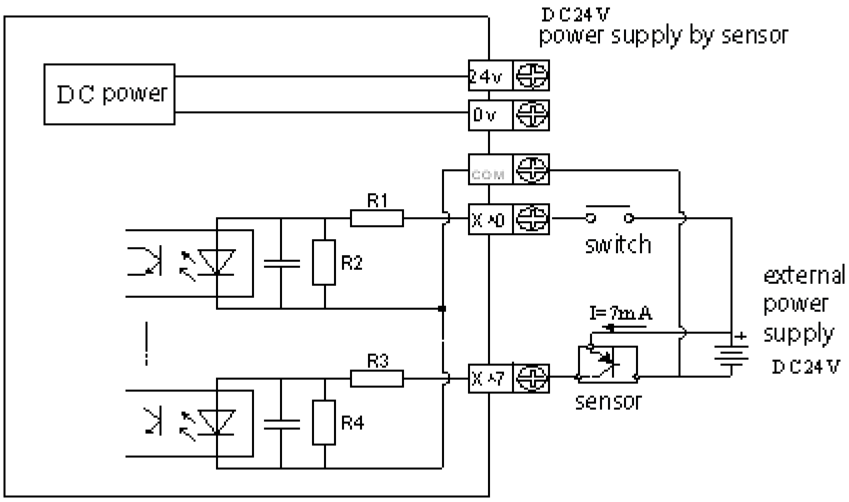
2	**External circuit used by sensors**

- NPN mode
  XD series PLC input current is supplied by its interior 24V power, so if use exterior power to drive sensor like photo electricity switch, the exterior power should be DC24V±4V, please use NPN open collector type for sensor’s output transistor.



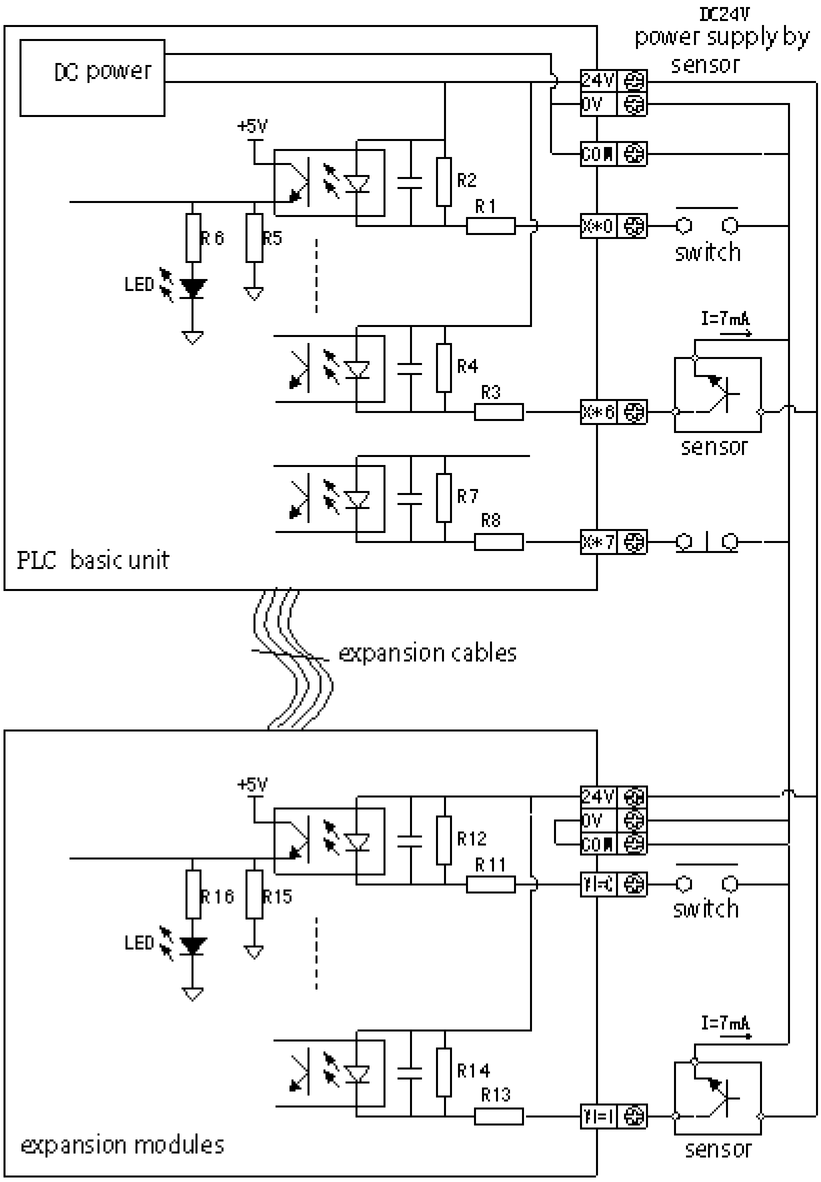
- PNP mode

  XD series PLC input current is supplied by its interior 24V power, so if use exterior power to drive sensor like photo electricity switch, the exterior power should be DC24V±4V, please use PNP open collector type for sensor's output transistor.

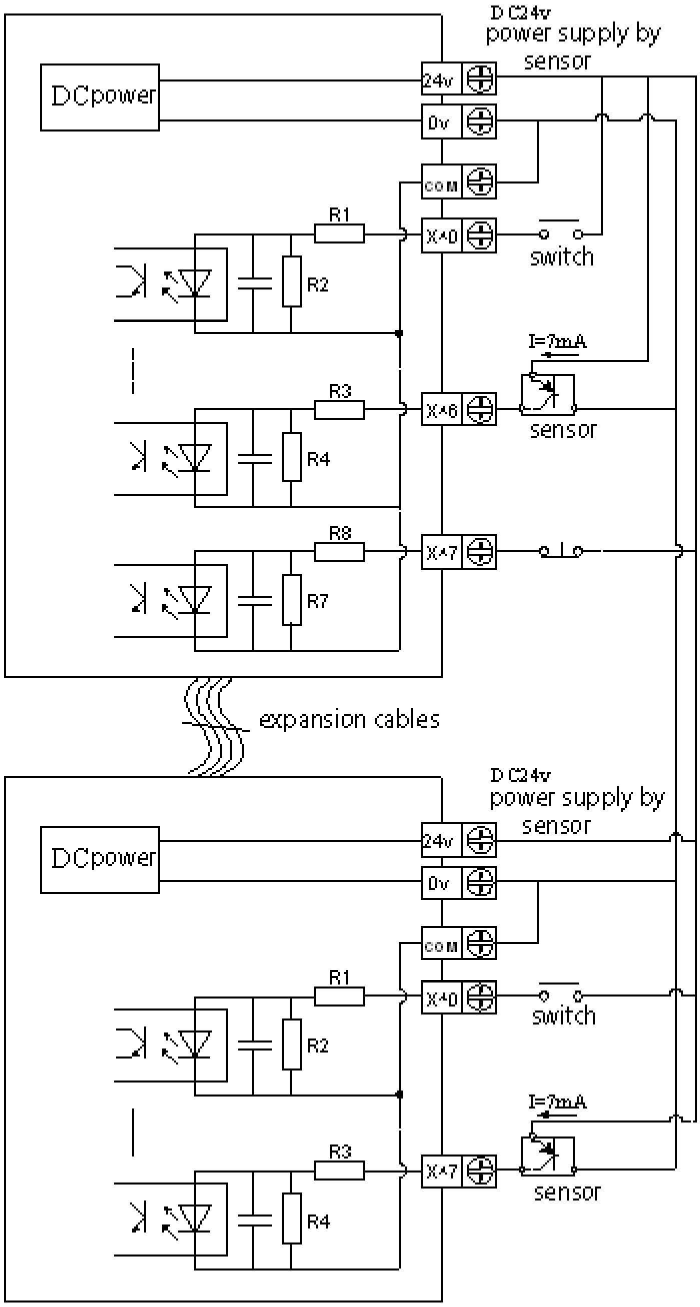


3	Input Wiring

- NPN mode


DC24V
power supply by sensor
DC power
24V
0V
COM
+5V
R2
R1
X*0
switch
R6
R5
LED
I=7mA
R4
R3
X*6
sensor
R7
R8
X*7
PLC basic unit
expansion cables
+5V
24V
0V
COM
R12
R11
switch
R16
R15
LED
I=7mA
R14
R13
sensor
expansion modules

- PNP mode



DC24v
power supply by
sensor
DCpower
24v
0v
COM
R1
X^0
switch
R2
I=7mA
R3
X^6
sensor
R4
R8
X^7
R7
expansion cables
DC24v
power supply by
sensor
DCpower
24v
0v
COM
R1
X^0
switch
R2
I=7mA
R3
X^7
sensor
R4

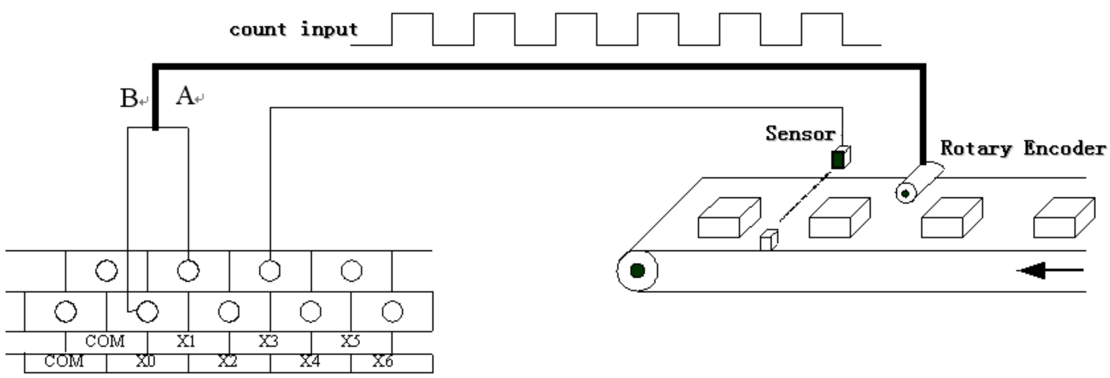
5-3. High Speed Counter Input

XD/XL series PLC support high speed count function which is irrelevant with the scan cycle and can test high speed input signal of measuring sensors and rotary encoders etc by selecting different counter, max measuring frequency can be up to 80KHz.

Note:

(1) If PLC input is NPN type, please select NPN and DC24V collector open output encoder. If PLC input is PNP type, please select PNP and DC24V collector open output encoder.

(2) When the input frequency is above 25Hz, please use high speed counter.

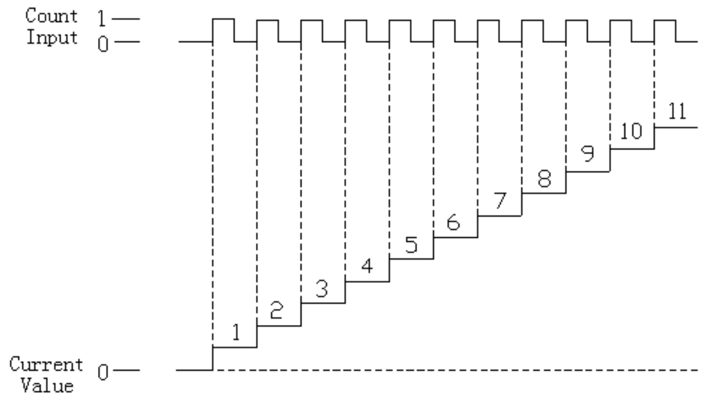


5-3-1. Counting mode

XD/XL series HSC function has two counting modes: Increment mode and AB-phase mode.

1	**Increment mode**

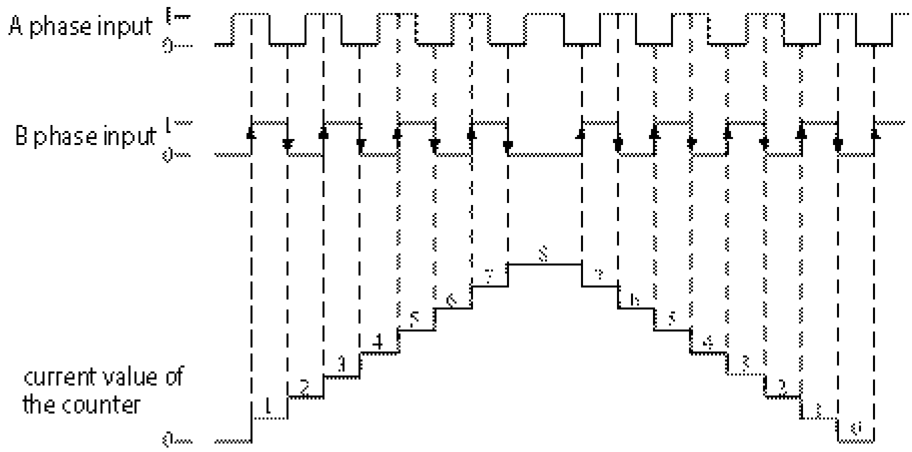
Under this mode, if counting input pulse signal, the counting value will increase one along with the rising edge of every pulse signal.



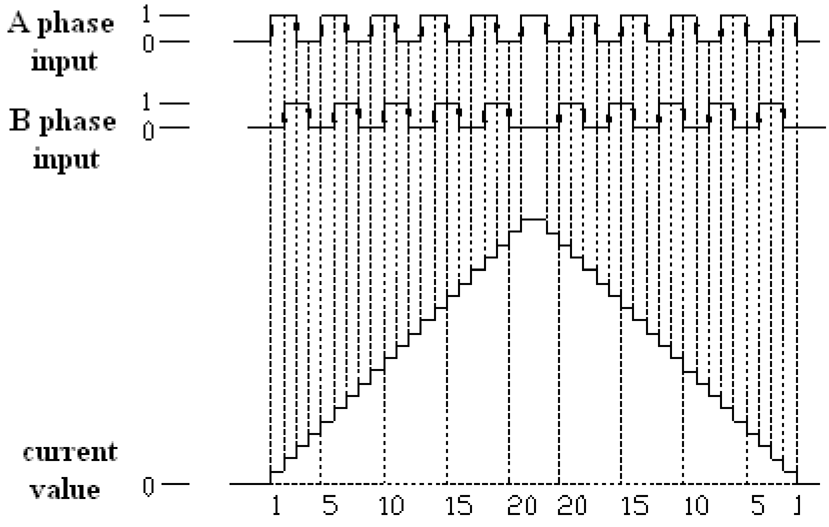
2	AB-phase

In this mode, the HSC value increase or decrease according to the two differential signal (A phase or B phase). According to the times number, the mode still can be divided to two modes (two-time frequency mode and four-time frequency mode). The default mode is four-time frequency mode.

Two–time Frequency Mode



Four–time Frequency Mode



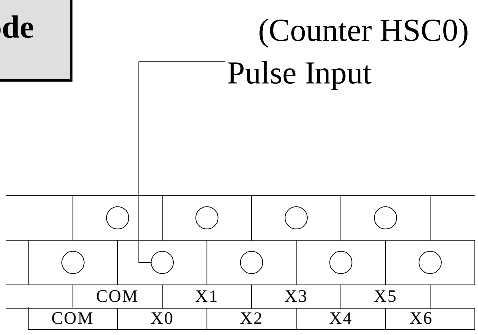
5-3-2. High Speed Counting Range

The HSC's counting range is: K-2,147,483,648 ~ K+2,147,483,647. If the counting value exceeds this range, up-flow or down-flow appears.
The up-flow means the counting value jumps from K+2,147,483,647 to K-2,147,483,648 and then continue to count. The down-flow means the counting value jumps from K-2,147,483,648 to K+2,147,483,647 and then continue to count.

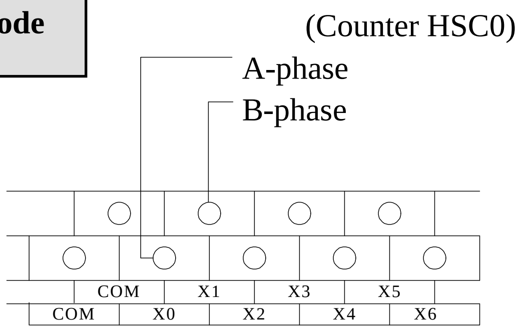
5-3-3. The Input Wiring Of HSC

For input terminal wiring of pulse counting, it differs according to PLC types and counting modes. Some typical wiring methods are as below (take XD3-32 PLC as an example):

1	Increment Mode



2	AB-phase Mode



5-3-4. Input Terminals Assignment

1. High Speed Counters assignment of XD series PLC:

PLC model		High speed counter channels	
		Increment mode	AB-phase mode
XD1	16/32	0	0
XD2/XD3	16/24/32/48/60	3	3
XD5	16/24/32/48/60	3	3
	24T4/32T4/48T4/60T4	4	4
	48T6/60T6	6	6
	60T10	10	10
XDM	24/32/48/60 4-axis	4	4
	60 points 10-axis	10	10
XDC	24/32/48/60	4	4
XD5E	30T4	4	4
	60T10	10	10
XDME	60T10	10	10
XL1	16	0	0
XL3	16	3	3
XL5	32	4	4
XL5E	32T4	4	4

2. Input Terminals definition of HSC:

Each letter's description:

U	A	B	Z
Counter's pulse input	A-phase input	B-phase input	Z-phase pulse capture

Normally, the input frequency of terminal X0, X1can reach 80KHz and 50KHz separately under single-phase and AB-phase mode; while other input terminals highest frequency can reach 10KHz under single-phase and 5KHz under AB phase mode. If X input terminals are not used as high speed input port, they can be used as common input terminals. Frequency times in the table: '2' stands for fixed 2 times frequency, ‘4’ stands for fixed 4 times frequency, ‘2/4’ stands for 2 or 4 times frequency adjustable. The detailed port assignment is shown as below:

XD2-16												
	Increment Mode							AB phase mode				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Highest frequency	10K	10K	10K					5K	5K	5K		
4 times frequency								2/4	2/4	2/4		
Counter interruption	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		

XD2-24/32, XD3-16/24/32, XD5-16/24/32, XL3-16												
	Increment Mode							AB phase mode				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Highest frequency	80K	10K	10K					50K	5K	5K		
4 times frequency								2/4	2/4	2/4		
Counter interruption	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		
X011												

XD2-48/60, XD3-48/60, XD5-48/60												
	Increment Mode							AB phase mode				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Highest frequency	80K	80K	10K					50K	50K	5K		
4 times frequency								2/4	2/4	2/4		
Counter interruption	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		
X011												

XD5-24T4/32T4/48T4/60T4, XD5E-30T4, XDM-24T4/32T4/60T4/60T4L, XDC-24/32/48/60T XL5-32T4, XL5E-32T4, XLME-32T4												
	Increment Mode						AB phase mode					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Highest frequency	80K	80K	80K	80K			50K	50K	50K	50K		
4 times frequency							2/4	2/4	2/4	2/4		
Counter interruption	√	√	√	√			√	√	√	√		
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		

X014												
X015												
X016												
X017												
X020												
X021												

XD5-48T6/60T6												
	Increment Mode						AB phase mode					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Highest frequency	80K	80K	80K	80K	80K	80K	50K	50K	50K	50K	50K	50K
4 times frequency							2/4	2/4	2/4	2/4	2/4	2/4
Counter interruption	√	√	√	√	√	√	√	√	√	√	√	√
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		
X014					U						A	
X015											B	
X016											Z	
X017						U						A
X020												B
X021												Z

XD5-60T10, XDM-60T10, XD5E-60T10, XDME-60T10												
	Increment Mode											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Highest frequency	80K	80K	80K	80K	80K	80K	80K	80K	80K	80K		
4 times frequency												
Counter interruption	√	√	√	√	√	√	√	√	√	√		
X000	U											
X001												
X002												
X003		U										
X004												
X005												
X006			U									
X007												
X010												
X011				U								
X012												
X013												
X014					U							
X015												
X016												
X017						U						
X020												
X021												
X022							U					
X023												
X024												
X025								U				
X026												
X027												
X030									U			
X031												
X032												
X033										U		
X034												

XD5-60T10, XDM-60T10, XD5E-60T10, XDME-60T10												
	AB phase mode											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Highest frequency	50K	50K	50K	50K	50K	50K	50K	50K	50K	50K		
4 times frequency	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4		
Counter interruption	√	√	√	√	√	√	√	√	√	√		
X000	A											
X001	B											
X002	Z											
X003		A										
X004		B										
X005		Z										
X006			A									
X007			B									
X010			Z									
X011				A								
X012				B								
X013				Z								
X014					A							
X015					B							
X016					Z							
X017						A						
X020						B						
X021						Z						
X022							A					
X023							B					
X024							Z					
X025								A				
X026								B				
X027								Z				
X030									A			
X031									B			
X032									Z			
X033										A		
X034										B		
X035										Z		

5-3-5. AB Phase Counter's Frequency Multiplication Setting

To AB phase counter, users can modify the value in FLASH data registers SFD321, SFD322, SFD323……SFD330 to set the frequency multiplication value. When the value is 1, it is 1 time frequency; when the value is 4, it is 4 times frequency.

Register	Function	Setting value	Content
SFD320	Frequency Multiplication of HSC0	2	2 times
		4	4 times
SFD321	Frequency Multiplication of HSC2	2	2 times
		2	4 times
SFD322	Frequency Multiplication of HSC4	2	2 times
		2	4 times
SFD323	Frequency Multiplication of HSC6	2	2 times
		4	4 times
SFD324	Frequency Multiplication of HSC8	2	2 times
		4	4 times
SFD325	Frequency Multiplication of HSC10	2	2 times
		4	4 times
SFD326	Frequency Multiplication of HSC12	2	2 times
		4	4 times
SFD327	Frequency Multiplication of HSC14	2	2 times
		4	4 times
SFD328	Frequency Multiplication of HSC16	2	2 times
		4	4 times
SFD329	Frequency Multiplication of HSC18	2	2 times
		4	4 times

※1: More about high speed counter application, please refer to XD/XL series PLC users' manual【Instruction】.

※2: To some special models, only one axis can be set as 2 times frequency or 4 times frequency, the other two axis are separately 2 times frequency and 4 times frequency.

※3: after setting the SFD register, please restart the high speed counter (cut off the trigger condition and turn on again) to make the setting effective.

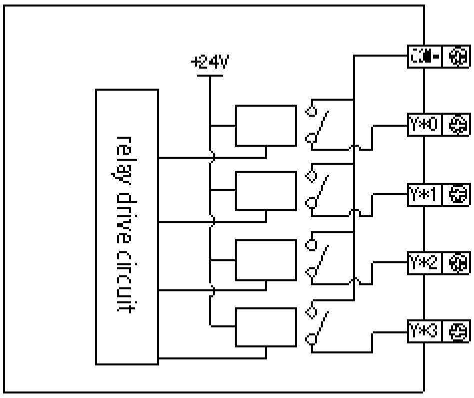
6 Output Specification and Wiring Methods

In this chapter we mainly introduce the output specification and external wiring methods of XD/XL series PLC. The connection methods differ due to different models; the main difference is the terminals' arrangement. For each model's terminals arrangement, please refer to chapter 2-3;

6-1. Output Specification

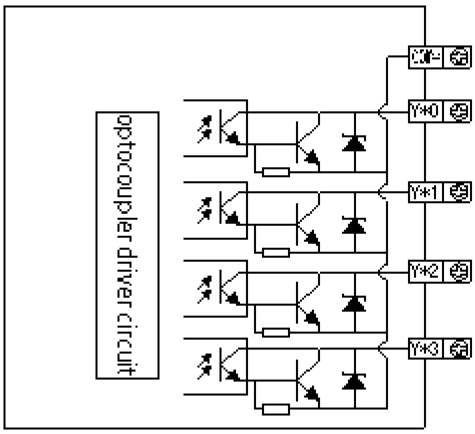
1 Relay Output

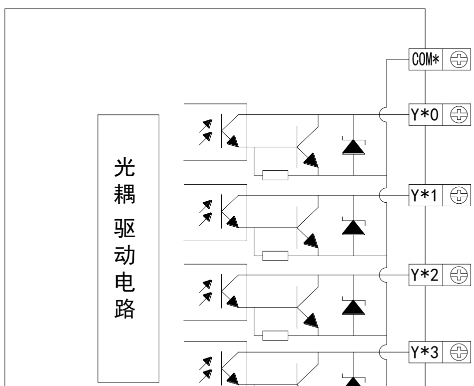
External power		Below AC250V, DC30V
Circuit insulation		Mechanical insulation
Action indicator		LED
Max load	Resistant load	3A
	Inductive load	80VA
	Lamp load	100W
Mini load		DC5V 2mA
Response time	OFF→ ON	10ms
	ON→ OFF	10ms



2 Normal Transistor Output

External power		Below DC5~30V
Circuit insulation		Light coupling insulation
Action indicator		LED
Max load	Resistant load	0.3A
	Inductive load	8W/DC24V
	Lamp load	1.5W/DC24V
Mini load		DC5V 2mA
Respon se time	OFF→ ON	Below 0.2ms
	ON→ OFF	Below 0.2ms





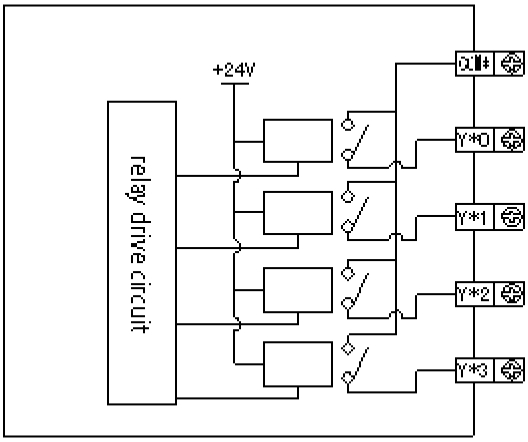
3	High Speed Pulse Output

Model	RT or T				
High Speed Pulse Output Terminal External Power Supply	-	Y0, Y1	Y0~Y3	Y0~Y5	Y0~Y11
	XD1 XL1	General models	XD5-24T4 XD5-32T4 XDM-60T4 XDM-60T4L XD5E-30T4 XL5-32T4 XL5E-32T4 XLME-32T4	XD5-48T6 XD5-60T6	XDM-60T10 XD5E-60T10 XDME-60T10
Action Indicator	Below DC5~30V				
Maximum Current	LED indicator				
Max output frequency of pulse	50mA				
High Speed Pulse Output Terminal	100KHz				

Note:
When using high-speed pulse output function, the PLC can output 100KHz ~ 200KHz pulse, but it can not guarantee the normal operation of all servos. Please connect about 500 ohms of resistance between the output and 24V power supply.

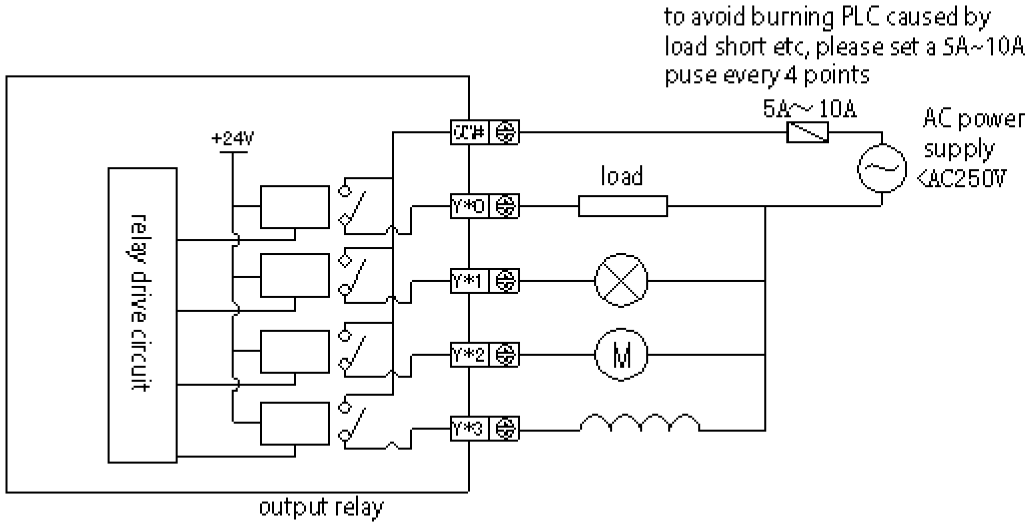
6-2. Relay Output Type

1 Relay Output Circuit



- **Output terminals**
  Relay output type has 2~4 public terminals. So each public-terminal unit can drive power system with different voltages (E.g.: AC200V, AC100V, DC24V etc.) load.
- **Circuit's insulation**
  Between the relay output coils and contacts, PLC's interior circuits and exterior load circuits are electrical insulating. Besides, each public terminal and block are separate from each other.
- **Action display**
  LED lamp lights when output relays' coils energize, output contacts are ON.
- **Response time**
  From the output relay energize (or cut off) to output contact ON (or OFF), the response time is about 10ms.
- **Output current**
  The output current that current and voltage below AC250Vcan drive the load made up of resistance is 3A per point, inductive load below 80VA (AC100V or AC200V) and lamp load below100W (AC100V or AC200V).
- **Open circuit's leak current**
  When output contact is OFF, there will be no leak current and can directly drive Ne lamp etc.
- **The life of relay output contacts**
  Standard life of AC inductive load such as contactor, electromagnetic valve: according to company's useful life test, about 500 thousand times for 20VA load; about 300 thousand times for 35VA; about 100 thousand for 80VA. But if the load parallel connect with surge absorber, the useful life will greatly improve.

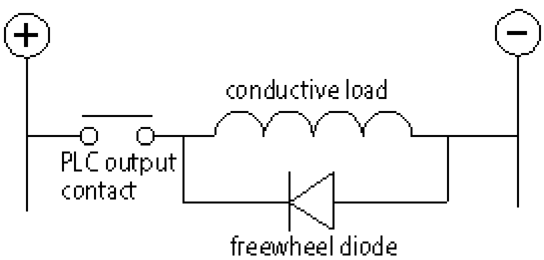
2	Output Connection Example



3	Constitution of output circuit

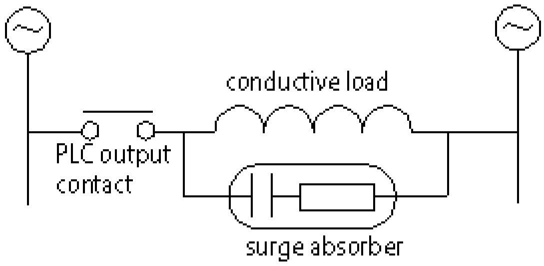
- For DC inductive load, please parallel connect with freewheel diode. Otherwise, contactor useful life will greatly decrease. Please select freewheel diode that can stand inverse voltage over 5~10 times of load voltage and forward current over load current.
- Parallel connection AC inductive load with surge absorber will decrease noise and increase service life of output delay.

4	DC Load



Note: the freewheeling diode is EN4007.

5	**AC Load**



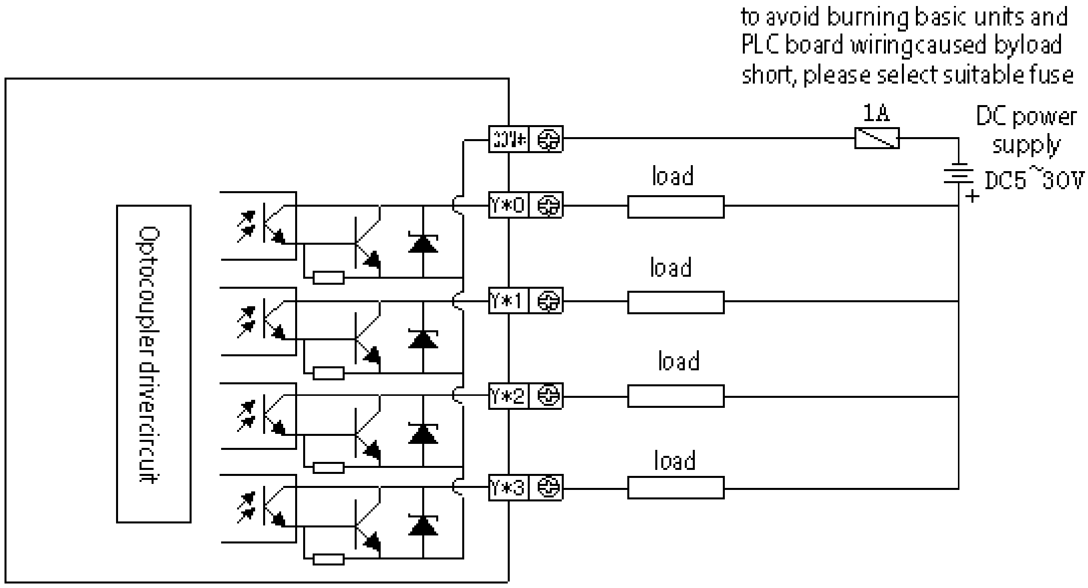
Note: the surge absorber is R=200Ω 2W, C=0.022uF 250VAC.

6-3. Transistor Output Type

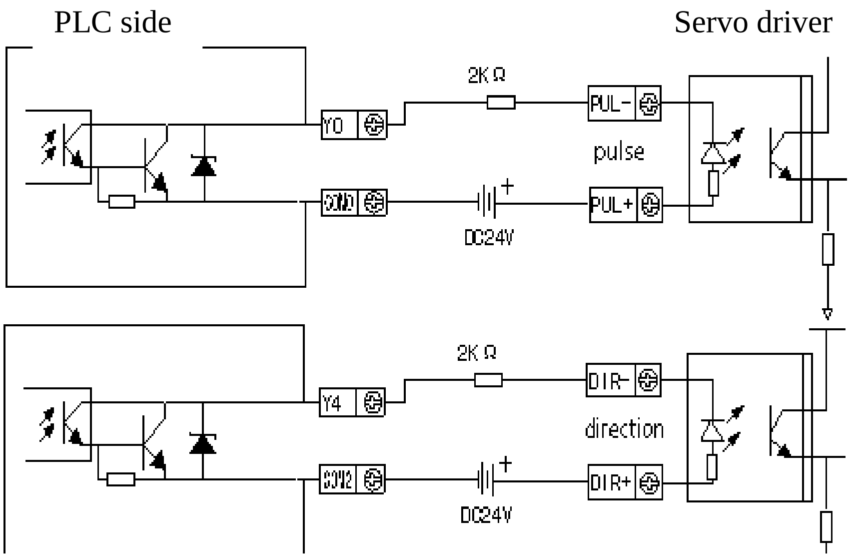
Transistor (NPN) output can support high speed pulse output and normal transistor two types.

1	**Normal Transistor Output**

- Output Terminals
  There are 1~4 COM outputs of CPU unit transistor outputs.
- External Power Supply
  Please use DC5~30V power supply to drive the load.
- Circuit Isolation
  Inside PLC, we use photoelectric couplers to isolate between internal circuits and output transistors; besides, the COM terminal blocks are separate from each other.
- Action Display
  When photoelectric couplers drive, LED will be ON and the output transistors will be ON.
- Response Time
  The time interval that PLC from photoelectric couplers energizing (or cutting) to transistor ON (or OFF) is below 0.2ms.
- Output current
  The current it outputs is 0.3A per point. But limited by the temperature rising, every 4 points current add up to 0.5A.
- Open circuit current
  Below 0.1mA



E.g.: Below is the connection of RT/T type PLC and servo driver diagram:



(Make sure the driver's photoelectric coupling input terminal has 8~15mA reliable current)

7 Run, Debug, Maintenance

In this chapter, we introduce XD/XL PLC process of programming and using, which includes PLC run, debug and daily maintenance etc.

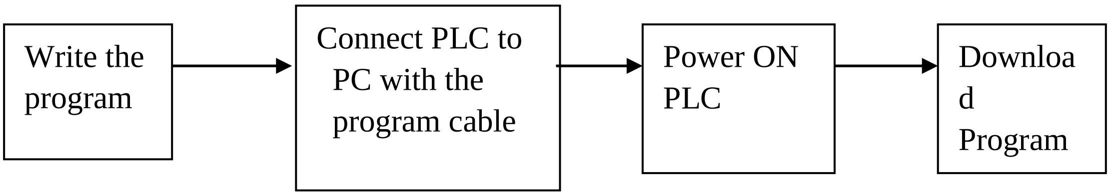
7-1. Run and Debug

1 Check the Products

Please check if the input/output terminals are correct and if there is any component missed when the users get the products. Generally, you can power on the PLC directly at this time and if products are normal, the PWR and RUN indicators will be ON.

2 Write and Download the Program

After confirming the products, write the program for PLC in your PC, and then download the program to PLC. The general operation steps are listed below:



※1: Please link the download cable before you power on the PLC. Otherwise, the COM port may be burned out! BD card and expansion connection is the same operation.

3 Debug the Products

In ideal condition, PLC is in running mode. But if you find some mistakes in the program and need modify, you should write program to the running PLC again.

- Connect PLC to PC with the program cable;
- Upload the program in PLC;
- Modify the uploaded program; and the modified program is suggested to save backup;
- Pause the running of PLC, and download the modified program to PLC;
- Use ladder monitor, free monitor to etc monitor PLC
- If the program still can't fulfill your requirement, you can go on modify it and download to PLC.

4 LED on PLC

- When PLC is running correctly, the **PWR** and **RUN** LED should be ON;
- If **ERR** LED is ON, it indicates that PLC running is in error, please correct the program in time.
- If **PWR** LED is OFF, it indicates that the power supply is in error, please check your wiring.

7-2. Daily Maintenance

1 Regular Check on Products

Even the PLC has certain anti-interference ability and strong stability, you should check the PLC regularly.

The check items include:

- Check if the input/output terminals, power supply terminals are loosen;
- Check if the ports are correct;
- Check if the PWR LED, I/O LED can be ON;
- Clear the dusts on PLC to avoid the dusts falling into PLC
- Manage to make PLC running and storage environment fits the standards described in chapter 2-1-1.

2 About the battery

The PLC can keep working if there is not component that could short its service life. But if the PLC supports clock function, its battery should be changed regularly.

- Battery service life normally is 3~5 years.
- Please change the battery once you find the battery power down.
- Please power the PLC on immediately after changing the battery. Otherwise, the battery power may run out.

3 Abandon

Abandon as industrial wast.

8 Switch between Soft Components

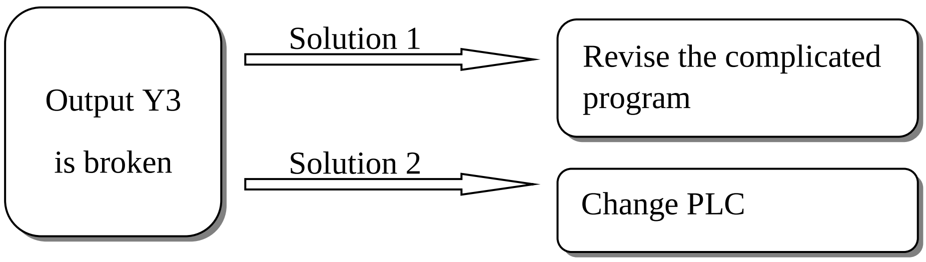
This chapter focuses on a special function of XD/XL series PLC, switch between soft components. This special function simplifies the PLC daily maintenance greatly. To the maintenance person, they will not bother any more if the terminals are damaged.

8-1. Function Summary

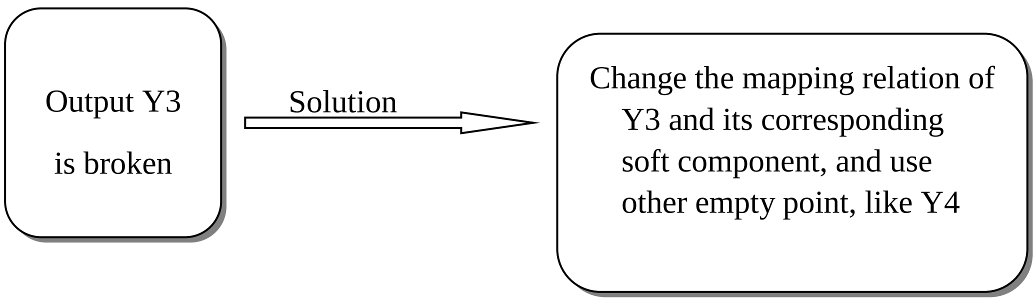
When the internal lighting coupling, relays or transistor are damaged, the corresponding input/output terminals will be out of use. Users either revise the program or ask the manufactures for help, which is very troublesome and affects the users' normal work schedule.

The new type PLC developed independently by Xinje can break the one-to-one correspondence, users only need to change the soft component's value by HMI, then the corresponding terminal will activate.

Before(Complicated and not effective)



Now (Simple, fast and effective)



8-2. Operation Method

It no needs to revise the program when we change the damaged input/output point mapping relation and replace the damaged point. In PLC special registers, we allocate certain address section for users to change the mapping relation. Users just need to find and revise the damaged input/output mapping register, and replace the value in this special register with value of replaced input/output.

Method 1: modify the FD register, below is the table for modifying the input/output points' mapping ID:

Table1 Mapping relation of the input and soft component

ID	Function	Description
SFD10	I00 correspond to X**	0 of input corresponds to the number of X**
SFD11	I01 correspond to X**	
SFD12	I02 correspond to X**	
……	……	
SFD87	I77 correspond to X**	Default is 77 (octal number)

Table2 mapping relation of the output and soft component

ID	Function	Description
SFD110	O00 correspond to Y**	0 of output corresponds to the number of Y**
SFD111	O01 correspond to Y**	
SFD112	O02 correspond to Y**	
……	……	
SFD187	O77 correspond to Y**	Default is 77 (octal number)

As show in the table above, the default value in SFD10 is 0. If we replace it with value '7',then all X0 in the program will correspond to external input X7. But meantime you should replace the value in SFD17 with 0, to realize exchange. Then original X0 will correspond to X7, and original X7 will correspond to external input X0.

※1: After changing the mapping relation, please power on PLC again.

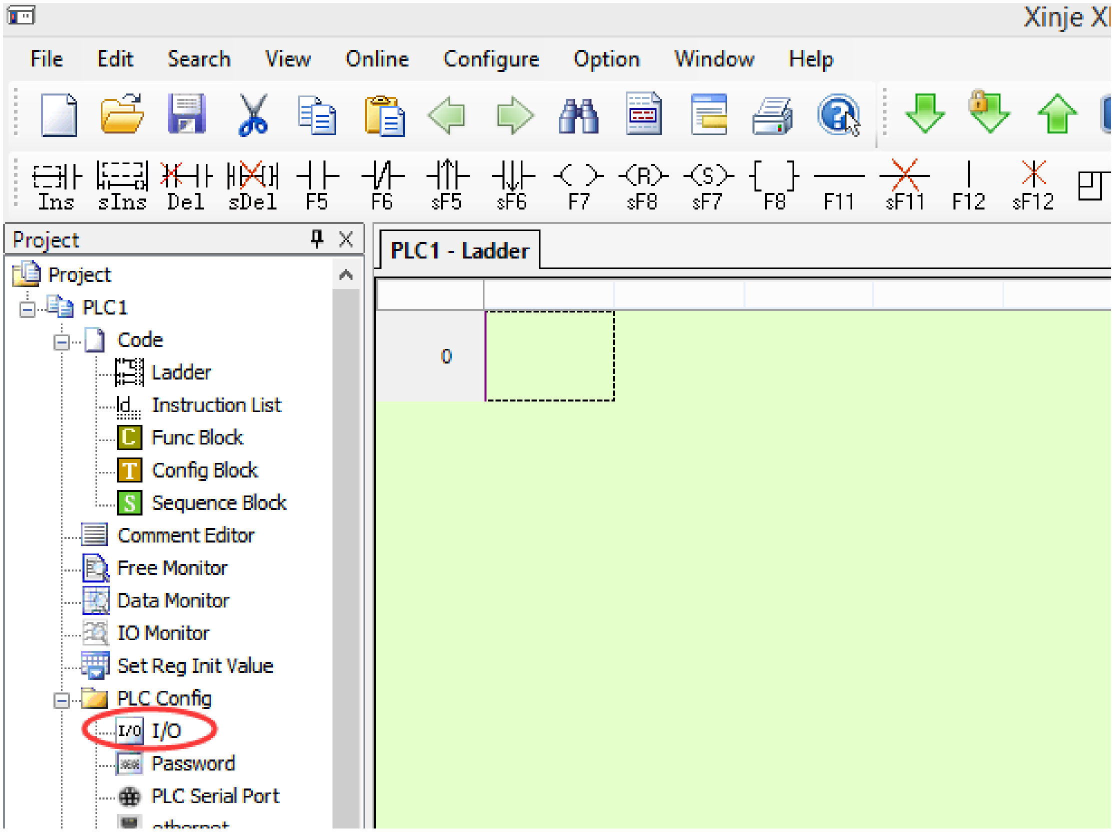
※2: When change the mapping relation, please pay attention, input/output data is octal number while ID is decimal number.

※3: Exchange the mapping relation when change. i.e. if modify X0 ID to be 5, make sure to change X5 ID to be 0;

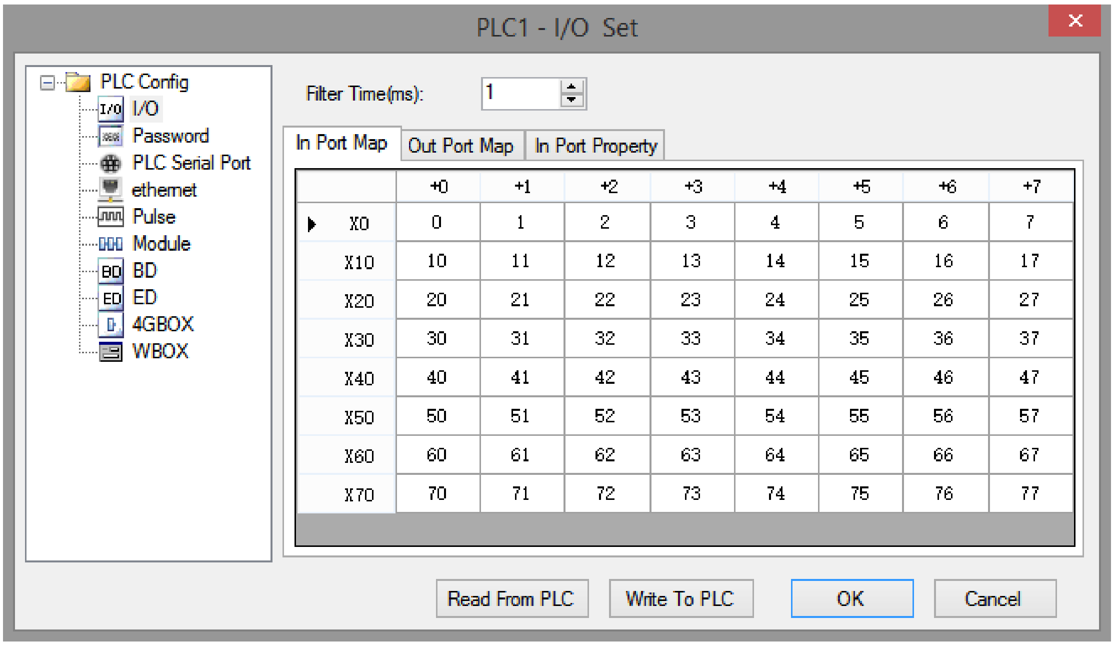
※4: Mapping relation, one terminal corresponds to one soft component.

※5: Users can modify the SFD value in the software, please see method 2.

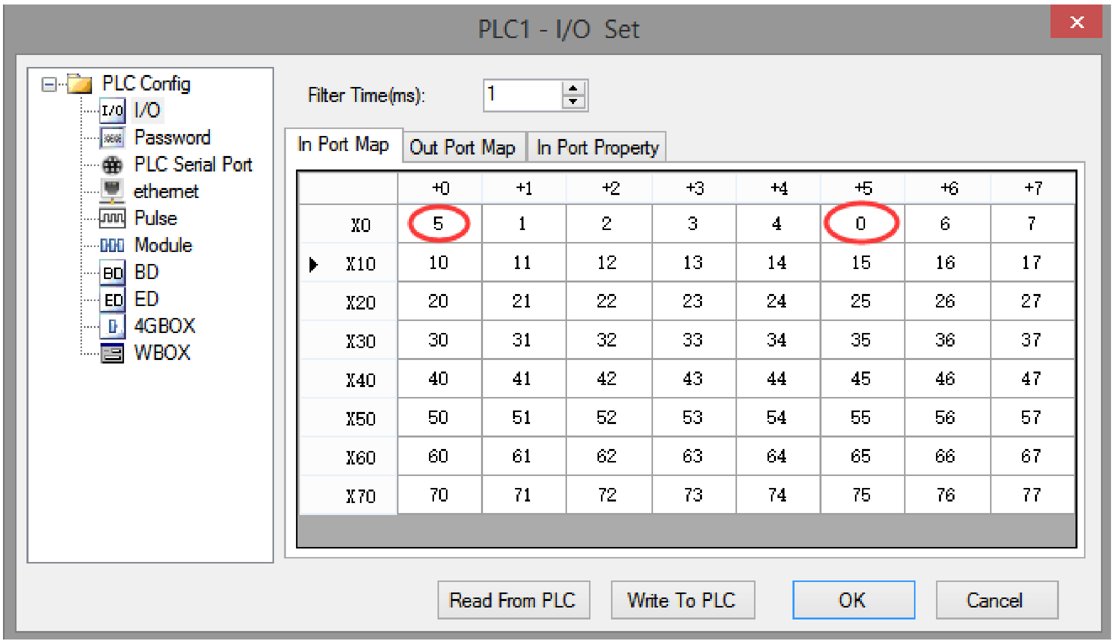
Method 2: modify in the software directly. Click the project bar/PLC config/I/O.



Change it in below window:



For example, it needs to switch X0 and X5, please change the mapping value of X0 to 5, X5 to 0.

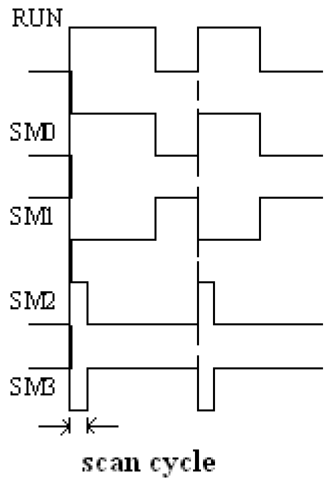

PLC1 - I/O Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
WBOX
Filter Time(ms):
1
In Port Map
Out Port Map
In Port Property
+0 +1 +2 +3 +4 +5 +6 +7
X0 5 1 2 3 4 0 6 7
X10 10 11 12 13 14 15 16 17
X20 20 21 22 23 24 25 26 27
X30 30 31 32 33 34 35 36 37
X40 40 41 42 43 44 45 46 47
X50 50 51 52 53 54 55 56 57
X60 60 61 62 63 64 65 66 67
X70 70 71 72 73 74 75 76 77
Read From PLC
Write To PLC
OK
Cancel

Appendix 1 Special Soft Element Schedules

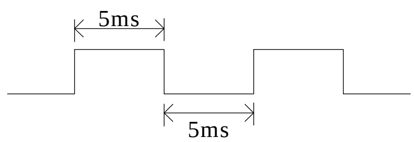
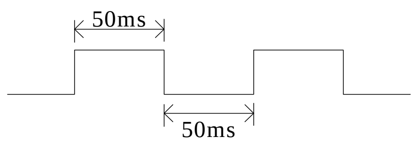
Appendix 1 mainly introduces the functions of XD/XL series PLC special soft element, data register, FlashROM and the address distribution of expansions for users to search.

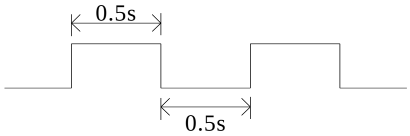
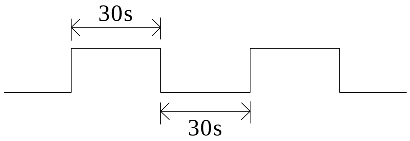
Appendix 1-1. Special Auxiliary Relay Schedule

Initial Status（SM0-SM7）

ID	Function	Description	
SM000	Coil ON when running		SM000 keeps ON when PLC running
SM001	Coil OFF when running		SM001 keeps OFF when PLC running
SM002	Initial positive pulse coil		SM002 is ON in first scan cycle
SM003	Initial negative pulse coil		SM003 is OFF in first scan cycle
SM004	PLC running error	When SM4 sets ON, it indicates that there is an error in the operation of PLC. (Firmware version V3.4.5 and above supports this function by PLC)	
SM005	Battery low alarm coil	When the battery voltage is less than 2.5V, SM5 will put ON (at this time, please replace the battery as soon as possible, otherwise the data will not be maintained)	
SM007	Power-off memory data error		

Clock（SM11-SM14）

ID	Function	Description
SM011	10ms frequency cycle	
SM012	100ms frequency cycle	

SM013	1s frequency cycle	
SM014	1min frequency cycle	

Mark（SM20-SM22）

ID	Function	Description
SM020	Zero bit	SM020 is ON when plus/minus operation result is 0
SM021	Borrow bit	SM021 is ON when minus operation overflows
SM022	Carry bit	SM022 is ON when plus operation overflows

PC Mode（SM32-SM34）

ID	Function	Description
SM032	Retentive register reset	When SM032 is ON, ON/OFF mapping memory of HM、HS and current values of HT、HC、HD will be reset.
SM033	Clear user's program	When SM033 is ON, all PLC user's program will be cleared.
SM034	All output forbidden	When SM034 is ON, all PLC external contacts will be set OFF.

Stepping Ladder

ID	Function	Description
SM040	The process is running	Set ON when the process is running

Interruption ban（SM50-SM90）

ID	Address	Function	Description
SM050	I0000/I0001	Forbid input interruption 0	After executing EI instruction, the input interruption couldn’t act independently when M acts, even if the interruption is allowed. E.g.： when SM050 is ON, I0000/I0001 is forbidden.
SM051	I0100/I0101	Forbid input interruption 1	
SM052	I0200/I0201	Forbid input interruption 2	
SM053	I0300/I0301	Forbid input interruption 3	
SM054	I0400/I0401	Forbid input interruption 4	
……	……	……	
SM069	I1900/I1901	Forbid input interruption 19	
SM070	I40**	Forbid timing interruption 0	After executing EI instruction, the timing interruption couldn’t act independently when M acts, even if the interruption is allowed.
SM071	I41**	Forbid timing interruption 1	
SM072	I42**	Forbid timing interruption 2	
SM073	I43**	Forbid timing interruption 3	
SM074	I44**	Forbid timing interruption 4	
……	……	……	
SM089	I59**	Forbid timing interruption 19	
SM090		Forbid all interruptions	Forbid all interruptions

High Speed Ring Counter（SM99）

address	Function	Note
SM099	High Speed Ring Counting enable	SM99 set ON， SD99 add one per 0.1ms, cycle between 0 and 32767

High speed count complete（SM100-SM109）

Address	Function	Note
SM100	HSC0 count complete flag（100 segments）	
SM101	HSC2 count complete flag（100 segments）	
SM102	HSC4 count complete flag（100 segments）	
SM103	HSC6 count complete flag（100 segments）	
SM104	HSC8 count complete flag（100 segments）	
SM105	HSC10 count complete flag（100 segments）	
SM106	HSC12 count complete flag（100 segments）	
SM107	HSC14 count complete flag（100 segments）	
SM108	HSC16 count complete flag（100 segments）	

SM109	HSC18 count complete flag（100 segments）	

High speed counter direction（SM110-SM119）

Address	Function	Note
SM110	HSC0 direction flag	
SM111	HSC2 direction flag	
SM112	HSC4 direction flag	
SM113	HSC6 direction flag	
SM114	HSC8 direction flag	
SM115	HSC10 direction flag	
SM116	HSC12 direction flag	
SM117	HSC14 direction flag	
SM118	HSC16 direction flag	
SM119	HSC18 direction flag	

High speed counter error（SM120-SM129）

address	Function	Note
SM120	HSC0 error flag	
SM121	HSC2 error flag	
SM122	HSC4 error flag	
SM123	HSC6 error flag	
SM124	HSC8 error flag	
SM125	HSC10 error flag	
SM126	HSC12 error flag	
SM127	HSC14 error flag	
SM128	HSC16 error flag	
SM129	HSC18 error flag	

Communication（SM140-SM193）

	Address	Function	Note
Serial port 0	SM140	Modbus instruction execution flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM141	X-NET instruction execution flag	When the instruction starts to execute, set ON When execution is complete, set OFF

	SM142	Free format communication sending flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM143	Free format communication receive complete flag	When receiving a frame of data or receiving data timeout, set ON. Require user program to set OFF
Serial port 1	SM150	Modbus instruction execution flag	Same to SM140
	SM151	X-NET instruction execution flag	Same to SM141
	SM152	Free format communication sending flag	Same to SM142
	SM153	Free format communication receive complete flag	Same to SM143
Serial port 2	SM160	Modbus instruction execution flag	Same to SM140
	SM161	X-NET instruction execution flag	Same to SM141
	SM162	Free format communication sending flag	Same to SM142
	SM163	Free format communication receive complete flag	Same to SM143
Serial port 3	SM170	Modbus instruction execution flag	Same to SM140
	SM171	X-NET instruction execution flag	Same to SM141
	SM172	Free format communication sending flag	Same to SM142
	SM173	Free format communication receive complete flag	Same to SM143
Serial port 4	SM180	Modbus instruction execution flag	Same to SM140
	SM181	X-NET instruction execution flag	Same to SM141
	SM182	Free format communication sending flag	Same to SM142
	SM183	Free format communication receive complete flag	Same to SM143
Serial port 5	SM190	Modbus instruction execution flag	Same to SM140
	SM191	X-NET instruction execution	Same to SM141

		flag	
	SM192	Free format communication sending flag	Same to SM142
	SM193	Free format communication receive complete flag	Same to SM143

Sequence Function BLOCK（SM240-SM399）

ID	Function	Description
SM300	BLOCK1 running flag	SM300 will be ON when block1 is running
SM301	BLOCK2 running flag	SM301 will be ON when block2 is running
SM302	BLOCK3 running flag	SM302 will be ON when block3 is running
SM303	BLOCK4 running flag	SM303 will be ON when block4 is running
SM304	BLOCK5 running flag	SM304 will be ON when block5 is running
SM305	BLOCK6 running flag	SM305 will be ON when block6 is running
……	……	
SM396	BLOCK97 running flag	SM396 will be ON when block97is running
SM397	BLOCK98 running flag	SM397 will be ON when block98 is running
SM398	BLOCK99 running flag	SM398 will be ON when block99 is running
SM399	BLOCK100 running flag	SM399 will be ON when block100 is running

Error check（SM400-SM413）

ID	Function	Description
SM400	I/O error	ERR LED keeps ON, PLC don not run and output, check when power on

SM401	Expansion module communication error	
SM402	BD communication error	
……		
SM405	No user program	Internal code check wrong
SM406	User program error	Implement code or configuration table check wrong
SM407	SSFD check error	ERR LED keeps ON, PLC don not run and output, check when power on
SM408	Memory error	Can not erase or write Flash
SM409	Calculation error	
SM410	Offset overflow	Offset exceeds soft element range
SM411	FOR-NEXT overflow	Reset when power on or users can also reset by hand.
SM412	Invalid data fill	When offset of register overflows, the return value will be SM372 value
SM413		

Error Message（SM450-SM452）

ID	Function	Description
SM450	System error check	
SM451	Hardfault interrupt flag	
SM452		
SM453	SD card error	
SM454	Power supply is cut off	
……		
SM460	Extension module ID not match	
SM461	BD/ED module ID not match	
SM462	Extension module communication overtime	
SM463	BD/ED module communication overtime	

Expansion Modules, BD Status（SM500）

ID	Function	Description
SM500	Module status read is finished	

Appendix 1-2. Special Data Register Schedule

Battery （SD5~SD7）

ID	Function	Description
SD005	Battery register	It will display 100 when the battery voltage is 3V, if the battery voltaeg is lower than 2.5V, it will display 0, it means please change new battery at once, otherwise the data will lose when PLC power off.
SD007	Power-off memory data error type	

Clock（SD10-SD019）

ID	Function	Description
SD010	Current scan cycle	100us， us is the unit
SD011	Min scan time	100us， us is the unit
SD012	Max scan time	100us， us is the unit
SD013	Second (clock)	0~59 (BCD code)
SD014	Minute (clock)	0~59 (BCD code)
SD015	Hour (clock)	0~23 (BCD code)
SD016	Day (clock)	0~31 (BCD code)
SD017	Month (clock)	0~12 (BCD code)
SD018	Year (clock)	2000~2099 (BCD code)
SD019	Week (clock)	0(Sunday)~6(Saturday)(BCD code)

Flag （SD020-SD031）

ID	Function	Note
SD020	Model type	
SD021	model（low-8）series（high-8）	
SD022	Compatiable system version（low）system version（high）	
SD023	Compatiable model version（low）model version（high）	
SD024	Model info	
SD025	Model info	
SD026	Model info	
SD027	Model info	
SD028	Suitable software version	
SD029	Suitable software version	
SD030	Suitable software version	
SD031	Suitable software version	

Step ladder（SD040）

ID	Function	Description
SD40	Flag of the executing process S	

High Speed Counting（SD100-SD109）

ID	Function	Description	
SD100	Current segment (No. n segment)		HSC00
SD101	Current segment (No. n segment)		HSC02
SD102	Current segment (No. n segment)		HSC04
SD103	Current segment (No. n segment)		HSC06
SD104	Current segment (No. n segment)		HSC08
SD105	Current segment (No. n segment)		HSC10
SD106	Current segment (No. n segment)		HSC12
SD107	Current segment (No. n segment)		HSC14
SD108	Current segment (No. n segment)		HSC16
SD109	Current segment (No. n segment)		HSC18

High speed counter error（SD120-SD129）

ID	Function	Note
SD120	HSC0 error info	
SD121	HSC2 error info	
SD122	HSC4 error info	
SD123	HSC6 error info	
SD124	HSC8 error info	
SD125	HSC10 error info	
SD126	HSC12 error info	
SD127	HSC14 error info	
SD128	HSC16 error info	
SD129	HSC18 error info	

communication（SD140~SD199）

	ID	Function	Note
Serial port 0	SD140	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD141	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD142	Free format communication send result	0: correct 410: free format send buffer overflow
	SD143	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error

			414: receive overtime 415: no start character 416: no end character
	SD144	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD149		
Serial port 1	SD150	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD151	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD152	Free format communication send result	0: correct 410: free format send buffer overflow
	SD153	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD154	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD159		
	SD160	Modbus read write instruction execution	0: correct 100: receive error

Serial port 2		result	101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD161	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD162	Free format communication send result	0: correct 410: free format send buffer overflow
	SD163	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD164	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD169		
Serial port 3	SD170~SD179		
Serial port 4	SD180~SD189		
Serial port 5	SD190~SD199		

Sequence Function Block（SD300-SD399）

ID	Function	Description
SD300	Executing instruction of BLOCK1	The value will be used when BLOCK monitors
SD301	Executing instruction of BLOCK2	The value will be used when BLOCK monitors
SD302	Executing instruction of BLOCK3	The value will be used when BLOCK monitors
SD303	Executing instruction of BLOCK4	The value will be used when BLOCK monitors
SD304	Executing instruction of BLOCK5	The value will be used when BLOCK monitors
SD305	Executing instruction of BLOCK6	The value will be used when BLOCK monitors
……	……	……
SD396	Executing instruction of BLOCK97	The value will be used when BLOCK monitors
SD397	Executing instruction of BLOCK98	The value will be used when BLOCK monitors
SD398	Executing instruction of BLOCK99	The value will be used when BLOCK monitors
SD399	Executing instruction of BLOCK100	The value will be used when BLOCK monitors

Error Check（SD400-SD413）

ID	Function	Note
SD400		
SD401	Extension module no. of communication error	Means module no.n is error
SD402	BD/ED module no. of communication error	
SD403	FROM/TO error type	
SD404	PID error type	
……		
SD409	Calculation error code	1: divide by 0 error 2: MRST, MSET front operand address less than back operand 3: ENCO, DECO data bits of encoding and

		decoding instructions exceed the limit. 4: BDC code error 7: Radical sign error
SD410	The number of offset register D when offset crosses the boundary	
SD411		
SD412	Invalid data fill value (low 16 bits)	
SD413	Invalid data fill value (high 16 bits)	

Error Check（SD450-SD452）

ID	Function	Description
SD450	1：Watchdog act (Default 200ms) 2：Control block application fail 3：Visit illegal address	
SD451	Hardware error type: 1：Register error 2：Bus error 3：Usage error	
SD452	Hardware error	
SD453	SD card error	
SD454	Power-off time	
SD460	Extension module ID not match	
SD461	BD/ED module ID not match	
SD462	Extension module communication overtime	
SD463	BD/ED module communication overtime	

Expansion Modules, BD Status（SD500-SD516）

ID	Function	Description	
SD500	Module number Expansion modules： #10000～10015 BD：#20000～20001 ED: #30000		
SD501~516	Expansion module, BD /ED status		16 registers

Module info（SD520-SD823）

ID	Function	Explanation	Note
SD520~SD535	Extension module info	Extension module 1	Each extension module, BD, ED occupies 16 registers
……	……	……	
SD760~SD775	Extension module info	Extension module 16	
SD776~SD791	BD module info	BD module 1	
SD792~SD807	BD module info	BD module 2	
SD808~SD823	ED module info	ED module 1	

Expansion Module Error Information

ID	Function	Description	
SD860	Error times of module read		Expansion module 1
SD861	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD862	Error times of module write		
SD863	Error types of module write		
SD864	Error times of module read		Expansion module 2
SD865	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD866	Error times of module write		
SD867	Error types of module write		
……			
SD920	Error times of module read		Expansion module 16
SD921	Error types of module	Module address error.	

	read	Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD922	Error times of module write		
SD923	Error types of module write		
SD924	Error times of module read		BD module 1
SD925	Error types of module read		
SD926	Error times of module write		
SD927	Error types of module write		
SD928	Error times of module read		BD module 2
SD929	Error types of module read		
SD930	Error times of module write		
SD931	Error types of module write		
SD932	Error times of module read		ED module 1
SD933	Error types of module read		
SD934	Error times of module write		
SD935	Error types of module write		

Version info（SD990~SD993）

ID	Function	Explanation	Note
SD990	Firmware version date	Low 16-bit	
SD991	Firmware version compilation date	High 16-bit	

SD992	FPGA version compilation date	Low 16-bit	
SD993	FPGA version compilation date	High 16-bit	

Appendix 1-3. Special Flash Register schedule

Special FLASH data register SFD

*** means it works only after repower on the PLC**

I filtering

ID	Function	Description
SFD0*	Input filter time	
SFD2*	Watchdog run-up time, default value is 200ms	

I Mapping

ID	Function	Description	
SFD10*	I00 corresponds to X**	Input terminal 0 corresponds to X** number	0xFF means terminal bad，0xFE means terminal idle
SFD11*	I01 corresponds to X**		
SFD12*	I02 corresponds to X**		
……	……		
SFD73*	I77 corresponds to X**	Default value is 77（Octonary）	

O Mapping

ID	Function	Description	
SFD74*	O00 corresponds to Y**	Output terminal 0 correspond to Y** number	0xFF means terminal bad，0xFE means terminal idle
		Default value is 0	
……	……		
SFD134*	O77 corresponds to Y**	Default value is 77（Octonary）	

I Attribute

ID	Function	Description	
SFD138*	I00 attribute	Attribute of input terminal 0	0：positive logic others：negative logic
SFD139*	I01 attribute		
……	……		
SFD201*	I77 attribute		

High Speed Counting

ID	Function	Description
SFD320	HSC0 frequency times	2：2 times frequency；4: 4 times frequency(effective at AB phase counting mode)
SFD321	HSC2 frequency times	Ditto
SFD322	HSC4 frequency times	Ditto
SFD323	HSC6 frequency times	Ditto
SFD324	HSC8 frequency times	Ditto
SFD325	HSC10 frequency times	Ditto
SFD326	HSC12 frequency times	Ditto
SFD327	HSC14 frequency times	Ditto
SFD328	HSC16 frequency times	Ditto
SFD329	HSC18 frequency times	Ditto
SFD330	Bit selection of HSC absolute and relative（24 segment）	bit0 corresponds to HSC0，bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：relative 1：absolute
SFD331	Interrupt circulating of 24 segments high speed counting	bit0 corresponds to HSC0，bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：single 1：loop
SFD332	CAM function	bit0 corresponds to HSC0，bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：do not support CAM function 1：support CAM function

Expansion Module Configuration

ID	Function	Explanation
SFD340	Extension module configuration status（#1#2）	Configuration Status of Extension Modules 1 and 2

SFD341	Extension module configuration status（#3#4）	Configuration Status of Extension Modules 3 and 4
……	……	……
SFD347	Extension module configuration status（#15#16）	Configuration Status of Extension Modules 15 and 16
SFD348	BD module configuration status（#1#2）	Configuration Status of BD Modules 1 and 2
SFD349	ED module configuration status（#1）	Configuration Status of ED Module 1
SFD350 : SFD359	Extension module configuration	Configuration of Extension Module 1
SFD360 : SFD369	Extension module configuration	Configuration of Extension Module 2
:	:	
SFD500 : SFD509	Extension module configuration	Configuration of Extension Module 16
SFD510 : SFD519	BD module configuration	Configuration of BD Module 1
SFD520 : SFD529	BD module configuration	Configuration of BD Module 2
SFD530 : SFD539	ED module configuration	Configuration of ED Module 1

Communication

ID	Function	Note
SFD600	COM1 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD610	COM2 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD620	COM3 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD630	COM4 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD640	COM5 free format communication buffer bit numbers	0: 8-bit 1: 16-bit

Appendix 2 Instruction Schedule

In appendix 2 all instructions that XD/XL series PLC support will be listed, including basic instructions, application instructions, special function instructions and motion control instructions and all instructions' corresponding application range will also be listed.

This part helps the users refer to instruction functions quickly. More about instructions application, please refer to XD/XL Series Programmable Controller【Instruction Part】.

Appendix 2-1. Basic Instruction List

Mnemonic	Function
LD	Initial logical operation contact type: NO(normally open)
LDI	Initial logical operation contact type: NC (normally closed)
OUT	Final logic operation type: coil drive
AND	Serial connection of NO
ANI	Serial connection of NC
OR	Parallel connection of NO
ORI	Parallel connection of NC
LDP	Operation start of pulse rising edge
LDF	Operation start of pulse falling edge
ANDP	Serial connection of pulse rising edge
ANDF	Serial connection of pulse falling edge
ORP	Parallel connection of pulse rising edge
ORF	Parallel connection of pulse rising edge
LDD	Read directly from the contact state
LDDI	Read directly NC
ANDD	Read directly from the contact state and connect serially
ANDDI	Read NC and connect serially
ORD	Read directly from the contact state and parallel connection
ORDI	Read NC and parallel connection
OUTD	Output the point directly
ORB	Parallel connection of serial circuit
ANB	Serial connection of parallel circuit
MCS	New bus line start
MCR	Bus line return
ALT	Alternate coil state
PLS	Connect on a scan cycle of pulse rising edge
PLF	Connect on a scan cycle of pulse falling edge
SET	Set coil on
RST	Set coil off
OUT	Drive counting coil
RST	Set coil off and current value rest to zero
END	I/O process and return to step 0
GROUP	Instruction block fold start
GROUPE	Instruction block fold end
TMR	Timing

Appendix 2-2. Application Instruction List

Sort	Mnemonic	Function
Program flow	CJ	Condition jump
	CALL	Call subroutine
	SRET	Subroutine return
	STL	Flow start
	STLE	Flow end
	SET	Open the assigned flow and close the current flow
	ST	Open the assigned flow and do not close the current flow
	FOR	Start of a FOR-NEXT loop
	NEXT	END of a FOR-NEXT loop
	FEND	End of main program
Data compare	LD=[※1]	LD activate if （S1）=（S2）
	LD>[※1]	LD activate if （S1）>（S2）
	LD<[※1]	LD activate if （S1）<（S2）
	LD<>[※1]	LD activate if （S1）≠（S2）
	LD>=[※1]	LD activate if （S1）≥（S2）
	LD<=[※1]	LD activate if （S1）≤（S2）
	AND=[※1]	AND activate if （S1）=（S2）
	AND>[※1]	AND activate if （S1）>（S2）
	AND<[※1]	AND activate if （S1）<（S2）
	AND<>[※1]	AND activate if （S1）≠（S2）
	AND>=[※1]	AND activate if （S1）≥（S2）
	AND<=[※1]	AND activate if （S1）≤（S2）
	OR=[※1]	OR activate if （S1）=（S2）
	OR>[※1]	OR activate if （S1）>（S2）
	OR<[※1]	OR activate if （S1）<（S2）
	OR<>[※1]	OR activate if （S1）≠（S2）
	OR>=[※1]	OR activate if （S1）≥（S2）
	OR<=[※1]	OR activate if （S1）≤（S2）
Data move	CMP[※1]	Data compare
	ZCP[※1]	Data zone compare
	MOV[※1]	Move
	BMOV	Block move
	PMOV	Block move
	FMOV[※1]	Multi-bit data move
	EMOV	Float move
	FWRT[※1]	FlashROM written

	MSET	Multi data set
	ZRST	Zone reset
	SWAP	Switch high bytes and low bytes
	XCH[※1]	Exchange data
Data operation	ADD[※1]	Addition
	SUB[※1]	Subtraction
	MUL[※1]	Multiplication
	DIV[※1]	Division
	INC[※1]	Increase 1
	DEC[※1]	Decrease 1
	MEAN[※1]	Mean
	WAND[※1]	Logic and
	WOR[※1]	Logic or
	WXOR[※1]	Logic exclusive or
	CML[※1]	Complement
	NEG[※1]	Negative
Data shift	SHL[※1]	Arithmetic shift left
	SHR[※1]	Arithmetic shift right
	LSL[※1]	Logic shift left
	LSR[※1]	Logic shift right
	ROL[※1]	Rotation shift left
	ROR[※1]	Rotation shift right
	SFTL[※1]	Bit shift left
	SFTR[※1]	Bit shift right
	WSFL	Word shift left
	WSFR	Word shift right
Data switch	WTD	Single word integer convert to double word integer
	FLT[※1]	16 bits integer convert to float
	FLTD[※1]	64 bits integer convert to float
	INT[※1]	Float convert to integer
	BIN	BCD convert to binary
	BCD	Binary convert to BCD
	ASCI	Hex convert to ASCⅡ
	HEX	ASCⅡ convert to Hex
	DECO	Coding
	ENCO	High bit coding
	ENCOL	Low bit coding

Sort	Mnemonic	Function
Float	ECMP[※2]	Float compare

Operation	EZCP※2	Float zone compare
	EADD※2	Float addition
	ESUB※2	Float subtraction
	EMUL※2	Float multiplication
	EDIV※2	Float division
	ESQR※2	Float square root
	SIN※2	Sine
	COS※2	Cosine
	TAN※2	tangent
	ASIN※2	Float arcsin
	ACOS※2	Float arccos
	ATAN※2	Float arctan
Clock	TRD	Read RTC data
	TWR	Write RTC data

※1: All the instructions are 16 bits and no 32 bits format in general. ※1 has 32 bits. 32 bits instructions are added D in front of its 16 bits instruction. Such as ADD(16 bits) / DADD(32 bits).

※2：These instructions are 32 bits, and have no 16 bits format.

Appendix 2-3. Special Instructions List

Sort	Mnemonic	Function
Pulse	PLSR[※1]	multi-segment pulse output
	PLSF[※1]	variable frequency pulse output
	DRVI[※2]	Relative single segment pulse output
	DRVA[※2]	Absolute single segment pulse output
	STOP	Pulse stop
	GOON	Pulse continue
	ZRN[※1]	Mechanical origin return
High speed count	CNT[※2]	Single-phase high speed count
	CNT_AB[※2]	AB phase high speed count
	RST	High speed counter reset
	DMOV[※2]	Read and write the high speed counter
High speed counter interruption	CNT[※2]	Single-phase 100 segments high speed counter(with interruption)
	CNT_AB[※2]	AB-phase 100 segments high speed counter(with interruption)
MODBUS communication	COLR	MODBUS coil read
	INPR	MODBUS input coil read
	COLW	MODBUS single coil write
	MCLW	MODBUS multi coil write
	REGR	MODBUS register read
	INRR	MODBUS input register read
	REGW	MODBUS single register write
	MRGW	MODBUS multi register write
Precision timing	STR[※2]	Precision timing
	DMOV[※1]	Read precise timing register
	STOP	Stop precise timing
Interrupt	EI	Enable interrupt
	DI	Disable interrupt
	IRET	Interrupt return
BLOCK	SBSTOP	BLOCK stop
	SBGOON	Carry on the suspensive BLOCK
	WAIT	Wait
	FROM/TO	Read/write module
Others	PWM	Pulse width modulation
	PID	PID operation control
	NAME_C	C function block

※1： All the instructions are 16 bits except the instructions with ※1 which has 32 bits. 32 bits instructions are added D in front of its 16bits instruction. Such as ADD(16bits) / DADD(32bits).

※2： The table doesn’t include X-NET instructions, please refer to X-NET fieldbus manual.

Appendix 3 PLC Configuration List

This part is used to check each model's configurations. Via this table, we can judge products type easily.

○ Selectable × Not support √ Support

series	USB port	232 port	485 port	RJ 45	Ex module	BD	High speed counter		Pulse output Channel(T /RT)	External interruption
							Incremental mode	AB phase		
XD1										
XD1-16	×	2	×	×	×	×	×	×	×	6
XD1-32	×	2	√	×	×	×	×	×	×	10
XD2										
XD2-16	×	2	√	×	×	×	3	3	2	6
XD2-24	×	2	√	×	×	1	3	3	2	10
XD2-32	×	2	√	×	×	1	3	3	2	10
XD2-48	×	2	√	×	×	2	3	3	2	10
XD2-60	×	2	√	×	×	2	3	3	2	10
XD3										
XD3-16	1	1	√	×	10	×	3	3	2	6
XD3-24	1	1	√	×	10	1	3	3	2	10
XD3-32	1	1	√	×	10	1	3	3	2	10
XD3-48	1	1	√	×	10	2	3	3	2	10
XD3-60	1	1	√	×	10	2	3	3	2	10
XD5										
XD5-16	1	1	√	×	16	×	3	3	2	10
XD5-24	1	1	√	×	16	1	3	3	2	10
XD5-32	1	1	√	×	16	1	3	3	2	10
XD5-48	1	1	√	×	16	2	3	3	2	10
XD5-60	1	1	√	×	16	2	3	3	2	10
XD5-24T4	1	1	√	×	16	1	4	4	4	10
XD5-32T4	1	1	√	×	16	1	4	4	4	10
XD5-48T4	1	1	√	×	16	2	4	4	4	10
XD5-48T6	1	1	√	×	16	2	6	6	6	10
XD5-60T4	1	1	√	×	16	2	4	4	4	10
XD5-60T6	1	1	√	×	16	2	6	6	6	10
XD5-60T10	1	1	√	×	16	2	10	10	10	10
XDM										
XDM-24T4	1	1	√	×	16	1	4	4	4	10
XDM-32T4	1	1	√	×	16	1	4	4	4	10
XDM-60T4	1	1	√	×	16	2	4	4	4	10
XDM-60T4L	1	1	√	×	16	2	4	4	4	10

series	USB port	232 port	485 port	RJ 45	Ex module	BD	High speed counter		Pulse output Channel(T /RT)	External interruption
							Incremental	AB phase		
XDM										
XDM-60T10	1	1	√	×	16	2	10	10	10	10
XDC										
XDC-24	×	2	√	×	16	1	4	4	2	10
XDC-32	×	2	√	×	16	1	4	4	2	10
XDC-48	×	2	√	×	16	2	4	4	2	10
XDC-60	×	2	√	×	16	2	4	4	2	10
XD5E										
XD5E-30T4	1	1	√	1	16	1	4	4	4	10
XD5E-60T10	×	1	√	2	16	2	10	10	10	10
XDME										
XDME-60T10	×	1	√	2	16	2	10	10	10	10
XL1										
XL1-16	×	1	√	×	×	×	×	×	×	6
XL3										
XL3-16	1	1	√	×	10	×	3	3	2	6
XL5										
XL5-32T4	1	1	√	×	16	1	4	4	4	10
XL5E										
XL5E-32T4	×	1	√	2	16	1	4	4	4	10
XLME										
XLME-32T4	×	1	√	2	16	1	4	4	4	10

Appendix 4 Common Questions Q&A

The following are the common questions may happen when using the PLC.

Q1：Why the coil is not set when the condition is satisfied?

A1： The possible reasons:

(1) Users may use one coil for many times, which leads to double coils output. And at this time, the later coil has priority.

(2) Coil may be reset, users can find the reset point by monitor function and modify the program.

Q2：What's the difference between COM1 and COM2?

A2： Both COM1 and COM2 support Modbus-RTU and Modbus-RTU/ASCII format. The difference is COM1 parameters can be set to default value by power on and off function of PLC.

Q3：Why PLC can not communicate with other devices?

A3： The possible reasons:

(1) communication parameters: PLC com port and device parameters must be the same.

(2) communication cable: Confirm connection correct and good and change cable to try again.

(3) communication serial port: Check the port by downloading PLC program. Rule out this problem if download successfully.

(4) contact manufacturer if all the above are ruled out.

Q4：How long can the PLC battery be used?

A4： Normally for 3~5 years.

WUXI XINJE ELECTRIC CO., LTD.
4th Floor Building 7,Originality Industry park,
Liyuan Development Zone, Wuxi City, Jiangsu
Province
www.xinje.com
Mail: Fiona.xinje@vip.163.com
Tel: (510) 85134136
Fax: (510) 85111290

XD/XL series PLC

User manual［positioning control］

Wuxi XINJE Electric Co., Ltd.

Data no.　PD02 20170518 3.4

XD/XL series PLC

User manual

[Positioning control]

Version 1

- Basic explanation

Thank you for purchasing Xinje XD/XL series PLC.

This manual mainly introduces XD/XL series PLC instructions.

Please read this manual carefully before using and wire after understanding the content.

About software and programming instructions, please refer to related manuals.

Please hand this manual over to operation users.

- Notices for users

Only experienced operator can wire the plc. If any problem, please contact our technical department.

The listed examples are used to help users to understand, so it may not act.

Please confirm that PLC specifications and principles are suitable when connect PLC to other products. Please conform safety of PLC and machines by yourself when use the PLC. Machines may be damaged by PLC errors.

- Responsibility declaration

The manual content has been checked carefully, however, mistakes may happen.

We often check the manual and will correct the problems in subsequent version. Welcome to offer advices to us.

Excuse us that we will not inform you if manual is changed.

- Contact information

If you have any problem about products, please contact the agent or Xinje company.

Tel: 0086 510-85134136 85123803

Fax: 0086 510-85111290

Address: Building 7 fourth floor, No.100, Dicui Rd, Wuxi, China.

Code : 214072

Catalog

Preface

———positioning control

This manual is XD/XL series PLC positioning control manual, it introduces pulse output and motion control function, is suitable for XD2, XD3, XD5, XDM, XDC, XD5E, XDME, XL3, XL5, XL5E, XLME series PLC (XD1 and XL1 have no positioning function).

1. XD/XL series PLC features:

- **Faster instruction processing speed**

XD/XL series PLC instruction processing speed is 12~15 times faster than XC series, especially for the floating number instruction, the unit of scanning period is μs.

- **Up to 10 to 16 modules and 2 BD cards, 1 ED module can be extended**

Similar to XC series PLC, XD3, XD5, XDM, XDC, XD5E series PLC also support extension module and BD card (XD1/XD2 cannot extend module and BD card), including digital, analog, temperature module. The extension modules can be 10 or 16, BD card 1 or 2.
XL series PLC can support 10 right extension modules, 1 left extension ED module.

- **Compatible with most functions of XC series**

XD/XL series PLC support most basic functions of XC series PLC.

- **Compatible with XC series program**

XD/XL series PLC software XDPPro can open the program of XC series PLC, but some different instructions will be shown in red colors, user only needs to modify this part of program.

- **XL has compact size**

XL series PLC is card type PLC, with a thinner and smaller appearance, which can greatly save the installation space.

- **X-NET fieldbus**

XD/XL PLC supports xnet fieldbus communication, which can realize fast and stable communication to XD/XL PLC and TG/TN touch screen. XDC series PLC supports the function of x-net motion bus and can control 20-axis synchronous motion.

- **Ethernet communication**

Ethernet PLC has RJ45 port and supports TCP/IP protocol. It can realize MODBUS-TCP communication and free format communication based on Ethernet. Supports program download, online monitoring, remote monitoring, and communication with other TCP/IP devices.

2. Product models

XD1 series models:

- XD1-16R/T-E/C
- XD1-32R/T-E/C

XD2 series models:

- XD2-16R/T-E/C
- XD2-24R/T/RT-E/C
- XD2-32R/T/RT-E/C
- XD2-48R/T/RT-E/C
- XD2-60R/T/RT-E/C

XD3 series models:

- XD3-16R/T/RT-E/C, XD3-16PT-E/C
- XD3-24R/T/RT-E/C, XD3-24PR/T/RT-E/C
- XD3-32R/T/RT-E/C, XD3-32PR/T/RT-E/C
- XD3-48R/T/RT-E/C, XD3-48PT-E/C
- XD3-60R/T/RT-E/C, XD3-60PT-E/C

XD5 series models:

- XD5-16R/T-E/C
- XD5-24R/T/RT-E/C, XD5-24T4-E/C
- XD5-32R/T/RT-E/C, XD5-32T4-E/C
- XD5-48R/T/RT-E/C
- XD5-60R/T/RT-E/C
- XD5-48T4-E/C
- XD5-48T6-E/C
- XD5-60T4-E/C
- XD5-60T6-E/C
- XD5-60T10-E/C

XDM series models:

- XDM-24T4-E/C, XDM-24PT4-E/C
- XDM-32T4-E/C, XDM-32PT4-E/C
- XDM-60T4-E/C
- XDM-60T10-E/C, XDM-60PT10-E/C
- XDM-60T4L-E

XDC series models:

- XDC-24T-E/C
- XDC-32T-E/C
- XDC-48T-E/C
- XDC-60T-E/C

XD5E series models:

- XD5E-30T4-E
- XD5E-60T10-E

XDME series models:

- XDME-60T10-E

3. XL series PLC

XL1 serise PLC:

- XL1-16T

XL3 serise PLC:

- XL3-16T, XL3-16R, XL3-16PR

XL5 serise PLC:

- XL5-32T4

XL5E serise PLC:

- XL5E-32T4

XLME serise PLC:

- XLME-32T4

4. Version requirements

XD series PLC: XDPpro software v3.2 and up.

XL series PLC: XDPpro software v3.5 and up.

Part of the instructions have version requirements, please refer to the instruction details.

1 Pulse output

Pulse output instruction list:

instruction	function	Instruction writing format	chapter
Pulse output			
PLSR	Multi-segment pulse output	PLSR S0 S1 S2 D	1-2-2
PLSF	Variable frequency pulse output	PLSF S0 S1 D	1-2-3
DRVI	Relative single segment positioning	DRVI S0 S1 S2 D1 D2	1-2-4
DRVA	Absolute single segment positioning	DRVI S0 S1 S2 D1 D2	1-2-5
ZRN	Mechanical return zero	ZRN S0 D	1-2-6
STOP	Stop pulse	STOP S0 S1	1-2-7
GOON	Continue to output pulse	GOON Yn	1-2-8

1-1. Function overview

XD2, XD3, XD5 (except XD5-48T6/60T6), XDC, XL3 series PLC have 2 channels of pulse output. XD5-48T6/60T6, XDM, XD5E series PLC have 4~10 channels of pulse output. The different pulse functions include single direction pulse output with or without accelertion/deceleration, multi-segment double direction pulse output. The max output frequency can up to 100KHz.

Note: as XC series PLC cannot write two or more pulse output instructions for same terminal in main program or process. But XD series PLC has no problem cause its condition is edge-triggered.

Pulse output terminal:

PLC model	Pulse channels	Pulse output terminal	output frequency	Output mode	Output format
XD2-16T/RT XD2-24T/RT XD2-32T/RT XD2-48T/RT XD2-60T/RT	2	Y0, Y1	0~100KHz	Open collector	Pulse+direction
XD3-16T/RT XD3-24T/RT XD3-32T/RT XD3-48T/RT XD3-60T/RT	2	Y0, Y1	0~100KHz	Open collector	Pulse+direction
XD5-16T XD5-24T/RT XD5-32T/RT XD5-48T/RT XD5-60T/RT	2	Y0, Y1	0~100KHz	Open collector	Pulse+direction
XD5-24T4 XD5-32T4 XD5-48T4 XD5-60T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector	Pulse+direction
XD5-48T6 XD5-60T6	6	Y0, Y1, Y2, Y3, Y4, Y5	0~100KHz	Open collector	Pulse+direction
XD5-60T10	10	Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11	0~100KHz	Open collector	Pulse+direction
XDM-24T4 XDM-32T4 XDM-60T4 XDM-60T4L	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector	Pulse+direction
XDM-60T10	10	Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11	0~100KHz	Open collector	Pulse+direction
XDC-24T	2	Y0, Y1	0~100KHz	Open	Pulse+direction

XDC-32T XDC-48T XDC-60T				collector	
XD5E-30T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector	Pulse+direction
XD5E-60T10	10	Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11	0~100KHz	Open collector	Pulse+direction
XDME-60T10	10	Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11	0~100KHz	Open collector	Pulse+direction
XL3-16T	2	Y0, Y1	0~100KHz	Open collector	Pulse+direction
XL5-32T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector	Pulse+direction
XL5E-32T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector	Pulse+direction
XLME-32T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector	Pulse+direction

Note:

※1: all the pulse can output frequency 100~200KHz, but not all the servo can work well, please connect 500Ω resistor between output and 24V power supply.

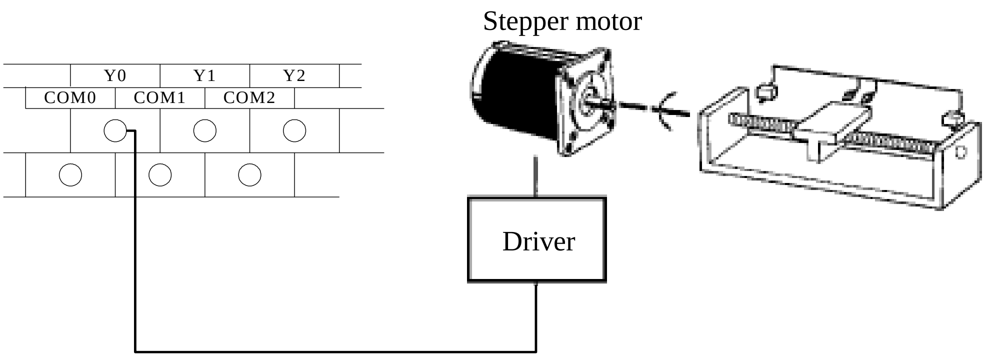
※2: the direction terminal can be set to any terminal except pulse output terminal when using positioning instruction.

※3: pulse output terminal transistor response time is below 0.5μs, other transistors is below 0.2ms.

※4: the pulse output terminal can be used to pulse direction output when it has no pulse output.

Load current

Please make the open collector transistor output load current in the range of 10~100mA (DC5~24V) when the basic unit (transistor output type) pulse output terminal is used to pulse output or positioning instruction.



Note:

※1: please use transistor terminal for pulse output. Such as XD3-16T-E or XD3-60T-E.

※2: it can choose any terminals for direction output except pulse output terminal.

※3: the pulse direction temirnal will keep the state after the pulse output finished. if the state is ON, it will keep ON after pulse output finished. if the pulse output instruction does not have direction, user can control the direction terminal state by manual. If the pulse output instruction has direction, the instruction will automatically control the direction terminal.

※4: the pulse output terminal LED will slight light when the pulse is outputting. Because the pulse is 50% empty square wave, so the LED will light in half of the period and off in another half of period.

※5: the pulse output terminal Yn will be ON in software when the pulse is outputting, and it will be OFF when the pulse output finished.

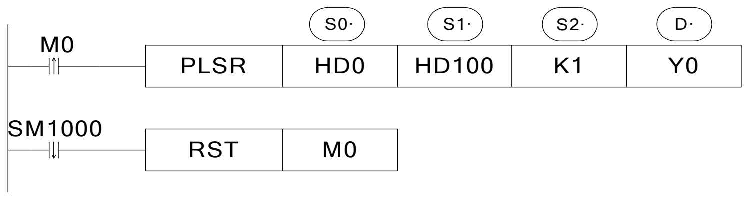
1-2. Pulse output type and instruction application

1-2-1. Pulse parameter and configuration

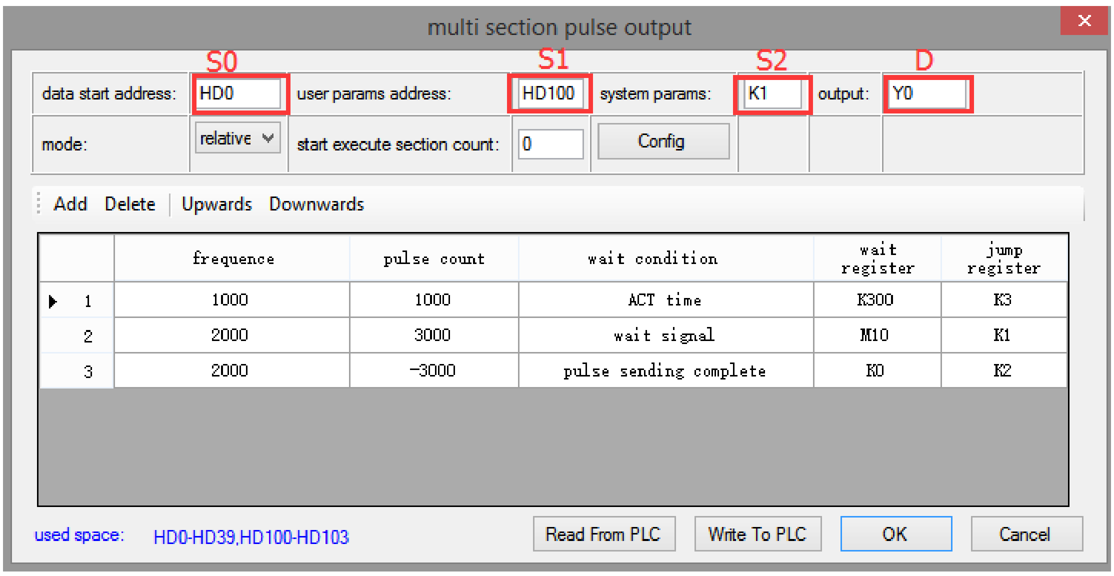
XD/XL series PLC pulse output function needs to configure the pulse data, user parameters and system parameters. This chapter will introduce all the parameters and configuration methods.

Now we take PLSR instruction as an example.

PLSR instruction write format:



Click  in the software or right click the PLSR instrution in the program to open the configuration window of PLSR.



Configuration table:

Configuration item	Function

Data start address	Pulse data parameter address, occupied 【S0】~【S0+N*10+8】(double words, N is pulse segment no.), store the pulse total segment number, pulse numbers, wait condition, register type and number, jump register type and number…
User parameter address	User parameter address, occupied【S1】~【S1+2】 (double words), store the mode (relative/absolute), starting execute segment no.
System parameter	Choose which group of parameters, each pulse output terminal can set four group of parameters, the default is K1 (group 1)
Mode	Relative, absolute mode, default is relative mode
Start execute section count	PLSR executed from which segment, default is 0 (start from segment 1)
Config	Set the system parameters which are saved in special Flash register SFD900~SFD2193, it can set 4 groups of parameters of 10 pulse output terminals

1-2-1-1. Pulse data parameters (S0)

The pulse data parameters are set in the address starting from S0, please refer to the following table:

◆ **Data starting address S0**

Address	Contents	Remark
S0+0 (double words)	Pulse total segment number (1~100)	
S0+2 (8 words)	Reserved (8 words)	
S0+10 (double words)	Segment 1 pulse frequency	Segment 1
S0+12 (double words)	Segment 1 pulse number	
S0+14	High 8-bit: 【wait condition】 (set when to send the next segment of pulse) H00: pulse output finished ("H" means hex format) H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse output finished	
	Low 8-bit:【wait condition register type】 (use together with【wait condition】) H00: constant H01: D H02: HD H03: FD H04: X H05: M	

	H06: HM	
S0+15 (double words)	【constant/register number (wait condition)】, use together with【wait condition】,【wait condition register type】	
S0+17	Low 8-bit:【jump register type】(set the next pulse segment no.) H00: constant H01: D H02: HD H03: FD	
S0+18 (double words)	【constant/jump register number】, use together with【jump register type】	
……	……	……
S0+N*10+0 (double words)	Segment N pulse frequency	Segment N
S0+N*10+2 (double words)	Segment N pulse numbers	
S0+N*10+4	Wait condition, wait condition register type	
S0+N*10+5 (double words)	Constant or register number (wait condition)	
S0+N*10+7	Jump register type	
S0+N*10+8 (double words)	Constant or register number (jump register)	

Note:

※1: pulse frequency is positive value (≥0), the value become larger is acceleration, become smaller is deceleration, it is not related to the pulse direction.

※2: pulse numbers can be positive or negative value, negative value means reverse direction pulse.

- **Wait condition (【S0+14】 high 8-bit)**

To set when to enter next segment of pulse.

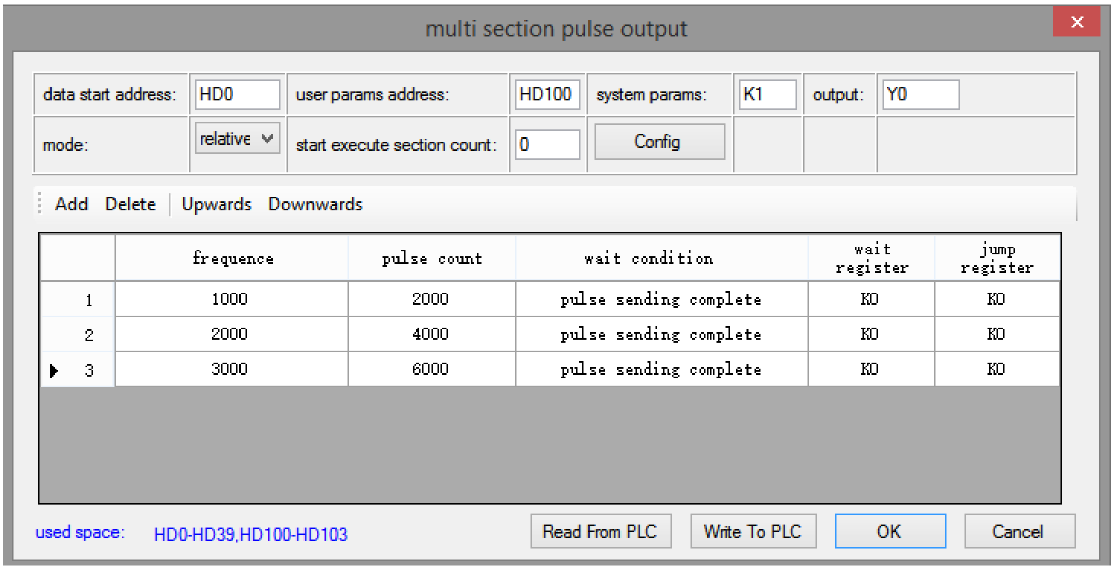
- **Pulse sending finished (H00)**

Jump to the setting pulse segment after executing this segment of pulse.

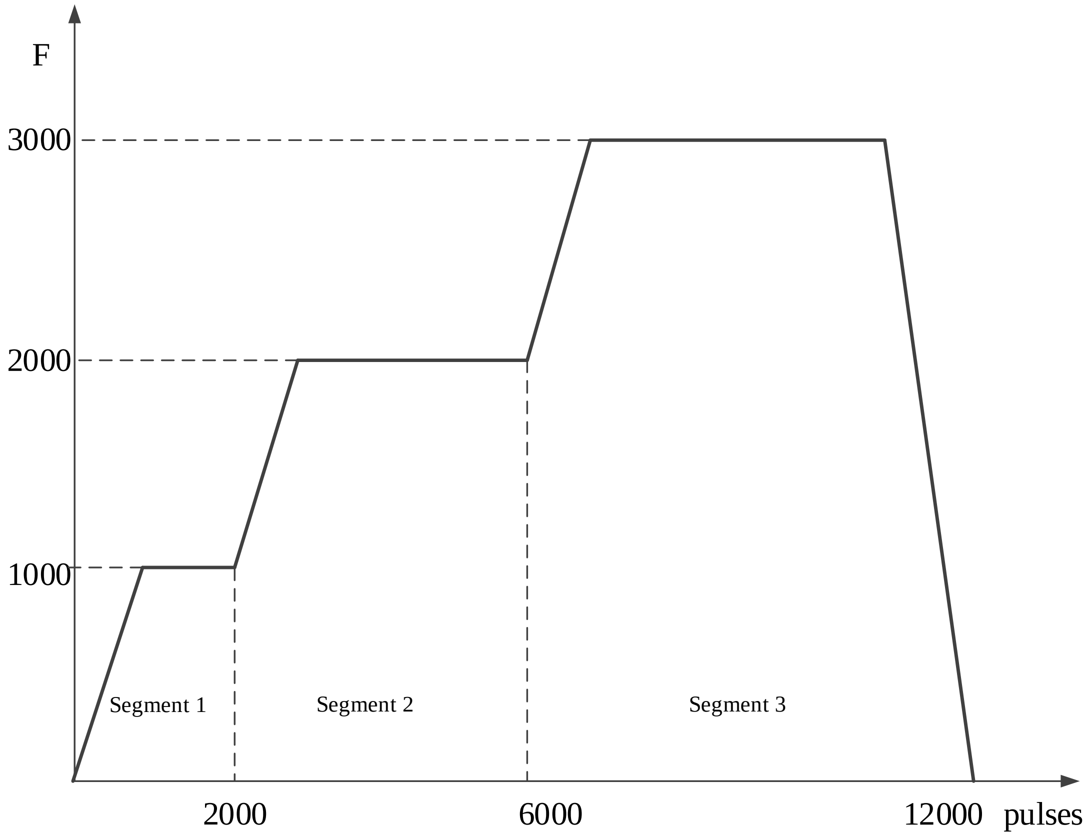
Example 1:

When the pulse intruction PLSR is triggered, it will send segment 1 2000 pulses with the speed 1000Hz, and jump to segment 2 at once after segment 1 finished. Segment 2 is 4000 pulses with speed 2000Hz. Then it will jump to segment 3 at once after semgent 2 finished. Segment 3 has 6000 pulses.

Configuration window:



Multi-segment pulse configuration

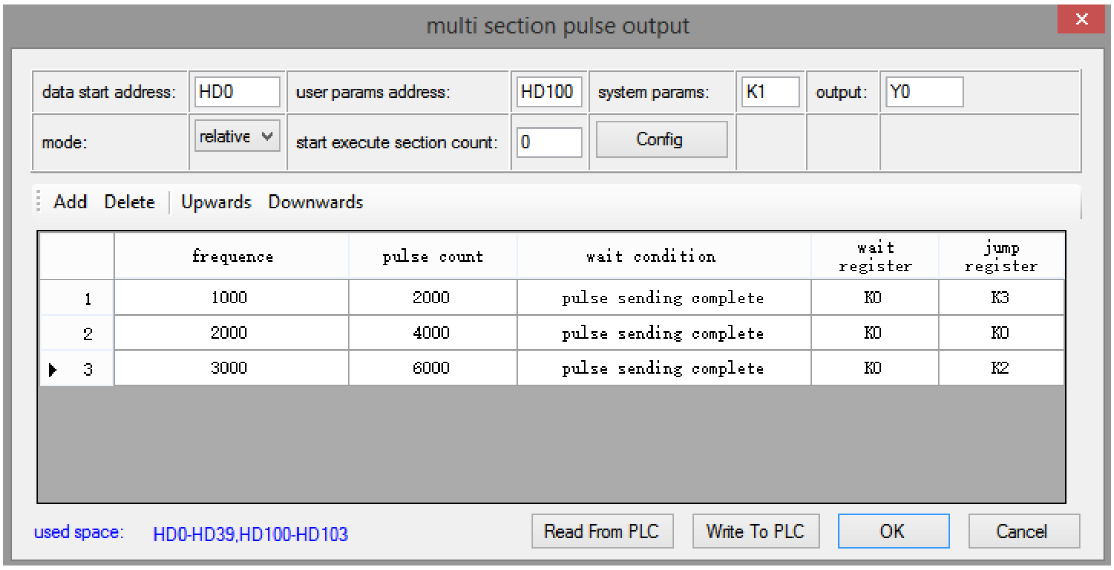


Multi-segment sequence control pulse wave

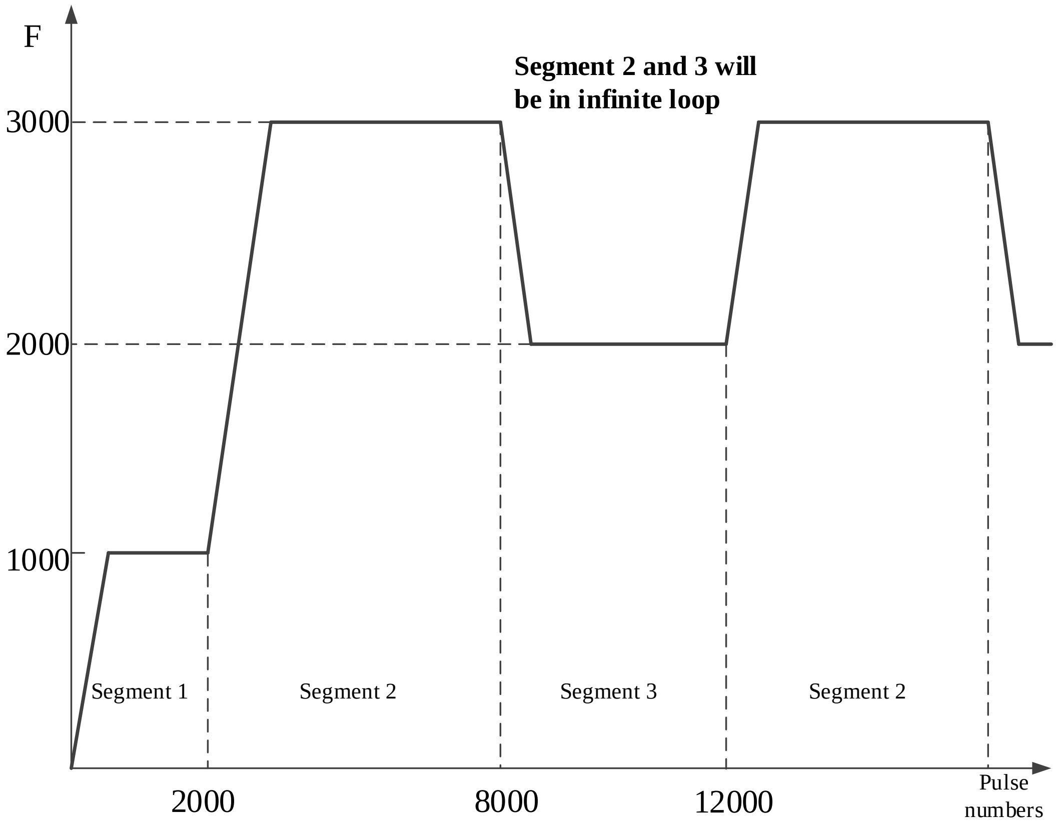
Example 2:

When the pulse instruction PLSR is triggered, it will send 2000 pulses with the speed 1000Hz, and jump to segment 3 to send 6000 pulses with the speed 3000Hz, then jump to segment 2 to send 4000 pulses, then jump to segment 3 to repeat the cycle.

The configuration window:



Multi-segment pulse output configuration table



Multi-segment pulse sending diagram

Note:

※1: the acceleration deceleration time can be set in 【config】 list, all the parameter details are in 【config guide】.

※2:【jump register】 set to K0, it will jump to the next segment. If it is not 0, it will jump to corresponding segment. For example, K3 will jump to segment 3.

※3: when setting multi-segment of pulse, and【jump register】is set, endless pulse outputting loop should be avoided.

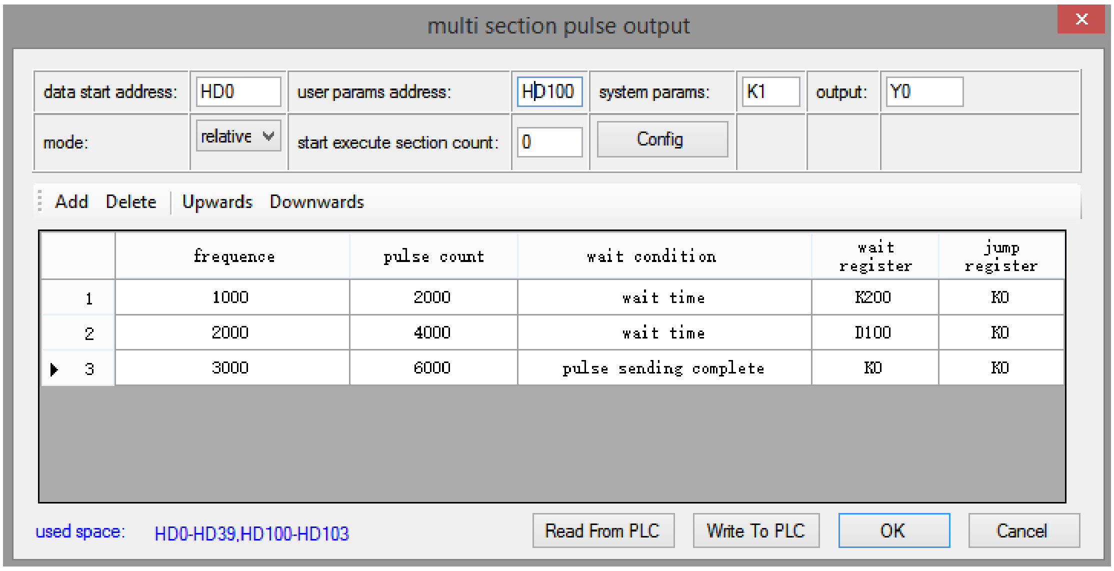
- **Wait time (H01)**

It starts to timing after present pulse segment end, it will jump to appointed segment when the time is up. The time can be constant or register D, HD, FD. The unit is ms.

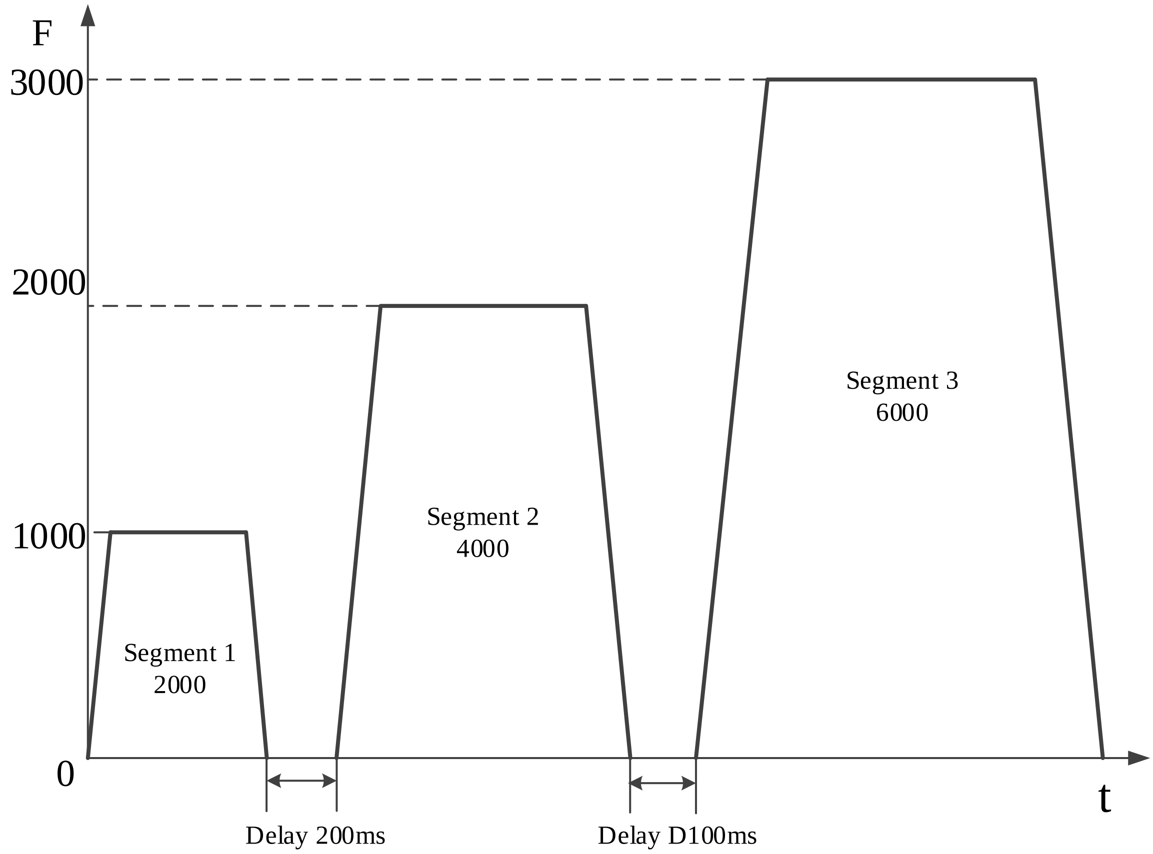
For example:

When the relative mode pulse instruction PLSR is triggered, it sends 2000 pulses with the speed of 1000Hz, it will delay 200ms after segment 1 end then jump to segment 2. It sends 4000 pulses with the speed 2000Hz, it will delay the time of D100 (if D100=100, it will delay 100ms), then jump to segment 3 which will send 6000 pulses.

Configurations:



Multi-segment pulse configuration table



Pulse sending diagram

Note:

※1: the acceleration deceleration time can be set in 【config】 list, all the parameter details are in 【config guide】.

※2: delay time range: 1~32767ms, set to 0 will be seemed to 1ms.

※3: if the delay time is over 32767ms, please use two pulse instructions, and timer between them.

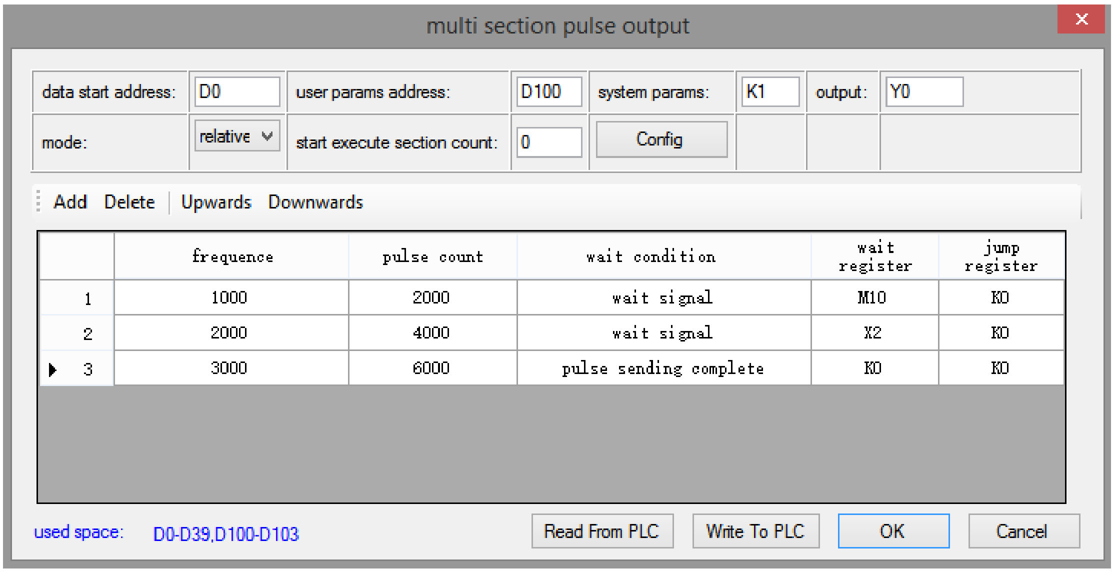
- **Wait signal (H02)**

It will wait for the wait signal after pulse sending finished. When the signal is ON or from OFF to ON, it will jump to appointed segment. The wait signal can be X, M, HM and so on.

For example:

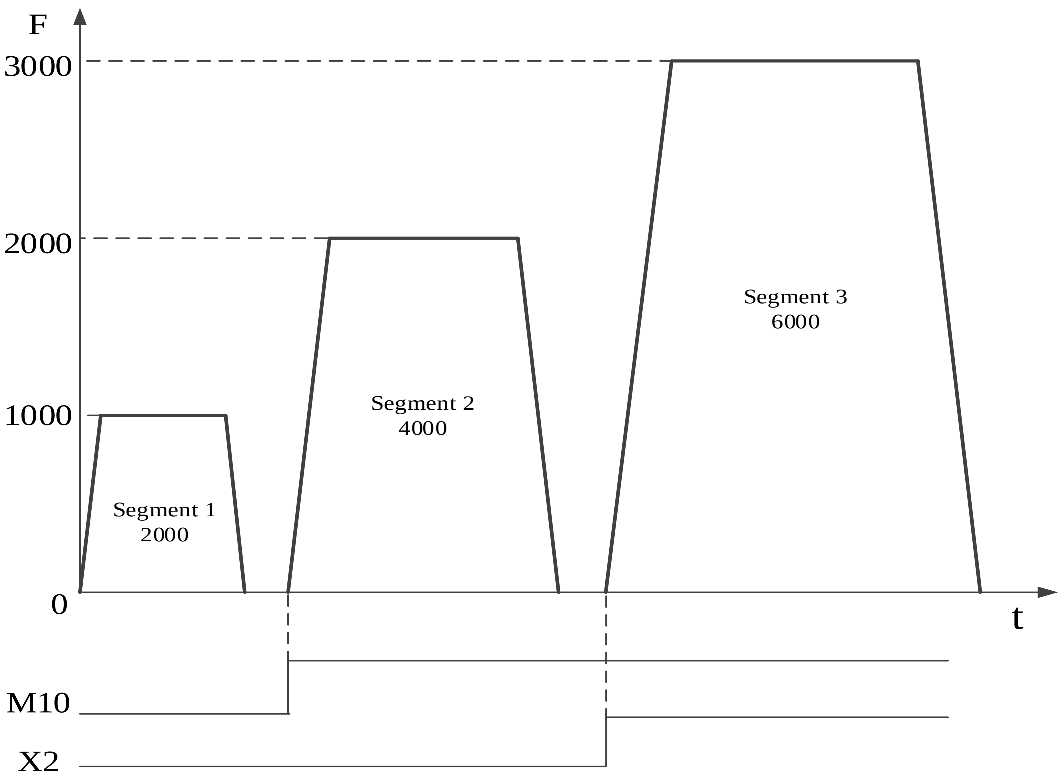
When the relative mode pulse instruction is triggered, it will send 2000 pulses with the speed 1000Hz, after segment 1 finished, it will wait for the M10 from OFF to ON, then jump to segment 2 which will send 4000 pulses with the speed 2000Hz, it will wait for X2 from OFF to ON, then jump to segment 3 which will send 6000 pulses.

Configurations:



	frequence	pulse count	wait condition	wait register	jump register
1	1000	2000	wait signal	M10	K0
2	2000	4000	wait signal	X2	K0
3	3000	6000	pulse sending complete	K0	K0

Multi-segment pulse output configuration table

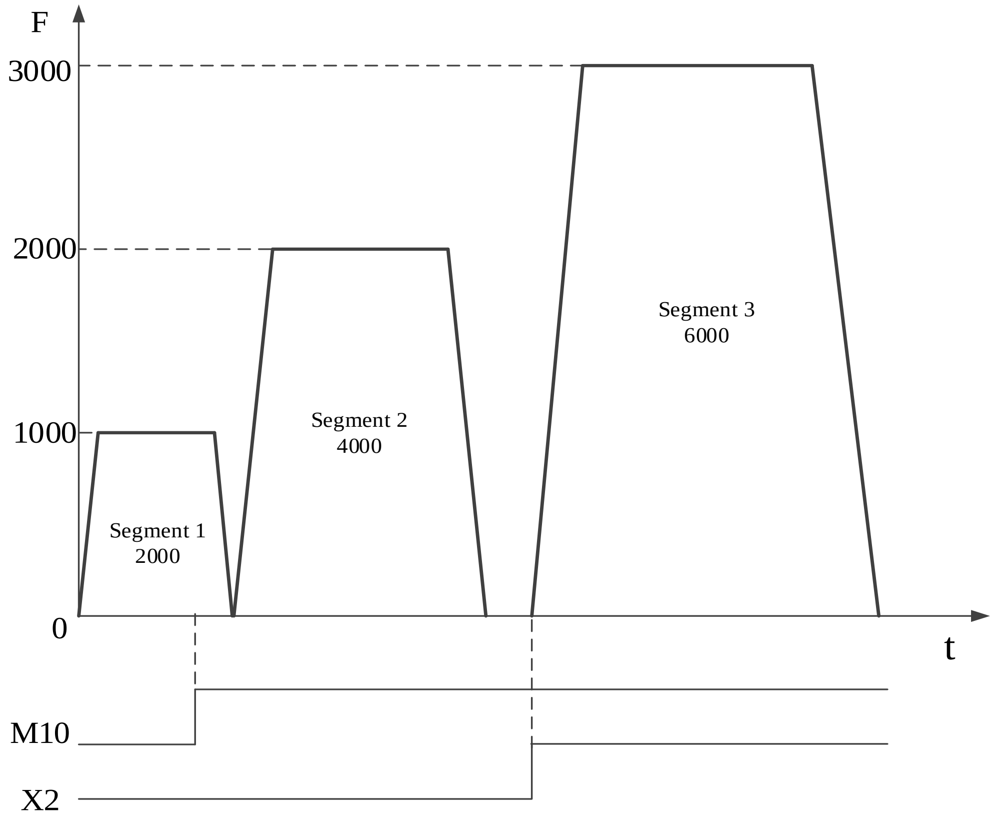


Pulse sending diagram

Note:

※1: the acceleration deceleration time can be set in 【config】 list, all the parameter details are in 【config guide】.

※2: if the present segment has not finished, but the wait signal is ON, it will jump to next segment after present segment finished, the wave is shown as below (M10 from OFF to ON in advance)



Pulse sending diagram

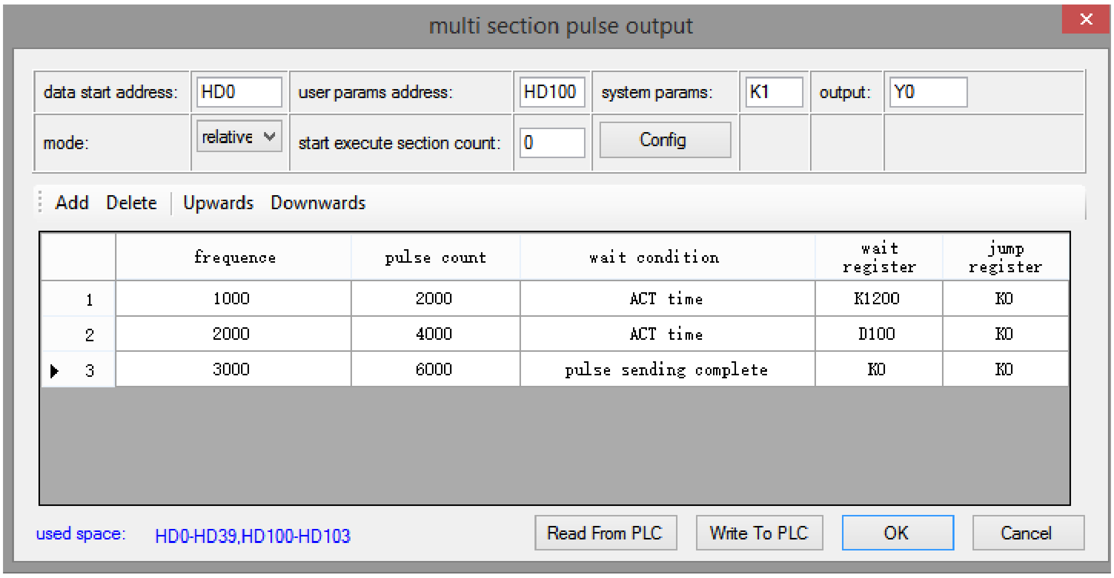
※3: if the wait signal is not ON after the present segment finished, it will wait until the signal is ON, then jump to the next segment.

- **ACT time (H03)**

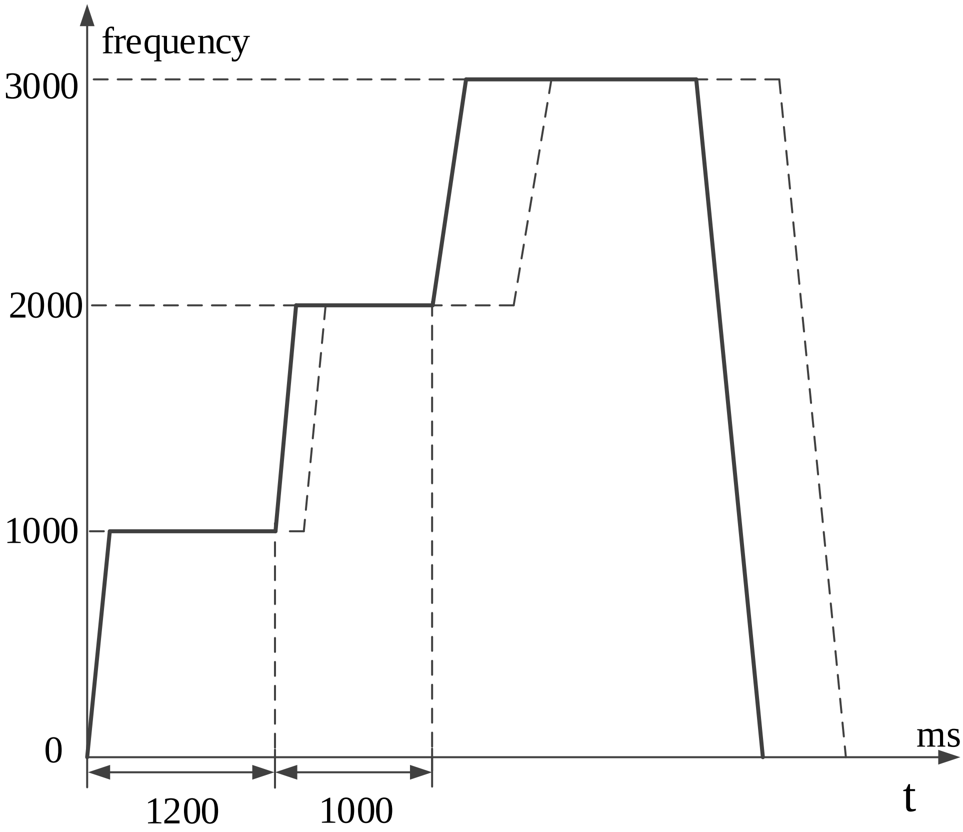
The pulse will output for the time appointed by ACT time, no matter the pulse sending process is finished or not, it will jump to the next segment at once. ACT time can be constant, or set through register D, HD, FD, the unit is ms.

For example: when the relative mode pulse instruction PLSR is triggered by pulse edge, it will output the first segment of pulse numbers with the speed 1000Hz, when the first segment pulse output time reaches 1200ms, no matter the pulse sending process is finished or not, it will jump to the second segment at once. When the second segment of pulse outputs with the speed 2000Hz and reaches the time setting in D100 (for example D100=1000), no matter the pulse sending process is finished or not, it will jump to the third segment at once and output 6000 pulses.

The configuration:



Multi-segment pulse output configuration

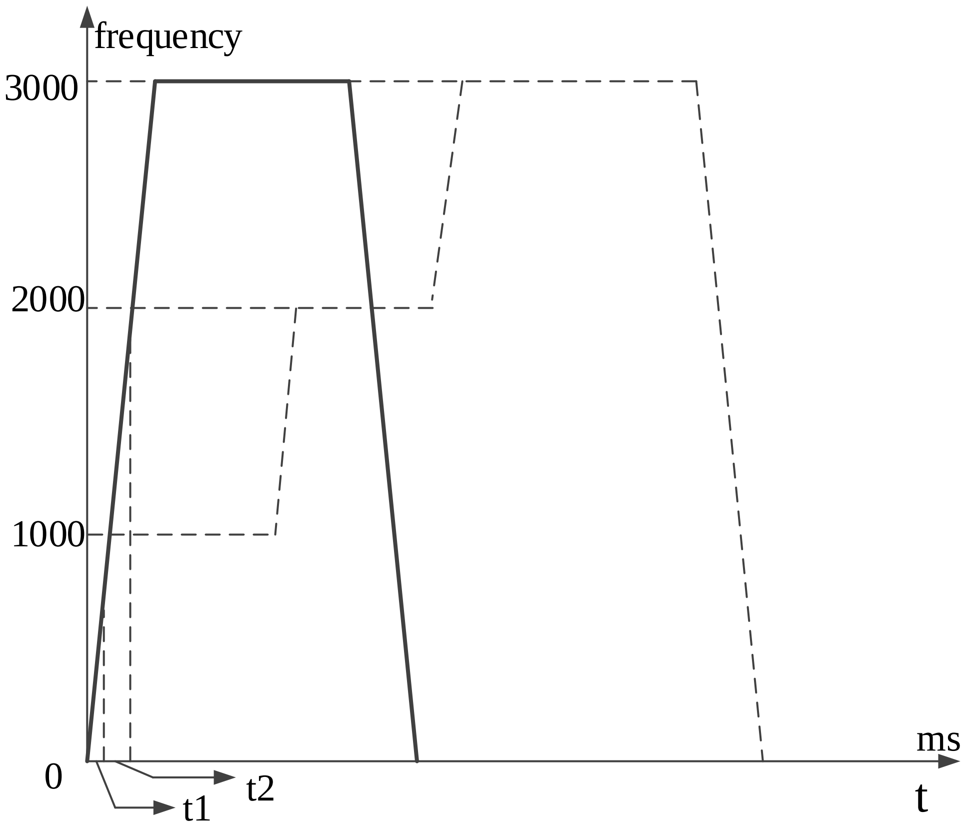


Pulse output diagram

Note:

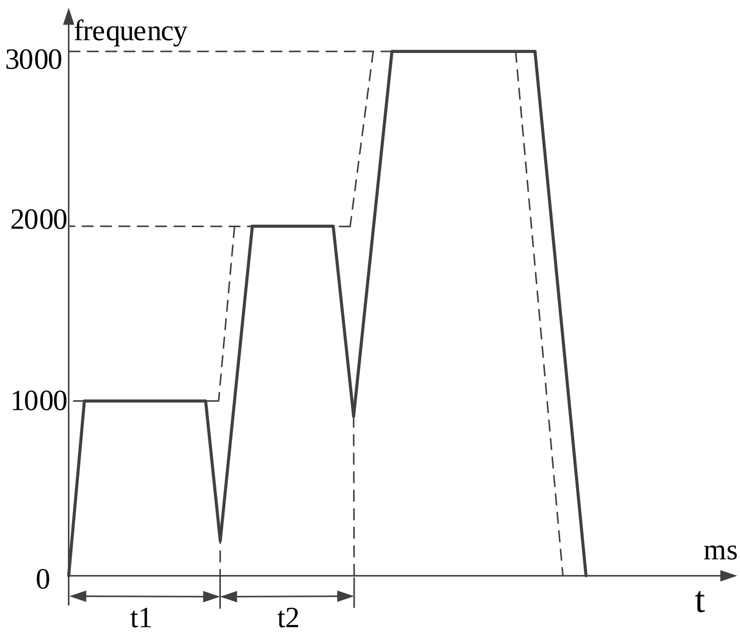
1: the accelertion time and deceleration time can be set in the parameter table, it will be explained in system parameters.

2: if the ACT time is very short and in the acceleration stage of the pulse segment, it will accelerate to the second segment from the position of ACT time reached, the same, it will accelerate to the third segment from the position of ACT time reached. Please see as the below diagram.



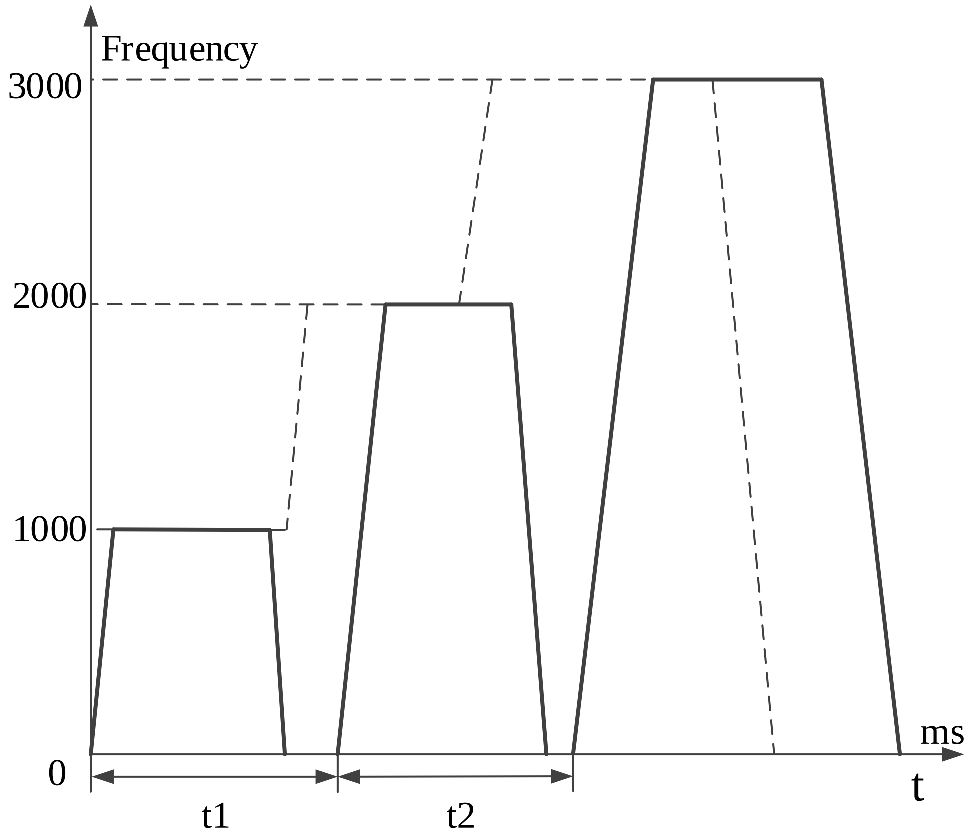
Pulse output diagram

3: if the ACT time is very long, and in the deceleration stage of the pulse segment, it will accelerate to the second segment from the position of ACT time reached, the same, it will accelerate to the third segment from the position of ACT time reached. Please see as the below diagram.



Pulse output digram

4: if the ACT time is very long, and the present pulse segment ends, it will wait the ACT time arrival and start the next segment. Please see the below diagram.



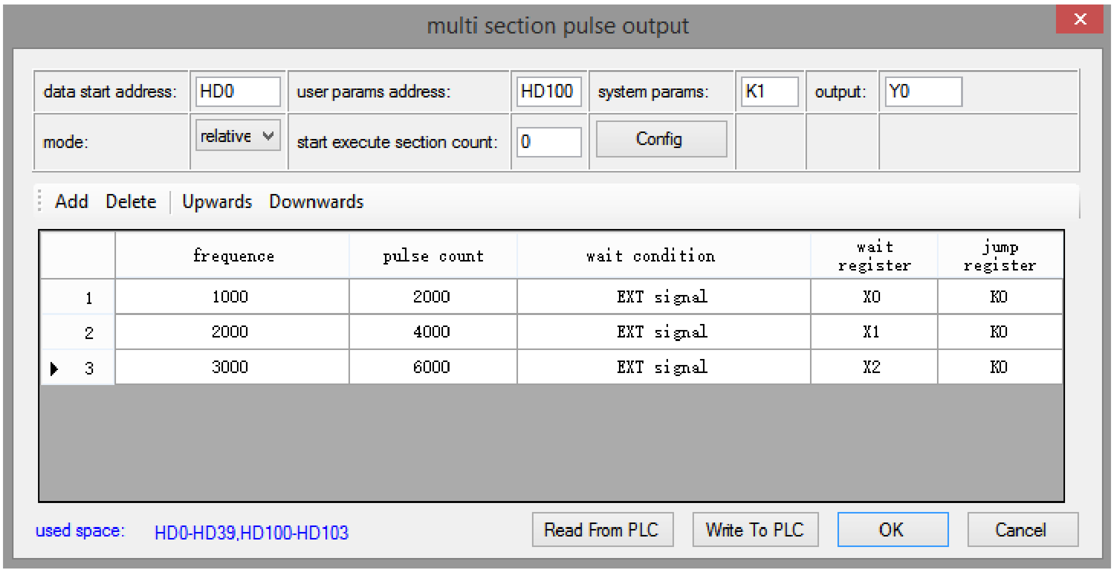
Pulse output diagram

- **EXT signal (H04)**

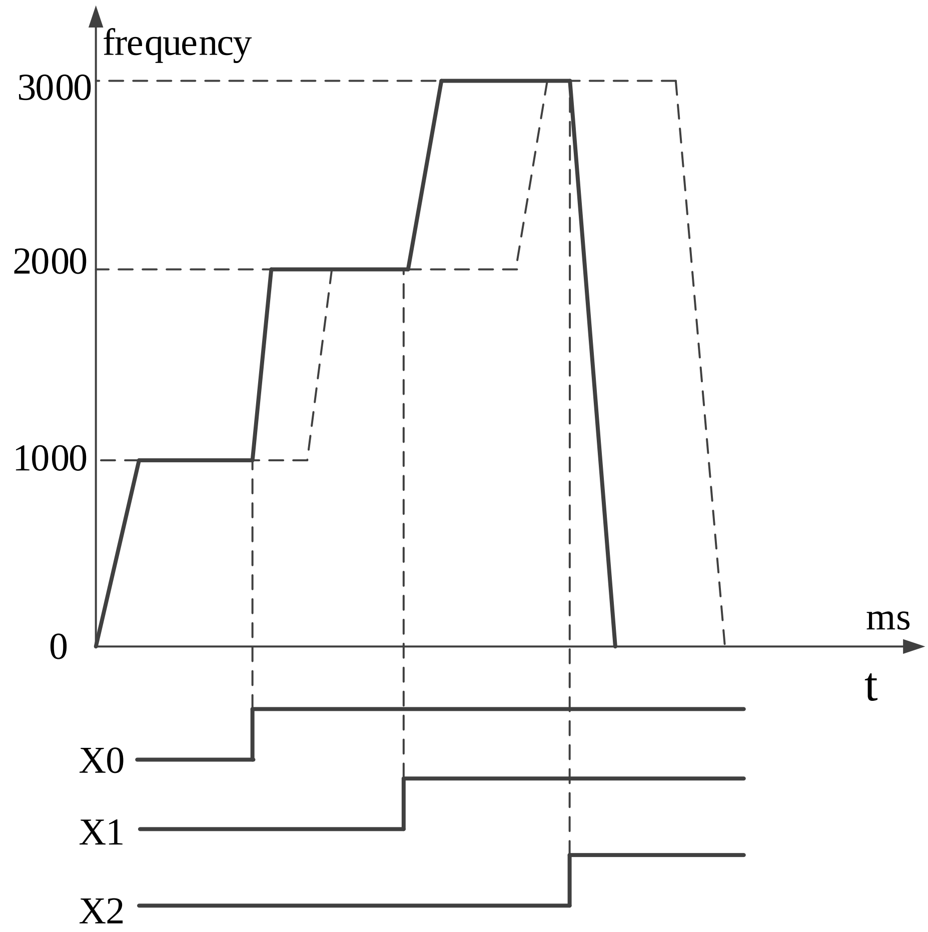
When the pulse is outputting (the pulse numbers have not been sent yet), if external signal is ON, it will jump to the next appointed segment. If the external signal has no action when the present pulse segment ends, it will wait for this signal. The external signal will input from X terminal (the response is higher if using external interruption terminal).

For example: when the relative mode pulse instruction PLSR is triggered by pulse edge, it will output the first segment of pulse numbers with the speed 1000Hz, the external signal inputs from X0 during the pusle is sending, it will jump to segment 2 at once. When the segment 2 pulse is sending with the speed 2000Hz, the external signal inputs from X1, it will jump to segment 3 at once. When the segment 3 pulse is sending with the speed 3000Hz, external signal inputs from X2, it will slow stop the pulse output at once.

The configuration window:



Multi-segment pulse output configuration

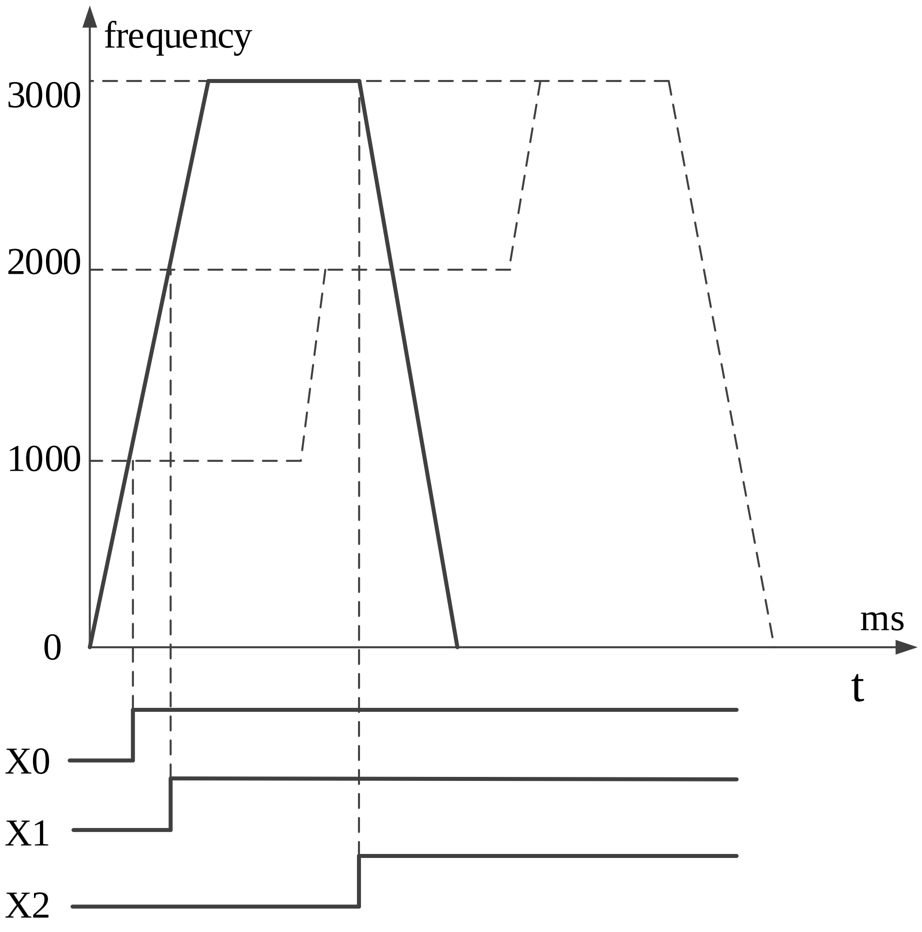


Pulse output diagram

Note:

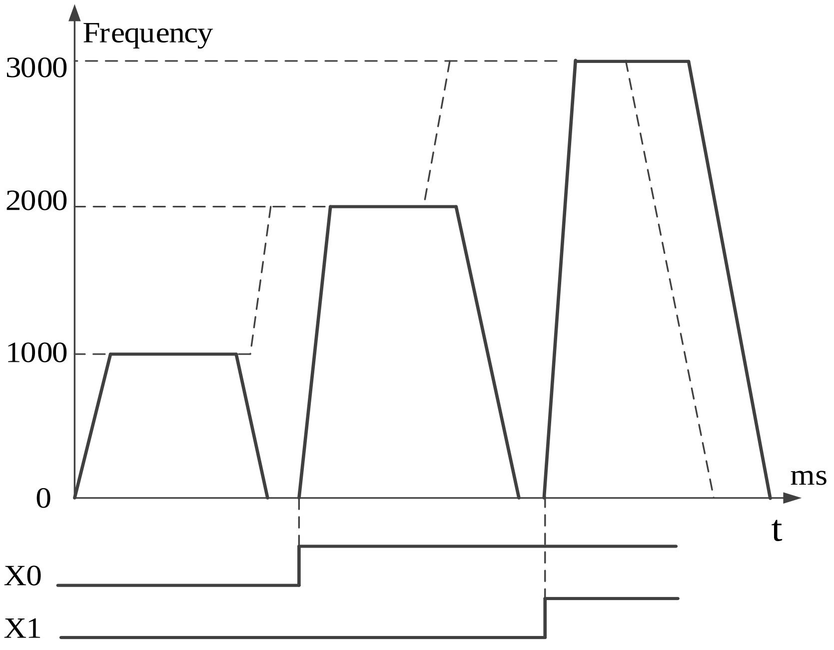
1: the acceleration and deceleration time can be set in parameter table, please refer to system parameters for details.

2: the pulse is accelerating when the EXT signal is triggered, it will accelerate from the present position to pulse segment 2. The same, it will accelerate from the present position of EXT singal triggered to segment 3. As shown of below diagram:



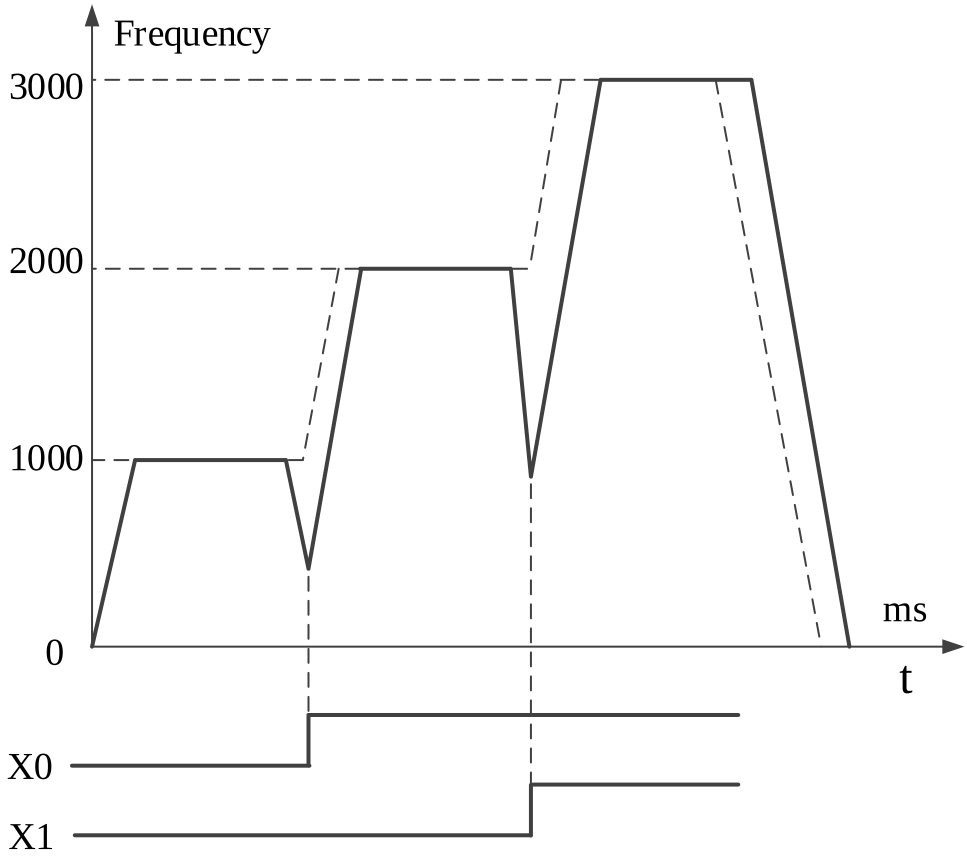
Pulse output diagram

3: if the EXT signal is triggered when the present pulse already ends, it will wait the EXT signal and start the next segment. Refer to below diagram.



Pulse output diagram

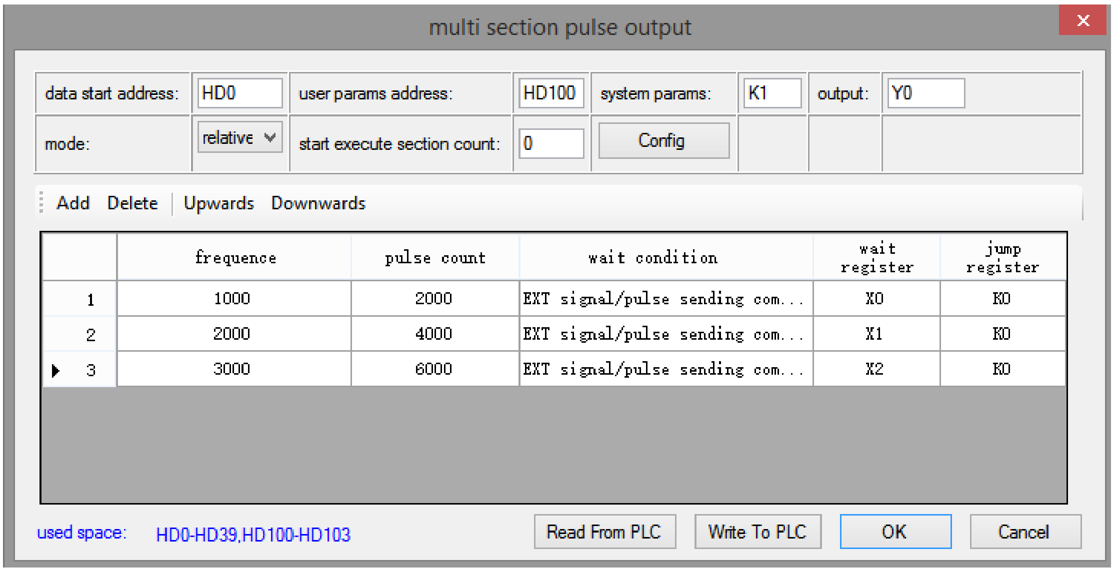
4: if the EXT signal is triggered when the pulse is decelearting, it will accelerate from present position to pulse segment 2, the same way, it will accelerate to pulse segment 3 from the position EXT signal is triggered. Refer to below diagram:



Pulse output diagram

- **EXT signal/pulse sending complete (H05)**

It will jump to appointed segment when the bit signal is triggered or pulse sending completes.
If the external signal is triggered before the pulse sending ends, it will jump to appointed segment, otherwise it will jump to appointed segment when present segment finishes (the pulse segment will send pulse as configuration parameters, if there is external EXT signal, it will not continue the present segment but jump to appointed segment).
For example:



Multi-segment pulse configuration

EXT signal X0 is valid when segment 1 pulse is sending(frequency 1000Hz, pulse number 2000), EXT signal X1 is valid when segment 2 pulse is sending(frequency 2000, pulse number 4000), EXT signal X2 is valid when segment 3 pulse is sending(frequency 3000Hz, pulse number 6000).

■ Wait register

- **Constant (H00)**

The value in register S0+N*10+5 (double word) is constant, range K0~K2147483647, eg. K2, K6, K3000.

- **D (H01)**

The value in register S0+N*10+5 (double word) is register D, for example, D0, D200.

- **HD (H02)**

The value in register S0+N*10+5 (double word) is register HD(latched register), for example HD0, HD200.

- **FD (H03)**

The value in register S0+N*10+5 (double word) is register FD(Flash register), for example, FD0, FD200.

- **X (H04)**

The value in register S0+N*10+5 (double word) is X(input signal), if the signal is external interruption terminal, the pulse will be triggered by interruption signal(response faster), for example X0, X6.

- **M (H05)**

The value in register S0+N*10+5 (double word) is M(normal coil), for example, M0, M200.

- **HM (H06)**

The value is register S0+N*10+5 (double word) is HM(latched coil), for example, HM0, HM200.

■ Jump register

- **Constant (H00)**

The register value in S0+N*10+8 (double word) is constant, range K0~K100, for example K2, K6.

- **D (H01)**

The value in register S0+N*10+8 (double word) is D(normal register), for example D0, D200.

- **HD (H02)**

The value in register S0+N*10+5 (double word) is HD(latched register), for example HD0, HD200.

- **FD (H03)**

The value in register S0+N*10+5 (double word) is FD(Flash register), for example FD0, FD200.

Note:

1: whatever it is constant or register, the value range is K0~K100.

2: this parameter means the present pusle segment ends and jumps to appointed segment. For example, the value is K6, it will jump to pulse segment 6 when the present pulse segment ends.

3: if the jump register or constant is 0, it will jump to next segment, if there is no next pulse segment, it will finish the present pulse segment then stop.

4: if the constant or register value is present segment number, it will infinite loop the present pulse segment.

1-2-1-2. Pulse user parameters (S1)

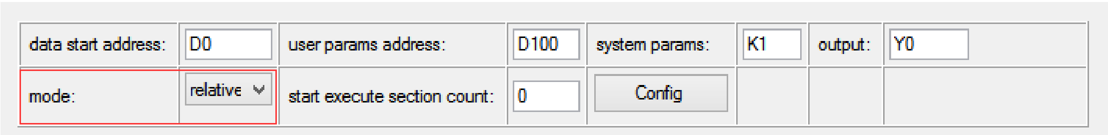
The pulse user parameters start from S1.

The pulse user parameters starting address (S1)

Address	Content
S1+0 (double word)	Pulse relative/absolute mode (0: relative 1: absolute) *[1]
S1+2 (double word)	Pulse start execution segment number (1~100)*[2]

a. Relative/absolute mode

S1+0 (double word) defines the pulse configuration mode is relative or absolute, default is relative mode.



For example:

There are 3 segments of pulse, segment 1 is 2000 pulse numbers, 1000Hz, segment 2 is 4000 pulse numbers, 2000Hz, segment 3 is 6000 pulse numbers, 3000Hz. The pulse configuration is shown as below:

	frequence	pulse count	wait condition	wait register	jump register
1	1000	2000	pulse sending complete	K0	K0
2	2000	4000	pulse sending complete	K0	K0
3	3000	6000	pulse sending complete	K0	K0

Relative mode configuration table

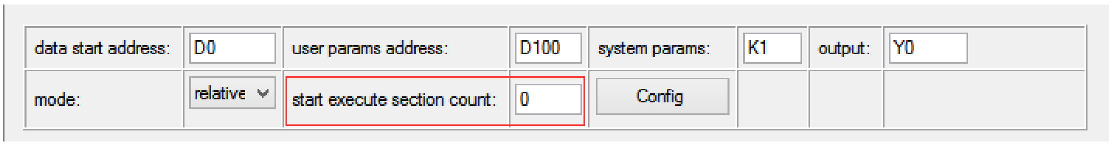
	frequence	pulse count	wait condition	wait register	jump register
1	1000	2000	pulse sending complete	K0	K0
2	2000	6000	pulse sending complete	K0	K0
3	3000	12000	pulse sending complete	K0	K0

Absolute mode configuration table

b. Start execution segment

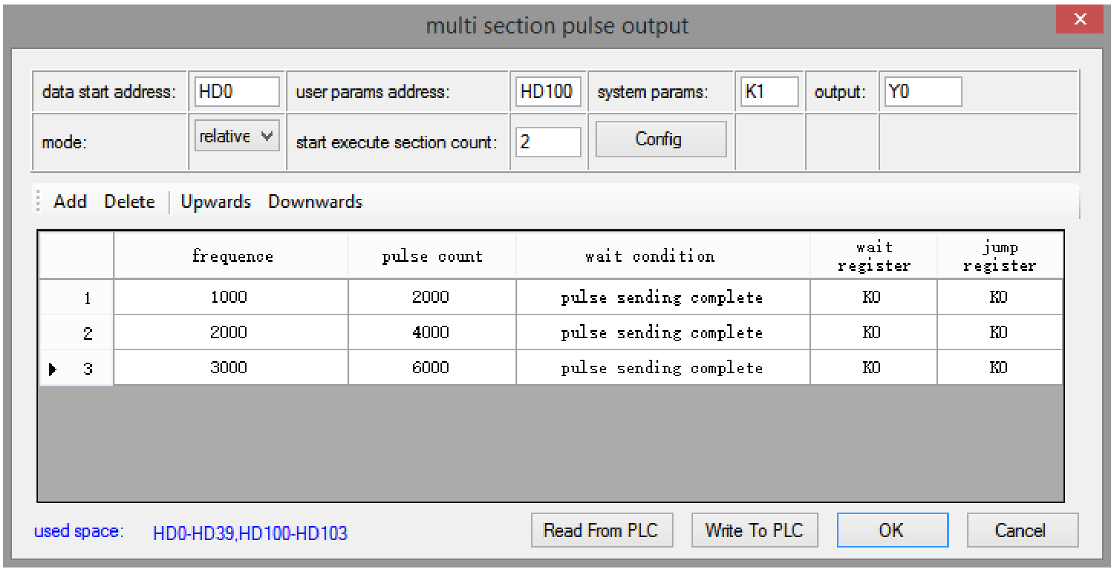
Start execution segment means the pulse instruction start segment (the pulse will start from the appointed segment but not segment 1).

Note: if it is set to 0 or 1, it will start from segment 1.



For example:

There are three segments of pulse: segment 1 is 1000Hz, 2000 pulse numbers, segment 2 is 2000Hz, 4000 pulse numbers, segment 3 is 3000Hz, 6000 pulse numbers, the start execution segment is 2:



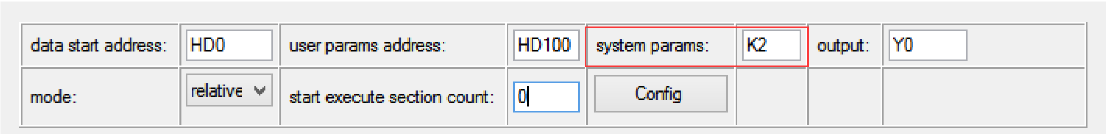
Multi-segment pulse output configuration table

The PLSR will send 4000 pulse numbers with the speed 2000Hz, then send 6000 pulse numbers with the speed 3000Hz.

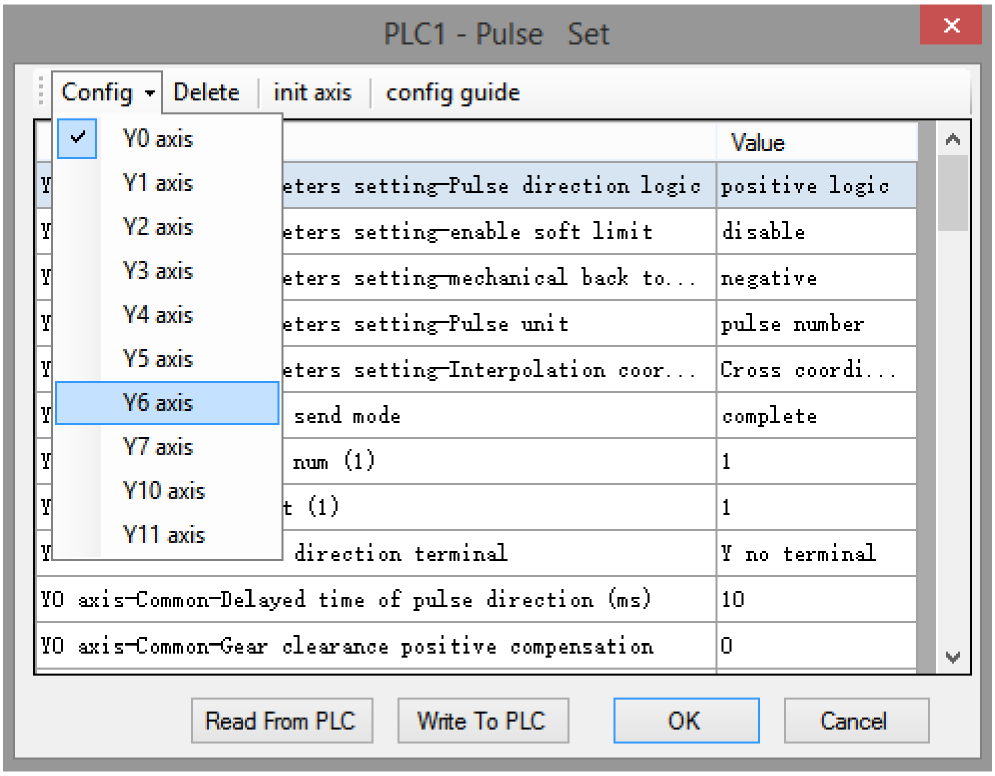
1-2-1-3. System parameters (S2)

There are 4 groups of system parameters. User can select one of them to execute the pulse output. Each pulse output terminal has related system parameter address.

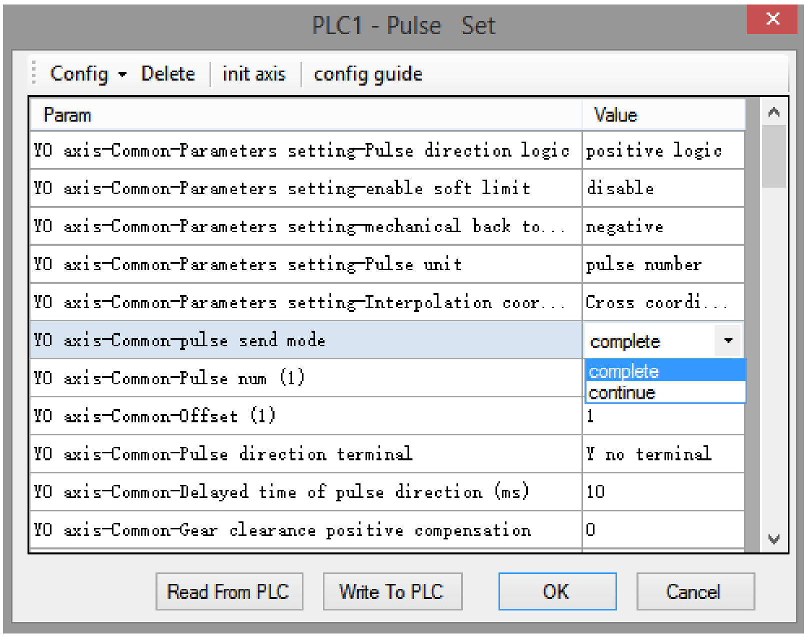
User can set the system parameter group no. in S2 (constant, register D, HD, FD…). As the following figure, system parameter group is 2, output terminal is Y0.



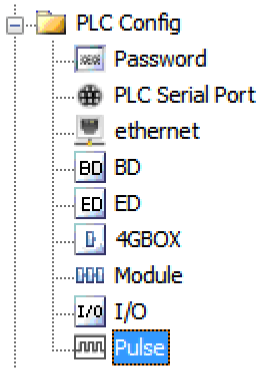
Click “config” button to enter system parameters.



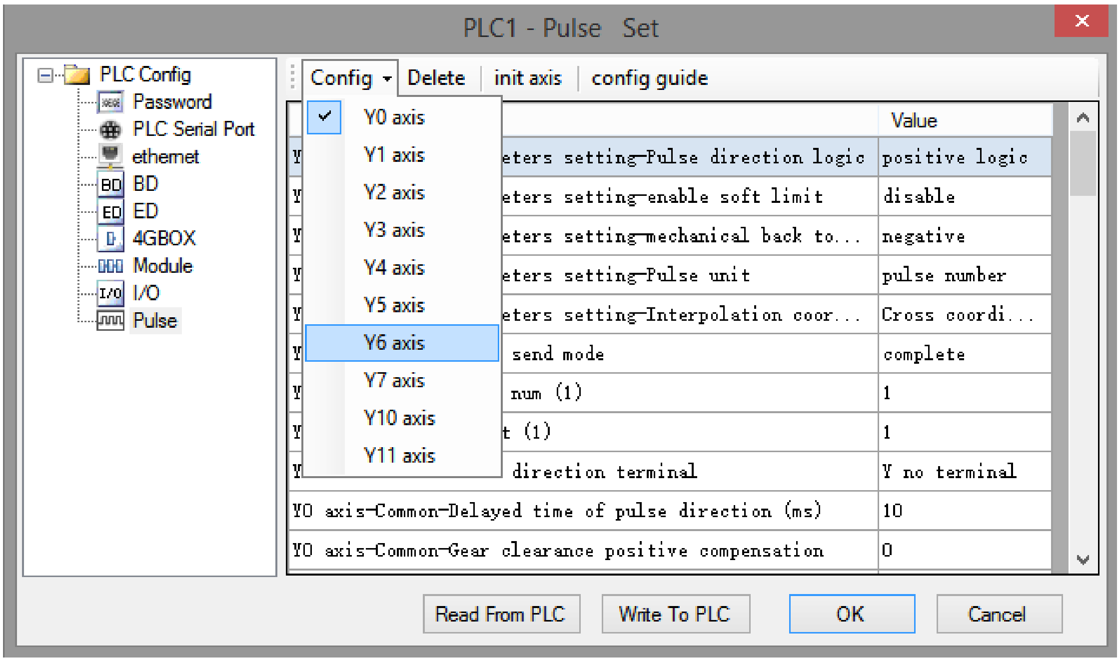
Click “config” can configure 10 channels (Y0~Y11) system parameters. Click each parameter to set the value:



Some instructions do not have panel configuration mode, when user needs to set the system parameters, please click the left side of software, and click "pulse" to set the parameters.



Then click "config" to set the parameters:



Note:

For the same pulse output terminal, the system parameters are shared. For example, if set the system parameters is K1, all the pulse instructions for Y0 will use system parameter group 1.

The following table shows the 4 groups of system parameter of first channel (Y0), each group of parameter can set different pulse default speed, pulse default speed acceleration and deceleration time, gear clearance acceleration/deceleration time, max speed limit, start speed and end speed… (please see below details).

Take first channel (Y0) as an example, other terminal system parameters please refer to appendix 3.

Address	Parameter	Explanation	Type	Output terminal
SFD900	Pulse parameters	Bit1: pulse direction logic 0: positive logic, 1: negative logic, default is 0 Bit2: soft position limit 0: OFF 1: ON, default is 0 Bit3: machine back to origin direction 0: negative direction 1: positive direction, default is 0 Bit10~ Bit8: pulse unit Bit8: 0: pulse numbers, 1: equivalent 000: pulse numbers 001: micron 011: centimillimeter 101: decimillimeter 111: millimeter Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate, 1: polar coordinate Default is 0	Common parameter	PULSE_1
SFD901	Pulse output mode	Bit0: pulse output mode 0: completion mode, 1: subsequent mode Default is 0		
SFD902	Pulse number/1 rotate low 16-bit			
SFD903	Pulse number/1 rotate high 16-bit			
SFD904	Movement amount/1 rotate low 16-bit			
SFD905	Movement amount/1 rotate high 16-bit			
SFD906	Pulse direction terminal	The number of terminal Y, 0xFF is no terminal		
SFD907	Direction delay time	Default is 20, unit: ms		
SFD908	Gear clearance positive compensation			
SFD909	Gear clearance negative compensation			

SFD910	Electric origin low 16-bit			
SFD911	Electric origin high 16-bit			
SFD912	Signal terminal state setting	Bit0: origin signal ON/OFF state Bit1: Z phase ON/OFF state Bit2: positive limit ON/OFF state Bit3: negative limit ON/OFF state 0: normally ON(positive logic), 1: normally close(negative logic), default is 0		
SFD914	Z phase terminal setting	Bit0~Bit7: X terminal number, 0xFF is no terminal		
SFD915	Limit terminal setting	Bit7~Bit0: positive limit X terminal number, 0xFF is no terminal Bit15~Bit8: negative limit X terminal number, 0xFF is no terminal		
SFD917	Zero clear CLR signal output terminal setting	Bit0~Bit7: Y terminal number, 0xFF is no terminal		
SFD918	Return speed VH low 16-bit			
SFD919	Return speed VH high 16-bit			
SFD922	Crawling speed VC low 16-bit			
SFD923	Crawling speed VC high 16-bit			
SFD924	Mechanical origin low 16-bit			
SFD925	Mechanical origin high 16-bit			
SFD926	Z phase numbers			
SFD927	CLR signal delay time	Default is 20, unit: ms		
SFD928	Wheel radius (polar coordinate)	Low 16-bit		
SFD929		High 16-bit		
SFD930	Soft limit positive pole value	Low 16-bit		
SFD931		High 16-bit		
SFD932	Soft limit negative pole value	Low 16-bit		
SFD933		High 16-bit		
……				

SFD950	Pulse default speed low 16-bit	It will output pulse with default speed when the speed is 0	Group1 parameter	
SFD951	Pulse default speed high 16-bit			
SFD952	Pulse default speed acceleration time			
SFD953	Pulse default speed deceleration time			
SFD954	Gear clearance acc/dec time			
SFD955	Acceleration deceleration mode	Bit1~Bit0: acc/dec mode 00: linear acc/dec 01: S curve acc/dec 10: sine curve acc/dec 11: reserved Bit15~ Bit2: reserved		
SFD956	Max speed limit low 16-bit			
SFD957	Max speed limit high 16-bit			
SFD958	Start speed low 16-bit			
SFD959	Start speed high 16-bit			
SFD960	End speed low 16-bit			
SFD961	End speed high 16-bit			
SFD962	Follow performance parameter	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 ticks.		
SFD963	Follow feedforward compensation parameter	0~100, percentage		
……				
SFD970	Pulse default speed low 16-bit	It will output pulse with default speed when the speed is 0	Group2 parameter	
SFD971	Pulse default speed high 16-bit			
SFD972	Pulse default speed acceleration time			
SFD973	Pulse default speed deceleration time			

SFD974	Gear clearance acc/dec time			
SFD975	Acceleration deceleration mode	Bit1~Bit0: acc/dec mode 00: linear acc/dec 01: S curve acc/dec 10: sine curve acc/dec 11: reserved Bit15~ Bit2: reserved		
SFD976	Max speed limit low 16-bit			
SFD977	Max speed limit high 16-bit			
SFD978	Start speed low 16-bit			
SFD979	Start speed high 16-bit			
SFD980	End speed low 16-bit			
SFD981	End speed high 16-bit			
SFD982	Follow performance parameter	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 ticks.		
SFD983	Follow feedforward compensation parameter	0~100, percentage		
……				
SFD990	Pulse default speed low 16-bit	It will output pulse with default speed when the speed is 0	Group3 parameter	
SFD991	Pulse default speed high 16-bit			
SFD992	Pulse default speed acceleration time			
SFD993	Pulse default speed deceleration time			
SFD994	Gear clearance acc/dec time			
SFD995	Acceleration deceleration mode	Bit1~Bit0: acc/dec mode 00: linear acc/dec 01: S curve acc/dec 10: sine curve acc/dec 11: reserved Bit15~ Bit2: reserved		

SFD996	Max speed limit low 16-bit			
SFD997	Max speed limit high 16-bit			
SFD998	Start speed low 16-bit			
SFD999	Start speed high 16-bit			
SFD1000	End speed low 16-bit			
SFD1001	End speed high 16-bit			
SFD1002	Follow performance parameter	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 ticks.		
SFD1003	Follow feedforward compensation parameter	0~100, percentage		
……				
SFD1010	Pulse default speed low 16-bit	It will output pulse with default speed when the speed is 0	Group4 parameter	
SFD1011	Pulse default speed high 16-bit			
SFD1012	Pulse default speed acceleration time			
SFD1013	Pulse default speed deceleration time			
SFD1014	Gear clearance acc/dec time			
SFD1015	Acceleration deceleration mode	Bit1~Bit0: acc/dec mode 00: linear acc/dec 01: S curve acc/dec 10: sine curve acc/dec 11: reserved Bit15~ Bit2: reserved		
SFD1016	Max speed limit low 16-bit			
SFD1017	Max speed limit high 16-bit			
SFD1018	Start speed low 16-bit			
SFD1019	Start speed high 16-bit			

SFD1020	End speed low 16-bit			
SFD1021	End speed high 16-bit			
SFD1022	Follow performance parameter	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 ticks.		
SFD1023	Follow feedforward compensation parameter	0~100, percentage		
…				

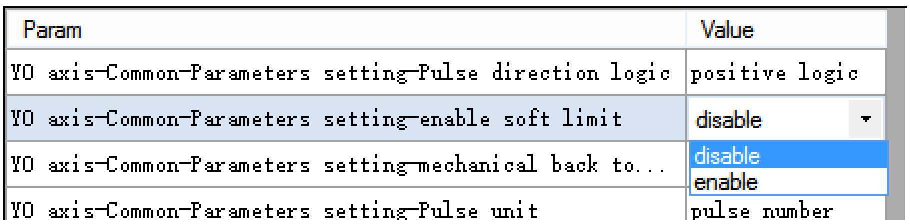
Common parameter

- Pulse direction logic

Pulse direction includes positive logic(default) and negative logic.
Positive logic: when the pulse numbers are positive value, it will output forward direction pulse (for example, HSD0 value is increasing), pulse direction terminal is ON. when the pulse numbers are negative value, it will output reverse direction pulse(for example, HSD0 value is decreasing), pulse direction terminal is OFF.
Negative logic: when the pulse numbers are positive value, it will output forward direction pulse (for example, HSD0 value is increasing), pulse direction terminal is OFF. when the pulse numbers are negative value, it will output reverse direction pulse(for example, HSD0 value is decreasing), pulse direction terminal is ON.
When the pulse is outputting, the direction terminal is ON, this terminal will not be reset automatically after the pulse output ends. The direction terminal will change the direction according to the pulse settings when pulse sends next time. If the pulse instruction has no direction, it needs to reset the direction terminal in the program.
Note:
1: this parameter default value is positive logic. All the program in this manual is made as positive logic.
2: fit for the instruction PLSR, PLSF, ZRN.

- Enable soft limit

In order to avoid the movement beyond the range of travel, the protection function is added to both ends of the travel. It is used to auto-search the origin signal and protect when backing to mechanical origin. It will judge the value of pulse accumulated register and protect the travel.
Note: soft limit and hardware limit can be used at the same time.
The parameter configuration:



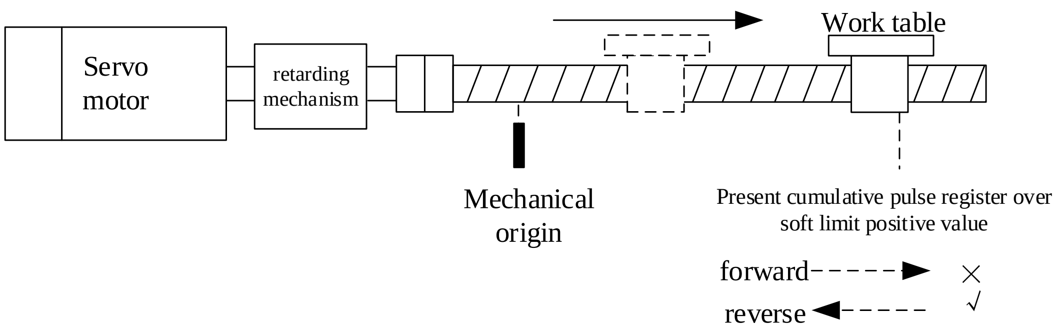
- Soft limit positive value

To prevent the table from moving beyond the range when executing the instruction PLSR, PLSF, DRVA, DRVI, interpolation instructions, it will add the value of present accumulated pulse register at the positive side of travel to protect the machine.

The configuration:

Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0
Y0 axis-group 1-Pulse default speed	0

If the forward sending pulse reaches soft limit positive value for instruction PLSR, PLSF, DRVA, DRVI, interpolation instruction, the pulse will slow stop. If the present cumulative pulse register value is over soft limit positive value, the forward pulse will always be prohibitted, but the reverse pulse can be triggered.



Note:

1: the parameter value cannot over max positive travel.

2: fit for PLSR, PLSF, DRVA, DRVI and interpolation instruction.

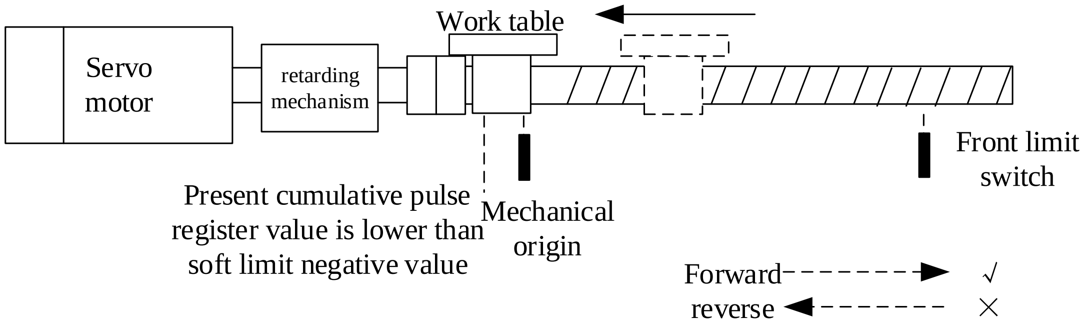
- Soft limit negative value

To prevent the table from moving beyond the range when executing the instruction PLSR, PLSF, DRVA, DRVI, interpolation instructions, it will add the value of present accumulated pulse register at the negative side of travel to protect the machine.

The configuration:

Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0
Y0 axis-group 1-Pulse default speed	0

If the forward sending pulse reaches soft limit negative value for instruction PLSR, PLSF, DRVA, DRVI, interpolation instruction, the pulse will slow stop. If the present cumulative pulse register value is lower than soft limit negative value, the reverse pulse will always be prohibitted, but the forward pulse can be triggered.



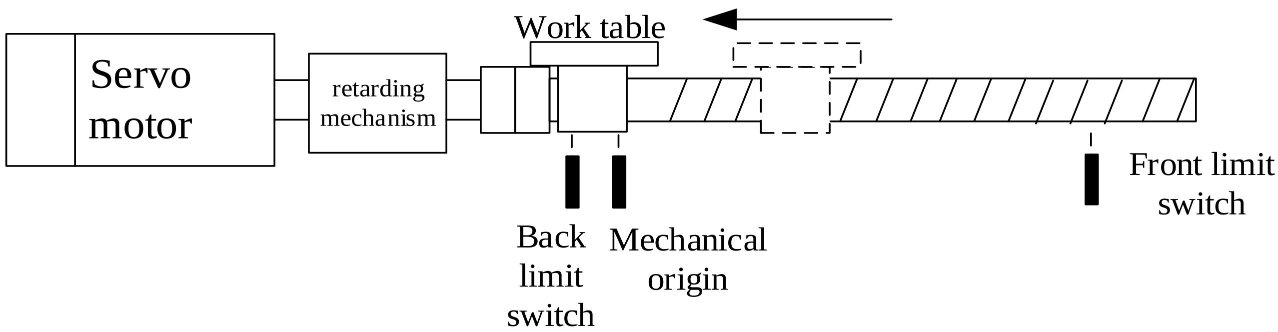
Note:
1: the parameter value cannot below min negative travel.
2: fit for PLSR, PLSF, DRVA, DRVI and interpolation instruction.

- Mechanical back to origin default direction

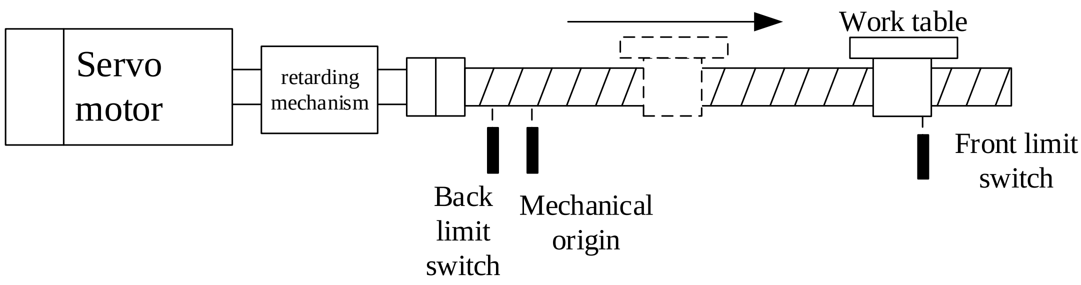
The work table default movement direction when the mechanical back to origin instruction ZRN is executed. The configuration:

Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to the...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coordina...	Cross coordi..

Negative: the work table will move in reverse direction when executing ZRN.



Positive: the work table will move in forward direction when executing ZRN.



- Pulse unit

The pulse unit include pulse number(default) and equivalent (1um, 0.01mm, 0.1mm,1mm optional).

axis-Common-Parameters setting-mechanical back to the...	negative
axis-Common-Parameters setting-Pulse unit	pulse number (pulse number / 1um / 0.01mm / 0.1mm / 1mm)
axis-Common-Parameters setting-Interpolation coordina...	
axis-Common-pulse send mode	
axis-Common-Pulse num (1)	

pulse number: if the pulse unit is pulse number, all the pulse frequency and number in the configuration table are calculated by pulse number. for example:

	frequence	pulse count	wait condition	wait register	jump register
1	1000	2000	pulse sending complete	K0	K0
2	2000	4000	pulse sending complete	K0	K0
3	3000	6000	pulse sending complete	K0	K0

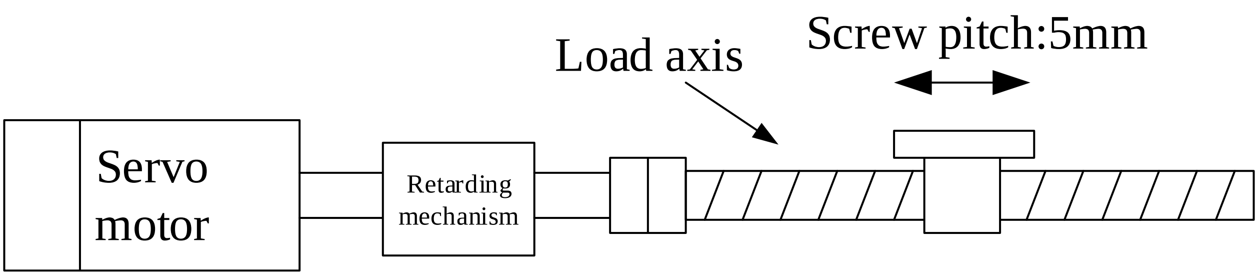
There are three segments in the configuration table, segment 1 will send 2000 pulses at the speed 1000Hz, segment 2 will send 4000 pulses at the speed 2000Hz, segment 3 will send 6000 pulses at the speed 3000Hz.

Equivalent: 1um, 0.01mm, 0.1mm, 1mm optional. All the pulse frequency and equivalent in the configuration table are calculated by length unit. Before explaining the equivalent, we will introduce pulse number (1 rotate) and offset(1 rotate) first.

- Pulse number (1 rotate)

The pulse number that the transmission mechanism rotates 1 circle. As there is retarding mechanism, the motor rotates one circle does not mean the transmission mechanism rotates one circle.

For example: one servo motor drives lead screw through retarding mechanism, the servo drive model is DS2-20P7-AS, servo motor model is MS-80ST-M02430B-20P7(encoder 2500 ppr), the servo drive electronic gear ratio is 1:1, reduction ratio of retarding mechanism is 1:5, the pitch of the ball screw is 5mm.



The pulse number of ball screw rotating one circle:

$$50000 = 2500 * 4 * \frac{5}{1}$$

- Offset(1 rotate)

The movement quantity of transmission mechanism rotates 1 circle. For example, in the above application, the offset is the ball screw pitch 5mm. If the object is synchronous belt, the offset is the synchronous belt transmission mechanism shaft perimeter.

After knowing the pulse number and offset, next we will understand how to set the equivalent. We will send three segments of pulse through the above mechanical structure.

	frequence	pulse count	wait condition	wait register	jump register
1	10	20	pulse sending complete	K0	K0
2	15	30	pulse sending complete	K0	K0
▶ 3	20	40	pulse sending complete	K0	K0

It configured three segments in above table. The pulse unit is equivalent. Segment 1 will move 20mm at the speed 10mm/s, segment 2 will move 30mm at the speed of 15mm/s, segment 3 will move 40mm at the speed of 20mm/s. The common parameters are configured as the below table:

axis-Common-Parameters setting-Pulse unit	1mm
axis-Common-Parameters setting-Interpolation coordina...	Cross coordi...
axis-Common-pulse send mode	complete
axis-Common-Pulse num (1)	50000
axis-Common-1mm(revolve)	5

transform the equivalent to related pulse frequency and pulse number, please see below table:

No.	Pulse unit	Frequency/speed	Pulse number/length
1	equivalent	10mm/s	20mm
	Pulse number	100000pulse/s	200000 pulse
2	equivalent	15mm/s	30mm
	Pulse number	150000pulse/s	300000 pulse
3	equivalent	20mm/s	40mm
	Pulse number	200000pulse/s	400000 pulse

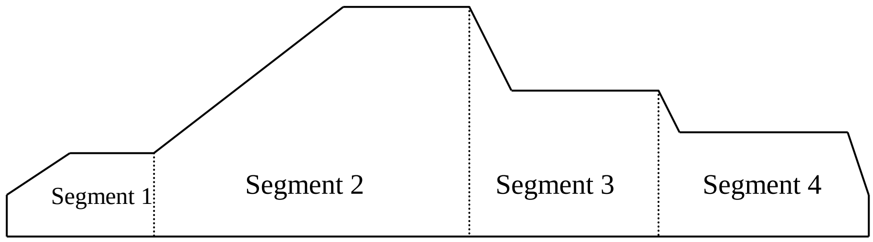
Note:
1: when the pulse unit is pulse number, Y0 axis cumulative pulse register HSD0 (double word) is pulse numbers. When the pulse unit is equivalent, Y0 axis cumulative pulse register HSD0 (double word) is pulse numbers. Register HSD2(double word) is cumulative equivalent length.
2: when the pulse unit is equivalent, all the parameters will execute as equivalent, the length unit will transform to the equivalent unit, for example 1mm, then all the unit will transform as 1mm. and the unit of offset(1 rotate) should be same to pulse unit setting, for example, pulse unit is 0.1mm, offset is 6, which means the offset of one rotate is 6*0.1mm=0.6mm, and other unit related to length and speed will be 0.1mm or 0.1mm/s.
3: please note the max output frequency cannot over 200Khz when the pulse unit is equivalent.
4: fit for instruction PLSR, PLSF, ZRN.

- Interpolation coordinate mode

This parameter is not valid for now, no need to modify.

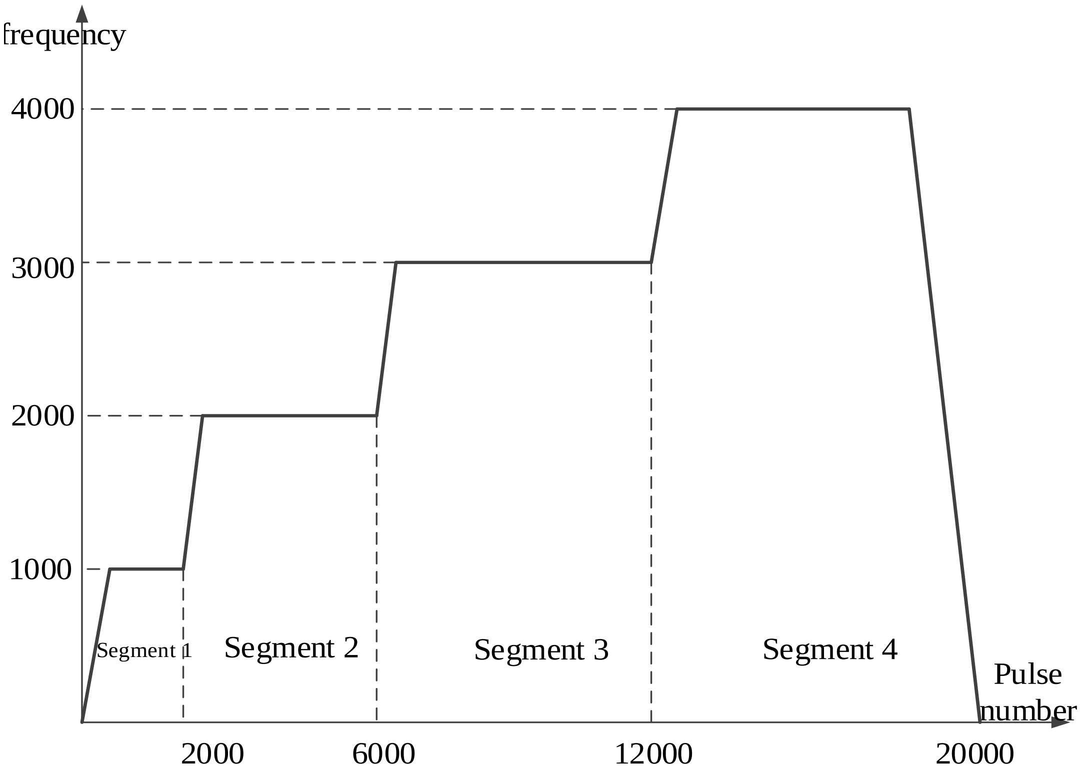
- Pulse send mode

It includes complete mode and continue mode.
Complete mode: it starts next segment of pulse when present segment pulse finishes.

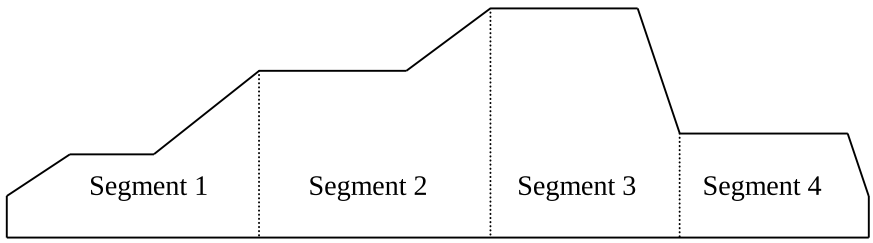


The pulse curve please refer to above diagram. Each segment will send the pulse numbers at setting speed. Except the last segment, each segment includes rising or falling part, stable part. The last segment includes rising part, falling part and stable part.

For example: the PLC needs to send four segments of pulse, segment 1 frequency is 1000Hz, pulse number is 2000, segment 2 frequency is 2000Hz, pulse number is 4000, segment 3 frequency is 3000Hz, pulse number is 6000, segment 4 frequency is 4000Hz, pulse number is 8000. It will send the pulse as complete mode, the curve please see below diagram.

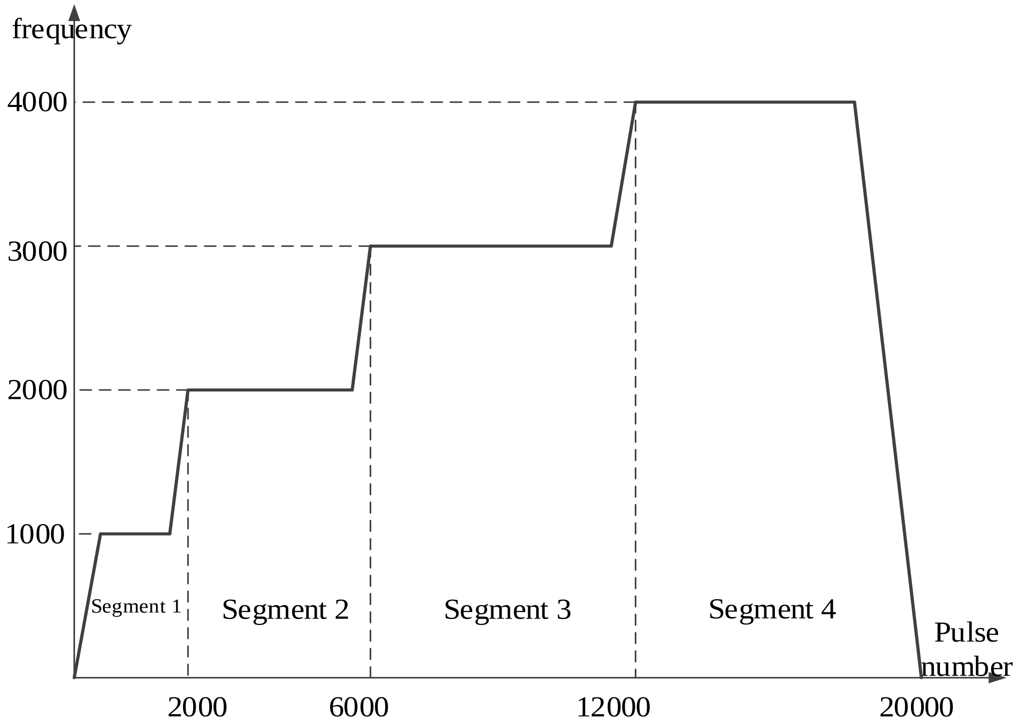


Continue mode: it already accelerates or decelerates to next segment when present segment pulse finishes sending.



The pulse curve diagram is as the above. When the present segment finishes sending, it already switch to next segment speed. Except segment 1, each segment includes stable part, rising part or falling part. Segment 1 includes rising part or falling part, stable part, rising or falling part.

For example: the PLC needs to send four segments of pulse, segment 1 frequency is 1000Hz, pulse number is 2000, segment 2 frequency is 2000Hz, pulse number is 4000, segment 3 frequency is 3000Hz, pulse number is 6000, segment 4 frequency is 4000Hz, pulse number is 8000. It will send the pulse as continue mode, the curve please see below diagram.



Note: the two modes are fit for instruction PLSR and PLSF.

- Pulse direction terminal

The pulse direction of PLSR needs to configure in the parameter table:

Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y no terminal
Y0 axis-Common-Delayed time of pulse direction (ms)	10

XD2, XD3, XD5 (except XD5-48T6/60T6) and XDC series transistor output PLC all have two channels of pulse output (Y0, Y1), the direction terminal can be any terminal except Y0 and Y1. XD5-48T6/60T6 has 6 channels of pulse output (Y0, Y1, Y2, Y3, Y4, Y5). XDM series has 4 channels or 10 channels pulse output (Y0, Y1, Y2, Y3 or Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11). The direction terminal can be any terminal except pulse output terminal.

The pulse output terminal uses high-speed optocoupler(response time below 5us), other terminals use normal optocoupler(response time below 0.2ms).

When Y0 is used to pulse output, and other pulse output terminals no need to output pulse, these terminals also can be pulse direction terminal. If Y0 no needs to output pulse, it also can be pulse direction terminal.

Note:

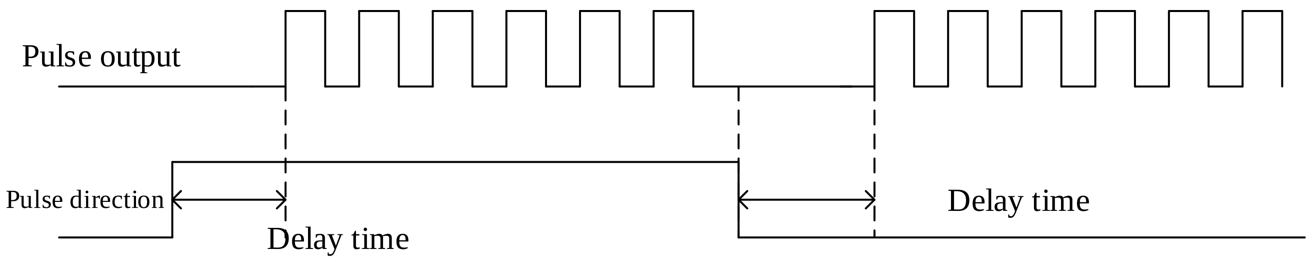
1: please do not choose the terminal over the actual output terminal number.

2: fit for PLSR, PLSF, ZRN.

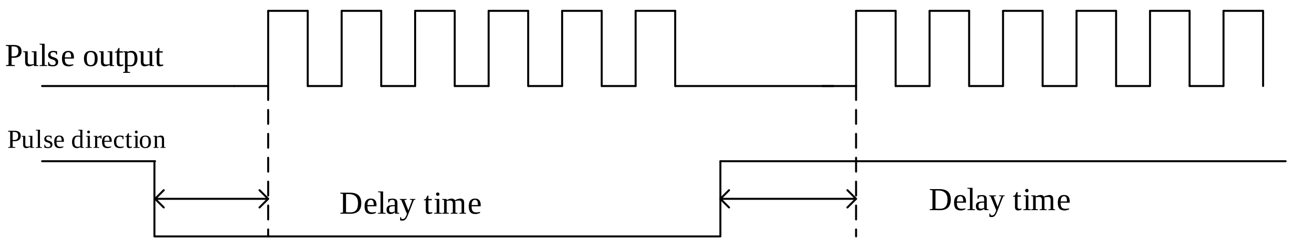
- Delayed time of pulse direction

When it is sending forward direction pulse, it will set ON the direction terminal first, then output the pulse after the delay time. When it is sending reverse direction pulse, it will set OFF the direction terminal first, then output the pulse after the delay time.

Y0 axis-Common-Pulse direction terminal	Y no terminal
Y0 axis-Common-Delayed time of pulse direction (ms)	10
Y0 axis-Common-Gear clearance positive compensation	0



Pulse start, forward pulse switch to reverse pulse



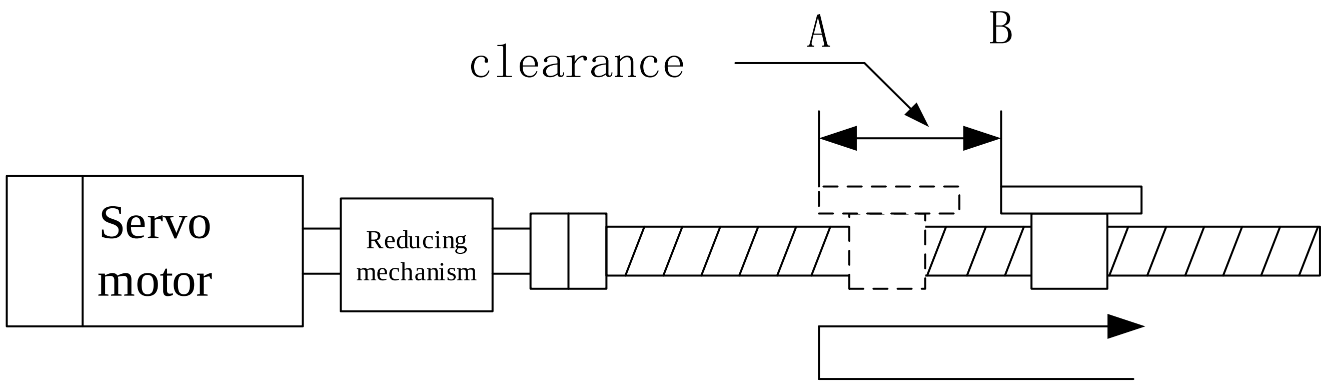
Reverse pulse switch to forward pulse

As the pulse output terminal is high-speed optocoupler(response time below 5us), other terminals are normal optocoupler(response time below 0.2ms)(such as XD3-32T-E) or relay output(about 10ms)(such as XD3-24R-E), the direction terminal will output after pulse terminal, so the direction terminal must be triggered first, then delay some time to output pulse. This can avoid the pulse error caused by direction switch lag(forward pulse switch to reverse pulse or reverse pulse switch to forward pulse).
The default pulse direction delay time is 10ms, user can adjust the time according to the terminal output type and scanning period(Y0 and Y1 response time is 5us, other transistor terminal is 0.2ms, relay output is 10ms).

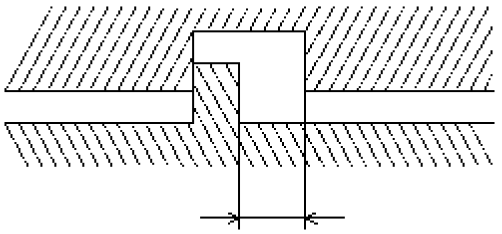
Note: suitable for PLSR, PLSF, ZRN.

- Gear clearance positive compensation

When the work table finished reverse moving and switched to forward moving, there is clearance between table and ball screw, it will cause the actual moving distance is less than setting value, this parameter can delete this error.

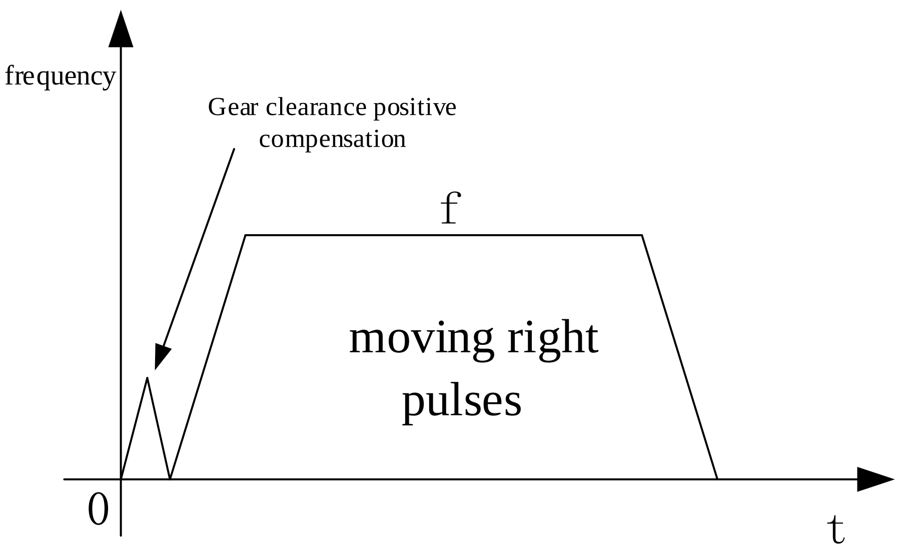


Mechanical structure



Mechanical clearance structure

The table moves from right to left, when the table left side moves to position A, it will stop and moves from left to right. As the ball screw clearance, it cannot move right for some pulses, and the actual moving distance is less than setting value. If there is no clearance, it will move from A to B. in order to delete the error, we must send some pulses before moving right, and then send the actual moving right pulses.



Note:

※1: it only execute the gear clearance positive compensation when the direction of last and present pulse segment is different.

※2: the gear clearance positive compensation pulses should output in separate segment, it cannot output in the same pulse segment of moving right pulses.

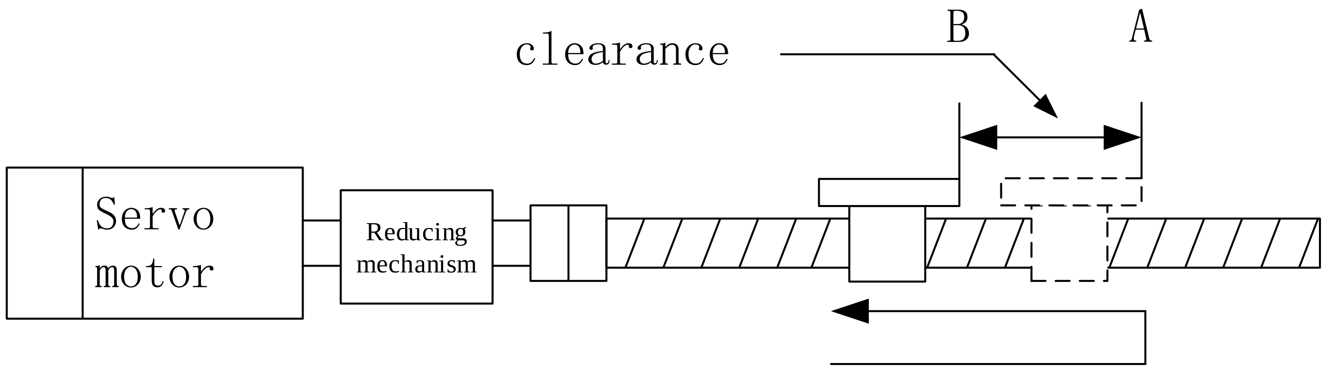
※3: the gear clearance positive compensation pulses will not be counted in pulse cumulative registers (such as HSD0 for Y0 output terminal).

※4: suitable for instruction PLSR, PLSF, ZRN.

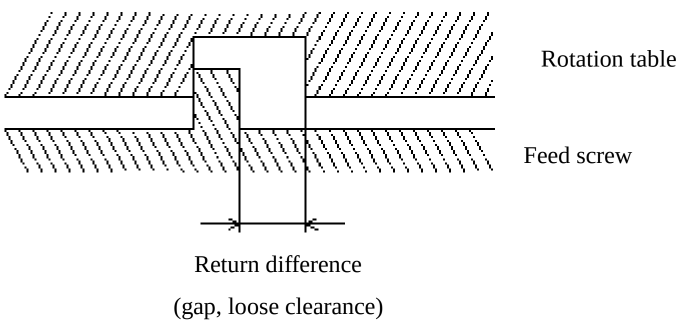
※5: the unit of gear clearance positive compensation is decided by pulse unit.

- Gear clearance negative compensation

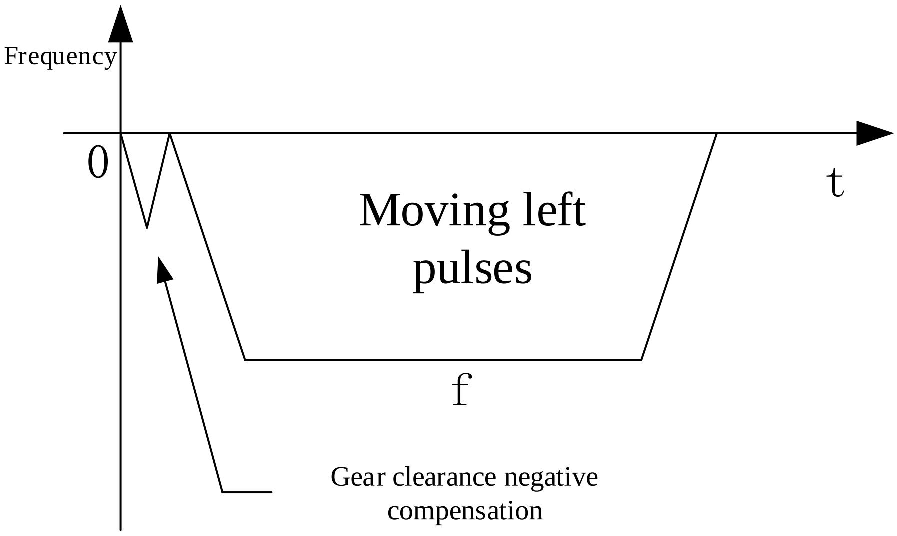
When the work table finished forward moving and switched to reverse moving, there is clearance between table and ball screw, it will cause the actual moving distance is less than setting value, this parameter can delete this error.



Mechanical structure



The table moves from left to right, when the table right side moves to position A, it will stop and moves from right to left. As the ball screw clearance, it cannot move left for some pulses, and the actual moving distance is less than setting value. If there is no clearance, it will move from A to B. in order to delete the error, we must send some pulses before moving left, and then send the actual moving left pulses.



Note:

※1: it only execute the gear clearance negative compensation when the direction of last and present pulse segment is different.

※2: the gear clearance negative compensation pulses should output in separate segment, it cannot output in the same pulse segment of moving left pulses.

※3: the gear clearance negative compensation pulses will not be counted in pulse cumulative registers (such as HSD0 for Y0 output terminal).

※4: suitable for instruction PLSR, PLSF, ZRN.

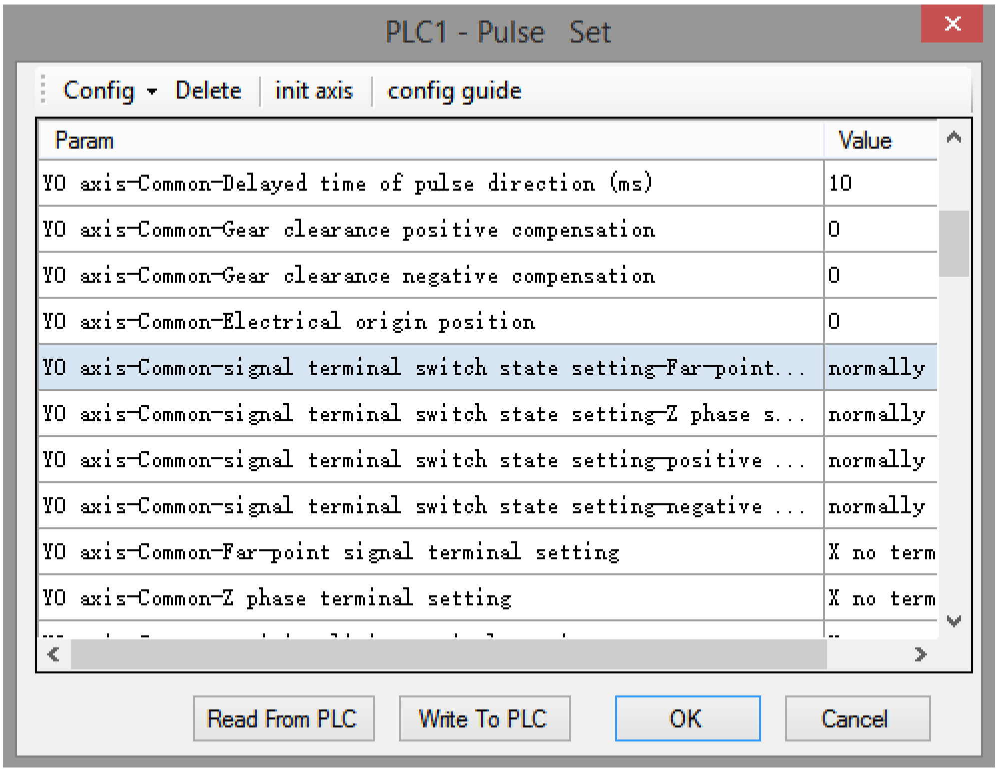
※5: the unit of gear clearance negative compensation is decided by pulse unit.

- Electrical origin position

This parameter cannot modify.

- Signal terminal switch state-point switch state setting

It can set the state of the signal collection terminal. The terminal state can be normally open and normally close. The signal terminal includes origin point, Z phase switch, positive limit switch, negative limit switch.



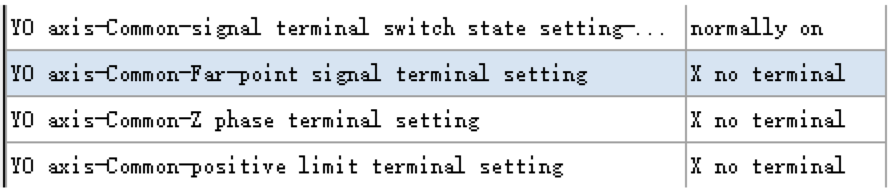
Take origin point as an example.

Normally open: the mechanical origin switch is normally open(OFF) when it returns origin, it will be ON when the machine touches the origin switch.

Normally close: the mechanical origin switch is normally close(ON) when it returns origin, it will be OFF when the machine touches the origin switch.

- Origin point signal terminal setting

The PLC input point of mechanical origin switch.



Note:

※1: the input point range cannot over actual input of PLC.

※2: only fit for mechanical return origin instruction ZRN.

※3: the origin point can be PLC input terminal, if the terminal is for external interruption input, the returning mechanical origin process will be operated as interruption and the precision will be improved (Z phase return origin has no effect). If the terminal is not for external interruption, the returning origin process will be affected by PLC scanning period (Z phase return origin has no effect).

※4: please refer to appendix 4 for details of external interruption terminal.

- Z phase terminal setting

When returning mechanical origin, it will move reverse slowly with slow speed and acceleration slop until reach origin creep speed, and it starts to count the Z phase signal at the moment of leaving the origin signal. Here can set the Z phase count input terminal.

Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal
Y0 axis-Common-negative limit terminal setting	X no terminal

Note:

※1: only fit for mechanical return origin instruction ZRN.

※2: Z phase terminal only can be PLC external interruption input. As the pulse width of Z phase signal outputting from servo drive is very narrow, normal PLC input filter time is 10ms, the Z phase signal only can be catched through high speed optical coupler input. If using normal terminal, it cannot catch the Z phase signal and cause returning mechanical origin error.

※3: Z phase input terminals:

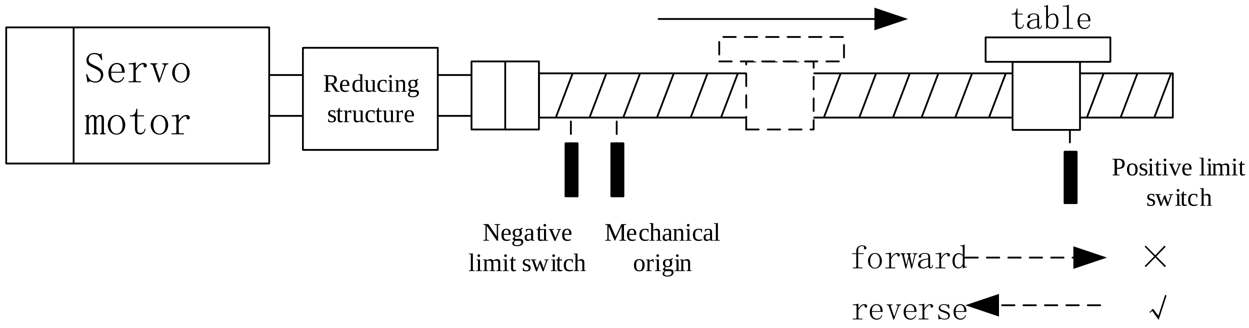
PLC model	Z phase input terminal
XD2-16/24/32/48/60	X2、X3、X4、X5、X6、X7
XD3-16/24/32/48/60	X2、X3、X4、X5、X6、X7
XD5-24/32	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XD5-24/32T4	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XD5-48/60	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XD5-48/60T6	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XDM-24/32T4	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XDM-60T4	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XDM-60T10	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XDC-24/32	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XDC-48/60	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XD5E-30T4	X2、X3、X4、X5、X6、X7、X10、X11、X12、X13
XL3-16	X2、X3、X4、X5、X6、X7

- Positive limit terminal setting

When the machine is returning origin (instruction ZRN), to prevent the table from moving beyond the range, the protection terminal is installed at both ends of the range. Please refer to ZRN instruction for details.

Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal

When the instruction ZRN, PLSR, PLSF are executed, if the forward pulse touches positive limit, the pulse will stop in slow stop mode (make sure the positive limit switch is in triggered state after pulse stop). The pulse will be always prohibitted when the positive limit switch is triggered, but the reverse pulse can be triggered.



Notes:

※1: the input terminal cannot over the PLC actual input range.

※2: make sure the positive limit block is long enough, to ensure the positive limit switch is still triggered after pulse stop. Otherwise the table will strick the machine when the forward pulse is triggered again.

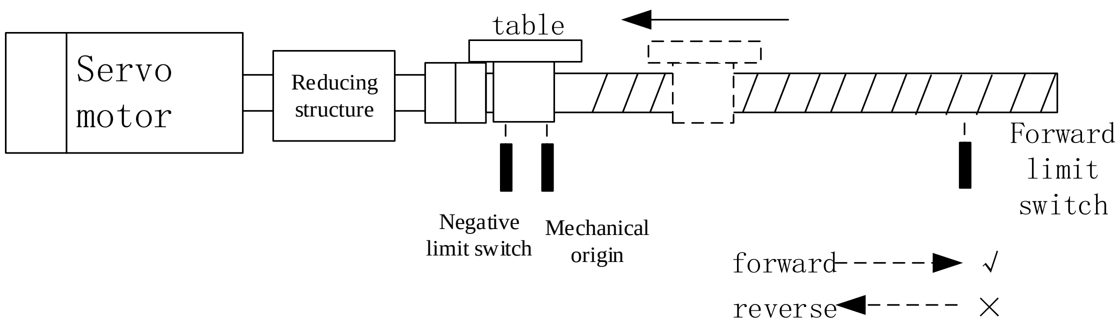
※3: fit for instruction PLSR, PLSF, ZRN.

- Negative limit terminal setting

When the machine is returning origin (instruction ZRN), to prevent the table from moving beyond the range, the protection terminal is installed at both ends of the range. Please refer to ZRN instruction for details.

Y0 axis-Common-positive limit terminal setting	X no terminal
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0

When the instruction ZRN, PLSR, PLSF are executed, if the reverse pulse touches negative limit, the pulse will stop in slow stop mode (make sure the negative limit switch is in triggered state after pulse stop). The pulse will be always prohibitted when the negative limit switch is triggered, but the forward pulse can be triggered.



Notes:

※1: the input terminal cannot over the PLC actual input range.

※2: make sure the negative limit block is long enough, to ensure the negative limit switch is still triggered after pulse stop. Otherwise the table will strick the machine when the reverse pulse is triggered again.

※3: fit for instruction PLSR, PLSF, ZRN.

- Zero clear CLR output setting

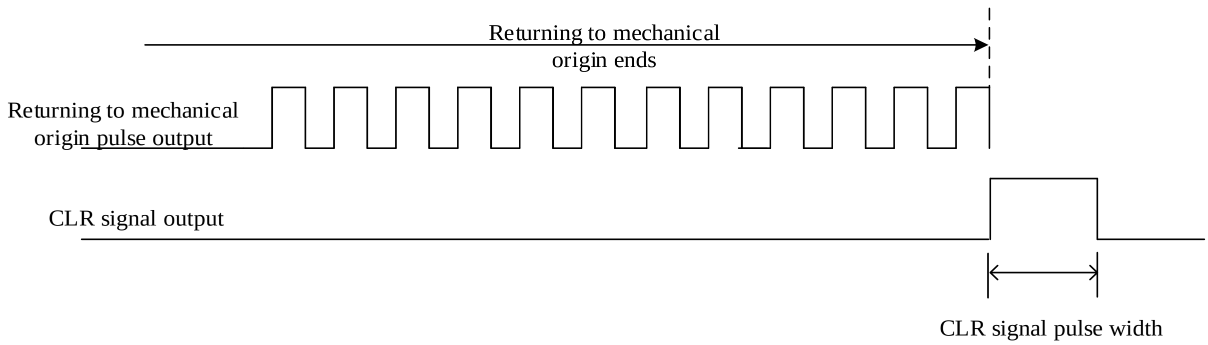
It will output the signal after the returning mechanical origin ends. This signal can send to other device such as servo drive to clear the servo motor error counter, then copy the mechanical origin position to present position to finish the returning to zero process.

Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0

- CLR signal delayed time

The CLR signal pulse width time, the unit is ms. The range is 0 to 32767 (default is 20ms).

Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0



CLR signal diagram

Notes:

※1: only fit for instruction ZRN.

※2: please use PLC main unit output terminal for CLR signal output.

※3: please do not set too small CLR signal delay time, otherwise the servo drive cannot receive too narrow pulse width signal.

- Return speed VH

When it starts to run ZRN, the table accelerates to return speed VH and moves towards mechanical origin, this can shorten the returning time.

Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0

Notes:

※1: only fit for instruction ZRN.

※2: when the ZRN starts, VH accelerates as setting acceleration slop, then decelerates as setting deceleration slop when touching the near origin signal or origin signal.

※3: if there is no near origin signal, please do not set the VH speed too large, otherwise it will cause mechanical oscillation as the VH speed quickly decelerating to zero.

※4: if there is no near origin signal, please do not set the VH speed too large and deceleration slop too small, otherwise it will cause the table out of origin signal and even touching the reverse limit signal when decelerating to zero as the table decelerating time is too long.

- Creeping speed VC

When it meets the origin signal, the start speed decelerates to zero, after delay time, it reverse accelerates to creeping speed. It will stop the creeping speed at once when the work table leaves origin signal. As the stop position of work table leaving origin signal is mechanical origin, in order to improve mechanical origin precision, generally, the creeping speed is small.

Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0

Note:

※1: only fit for instruction ZRN.

※2: the creeping speed acc/dec slope is same to setting acceleration/deceleration slope. It will urgent stop or count the Z phase pulse numbers when leaving origin signal.

※3: Do not set the creeping speed over 100r/min, otherwise it will affect the high precision returning to origin.

※4: Do not set the creeping speed larger than or equal to returning to origin speed VH.

- Mechanical zero position

The present position after returning to mechanical origin ends. Take axis Y0 as an example, set the present position value HSD0(double word) or HSD2(double word) after returning to mechanical origin.

Generally, the present value of mechanical origin is 0, it also can be set to other value. After the returning to mechanical origin, the related cumulative pulse register will be updated to setting value.

Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0

Note:
※1: only fit for instruction ZRN.
※2: if the pulse unit of axis Y0 is set to pulse numbers, the mechanical origin setting value will be written in HSD0(double word) after returning to mechanical origin. If the pulse unit of axis Y0 is set to equivalent (1mm, 0.1mm, 0.01mm, 1um), the mechanical origin setting value will be written in HSD2(double word) after returning to mechanical origin.

- Z phase numbers

When it meets the origin signal, the start speed decelerates to zero, after delay time, it reverse accelerates to creeping speed. It can count the servo motor Z phase pulse when the work table leaves origin signal. It will stop creeping speed at once when the count value reaches setting Z phase pulse numbers, and mechanical returning to origin ends.

Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20

Note:
※1: only fit for instruction ZRN.
※2: if the Z phase numbers is set to 0, it means Z phase pulse catching function is invalid, it will stop at once when leaving origin with creeping speed and returning to origin ends.
※3: please avoid the interval between work table leaving origin signal and Z phase signal is too short, otherwise the origin position will be error.
※4: Z phase signal maybe changed after install the servo motor again, please adjust it.
※5: if it is stepper motor, the external proximity switch signal can be used to Z phase signal.

- Grinding wheel radius(polar)

This parameter cannot be used right now.

Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0

Group 1 parameters (group 2 to 4 parameters please refer to group 1)

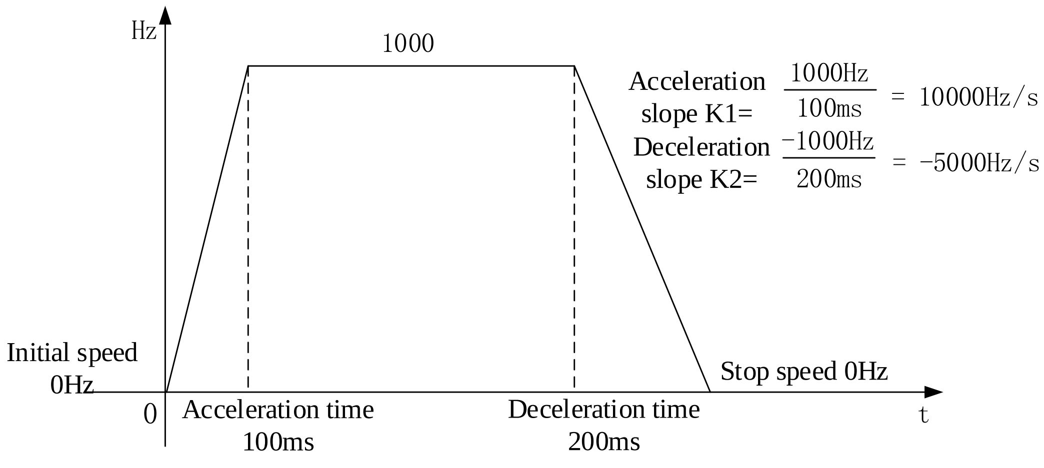
- Pulse default speed/acceleration time of default pulse speed/deceleration time of default pulse speed(ms)

The three parameters and initial speed, stop speed are used to define the pulse acceleration and deceleration slop. The pulse default speed unit is decided by pulse unit parameter.

Y0 axis-group 1-Pulse default speed	0
Y0 axis-group 1-Acceleration time of Pulse default s...	0
Y0 axis-group 1-Deceleration time of pulse default s...	0

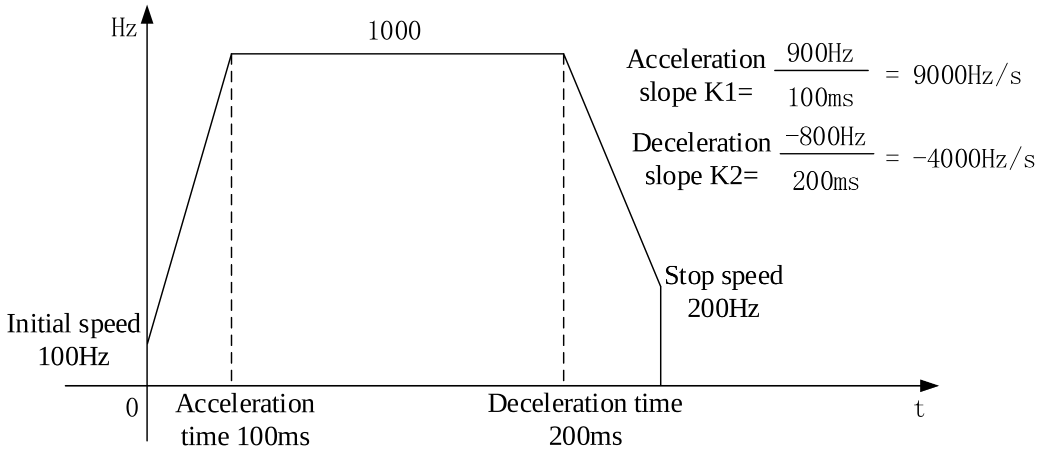
Example 1:

When the pulse unit is pulse numbers, pulse default speed is 1000Hz, acceleration time of pulse default speed is 100ms, deceleration time of pulse default speed is 200ms, initial speed is 0Hz, stop speed is 0Hz, it means the pulse frequency takes 100ms to increase 1000Hz and takes 200ms to decrease 1000Hz. If it accelerates from 0Hz to 5000Hz, the time is 5000/1000*100=500ms, if it decelerates from 5000Hz to 0Hz, the time is 5000/1000*200=1000ms.



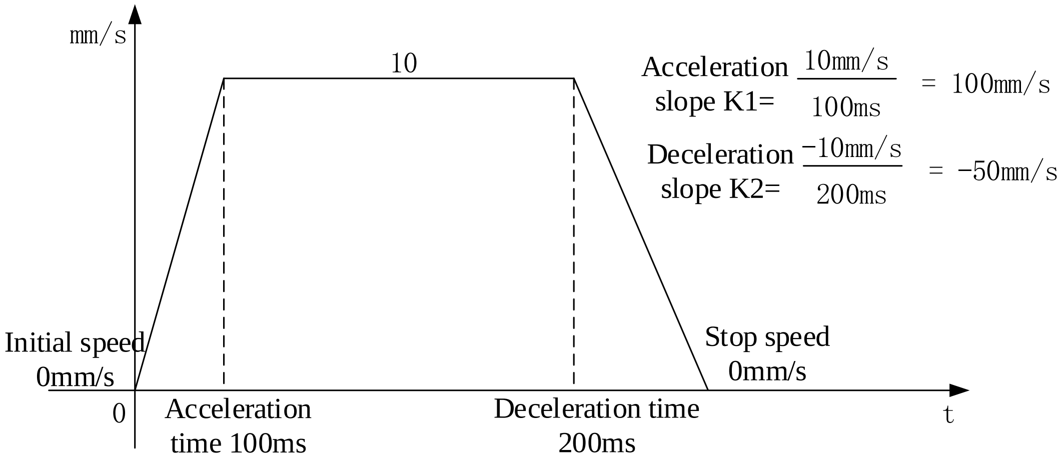
Example 2:

When the pulse unit is pulse numbers, pulse default speed is 1000Hz, acceleration time of pulse default speed is 100ms, deceleration time of pulse default speed is 200ms, initial speed is 100Hz, stop speed is 200Hz, it means the pulse frequency takes 100ms to increase (1000-100)=900Hz and takes 200ms to decrease (1000-200)=800Hz. If it accelerates from 0Hz to 5000Hz, the time is 5000/900*100=555ms, if it decelerates from 5000Hz to 0Hz, the time is 5000/800*200=1250ms.



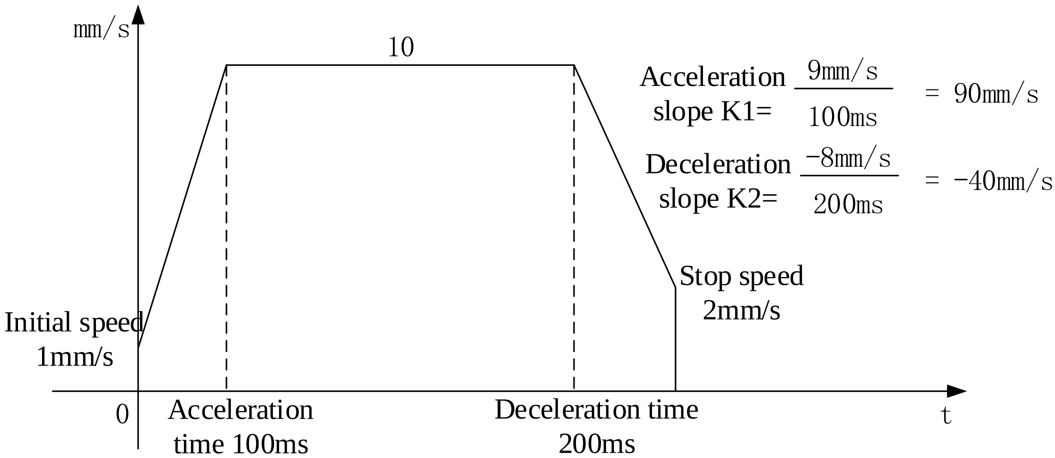
Example 3:

When the pulse unit is equivalent 1mm, pulse default speed is 10mm/s, acceleration time of pulse default speed is 100ms, deceleration time of pulse default speed is 200ms, initial speed is 0mm/s, stop speed is 0mm/s, it means the pulse frequency takes 100ms to increase 10mm/s and takes 200ms to decrease 10mm/s . If it accelerates from 0 to 50mm/s, the time is 50/10*100=500ms, if it decelerates from 50mm/s to 0, the time is 50/10*200=1000ms.



Example 4:

When the pulse unit is equivalent 1mm, pulse default speed is 10mm/s, acceleration time of pulse default speed is 100ms, deceleration time of pulse default speed is 200ms, initial speed is 1mm/s, stop speed is 2mm/s, it means the pulse frequency takes 100ms to increase (10-1)=9mm/s and takes 200ms to decrease (10-2)=8mm/s . If it accelerates from 0 to 50mm/s, the time is 50/9*100=555ms, if it decelerates from 50mm/s to 0, the time is 50/8*200=1250ms.



Note:

※1: the three parameters and initial speed, stop speed are used to define the acceleration and deceleration slope.

※2: the pulse acceleration slope is determined by the time accelerating from initial speed to default pulse speed, the pulse deceleration slope is determined by the time decelerating from default pulse speed to stop speed.

※3: the parameter is fit for instruction PLSR, PLSF, DRVI, DRVA, ZRN.

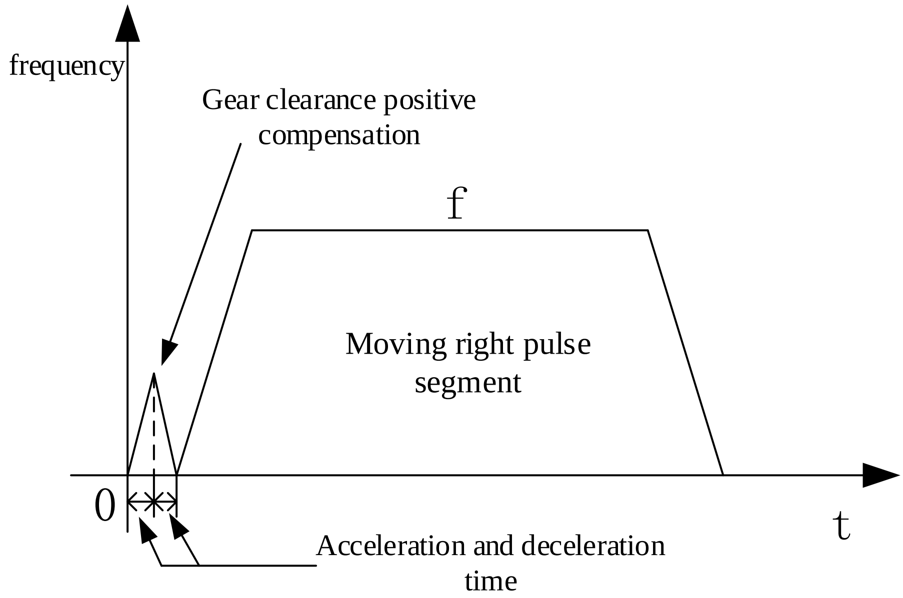
※4: initial speed and stop speed must be less than rated speed.

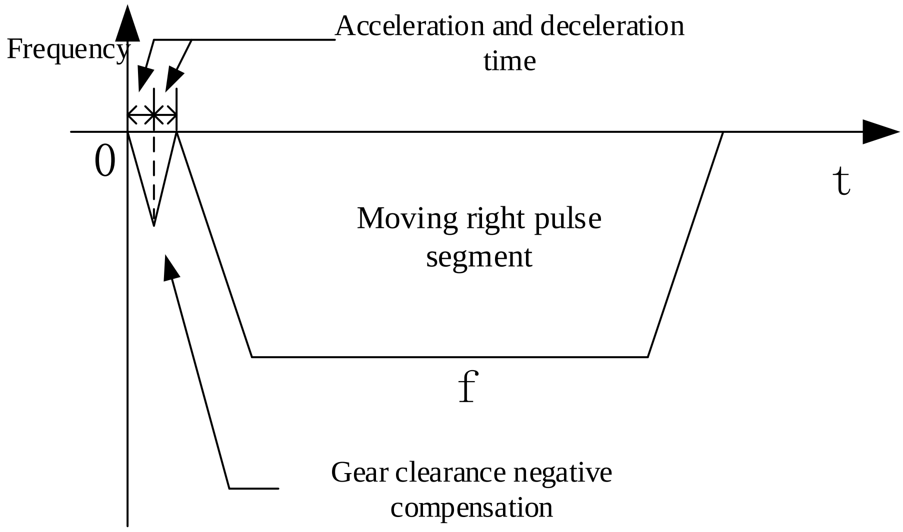
※5: the pulse default speed is not related to the pulse frequency, it is only used to set the acceleration and deceleration slope. But when the pulse frequency is 0, it will output pulse as the default pulse speed.

- Acceleration and deceleration time (ms)

This time is for gear clearance positive and negative compensation. This acceleration and deceleration time is same whatever how many is the gear clearance compensation quantity, the unit is ms.

Y0 axis-group 1-Deceleration time of pulse default s...	0
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	0





Note:

※1: the acceleration time and deceleration time is same.

※2: the acceleration and deceleration time is fixed value whatever how many is the gear

clearance compensation.
※3: this parameter is fit for instruction PLSR, PLSF, DRVI, DRVA, ZRN.

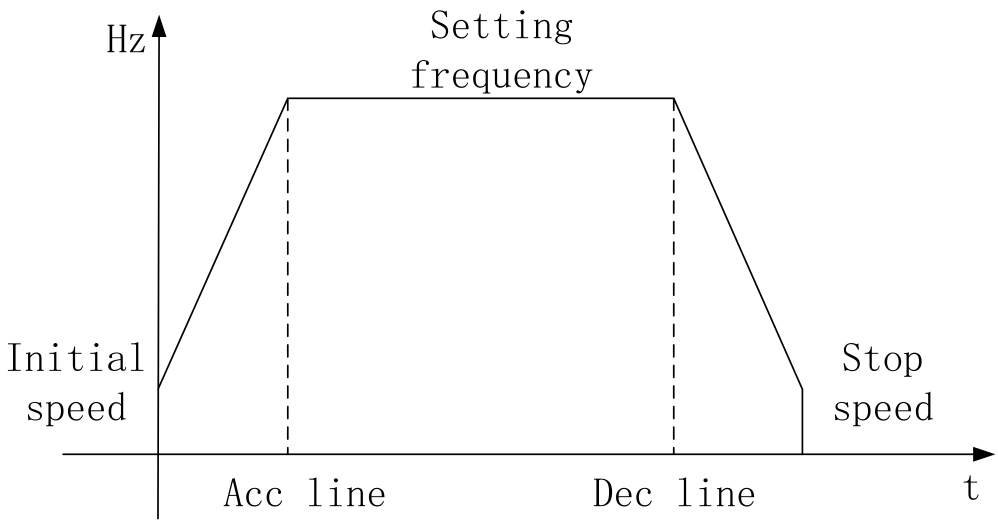
- Pulse acc/dec mode

The pulse acceleration mode accelerating from initial speed to setting frequency and pulse deceleration mode decelerating from setting frequency to initial speed.

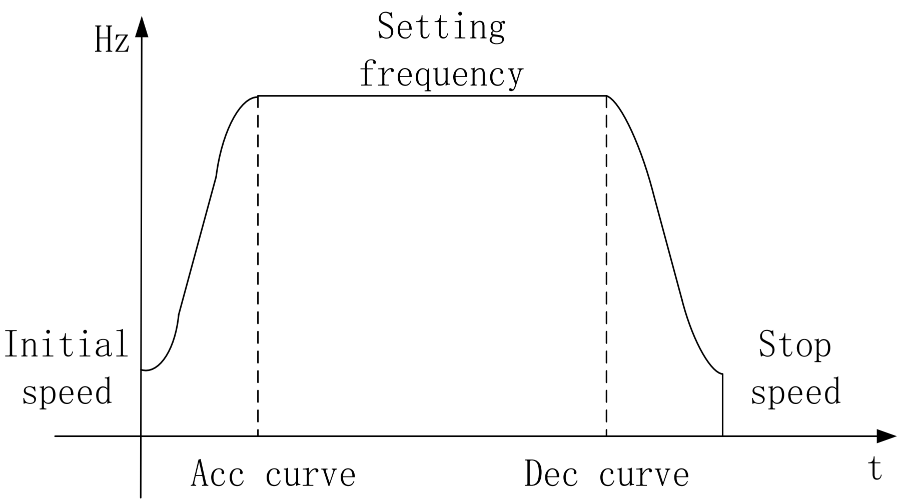
Y0 axis-group 1-Deceleration time of pulse default s...	0
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	0
Y0 axis-group 1-Initial speed	0

The pulse acc/dec mode include linear mode, S curve mode and sine curve mode.

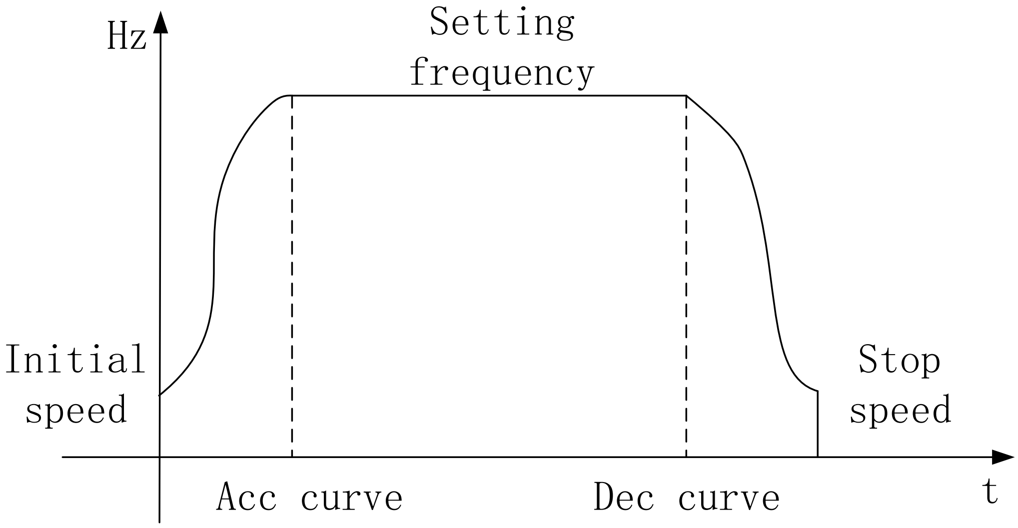
Linear mode: the speed changing for accelerating or decelerating is line.



S-curve mode: the speed changing for accelerating or decelerating is S-curve.



Sine curve mode: the speed changing for accelerating or decelerating is sine curve.



Sine-curve mode is fit for the receiving of stepper motor and servo motor and improve the run performance of stepper motor and servo motor. The details please refer to S-curve acceleration and deceleration.

Note: this parameter is fit for the instruction PLSR, PLSF, ZRN.

- Max speed

When all the pulse instructions in the program is executing parameter group 1, the highest pulse frequency cannot over the max speed, if it is over the max speed, PLC will run as the max speed.

Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	0
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0

Note:

※1: the max speed unit is changing as pulse unit(pulse number or equivalent).

※2: XD all series PLC pulse output frequency max speed is 200Khz. The max speed cannot over this value.

※3: when the pulse unit is equivalent, the transformed pulse frequency maybe very large and over max speed, please pay attention.

※4: User must set the max speed when using pulse instruction, otherwise the pusle cannot output normally.

※5: this parameter is fit for instruction PLSR, PLSF, ZRN.

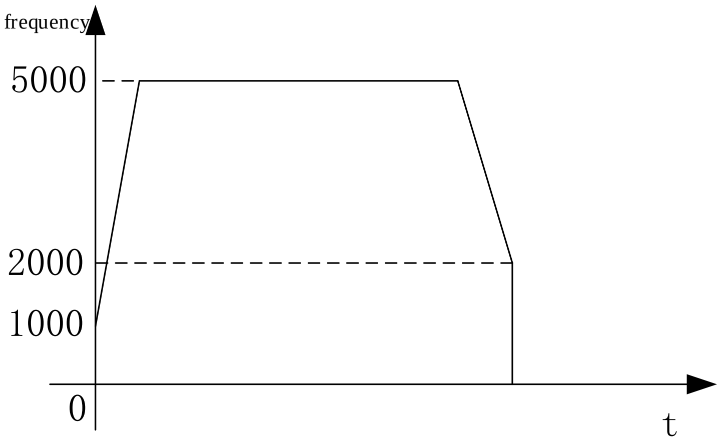
- Initial speed and stop speed

The pulse start frequency and end frequency for the pulse instruction start and completion. Generally, the initial and stop speed is 0, but for some special occasions, the pulse needs to start with non-zero speed and complete with non-zero speed.

Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	0
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50

For example, it needs to output 30000 pulses, and accelerates from 1000Hz, takes 100ms to reach 5000Hz. And it decelerates from 5000Hz, takes 50ms to reach 2000Hz, and the pulse will complete here. The configuration is shown as below:

Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	1000
Y0 axis-group 1-stop speed	2000



Note:

※1: the pulse unit of initial speed and stop speed is changing as the pulse number or equivalent.

※2: the initial speed and stop speed must be less than the max speed.

※3: when the pulse unit is equivalent, the transformed pulse frequency maybe very large and over max speed, please pay attention.

※4: make sure to set the initial speed and stop speed for pulse instruction, the default value is 0.

※5: this parameter is fit for instruction PLSR, PLSF, ZRN.

- Follow parameters

The FOLLOW instruction can make the slave axis servo motor or stepper motor following the master axis motor motion (which means the slave axis motion is consistant with main axis). The parameters include FOLLOW performance and FOLLOW feedforward compensation.

The FOLLOW instruction is motion following function, it can control the servo or stepper motor by outputting pulse according to motor encoder feedback.

FOLLOW performance: the function is similar to servo drive rigidity function. The smaller the value, the smaller the follow rigidity (delay time is long), the larger the value, the larger the follow rigidity (delay time is short).

FOLLOW feedforward compensation: there is delay time from receiving pulse to outputting pulse. In order to reduce the delay time, it can set the feedforward compensation, make the pulse a little advanced. But if the feedforward parameter is too large, it will enter infinite loop, the motor will vibrate when the follow process ends.

Y0 axis-group 1-stop speed	2000
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0
Y0 axis-group 2-Pulse default speed	0

1-2-1-4. Pulse interruption flag

Pulse instruction PLSR can set up to 100 segments of pulse. It can produce a interruption flag after each pulse segment completion.

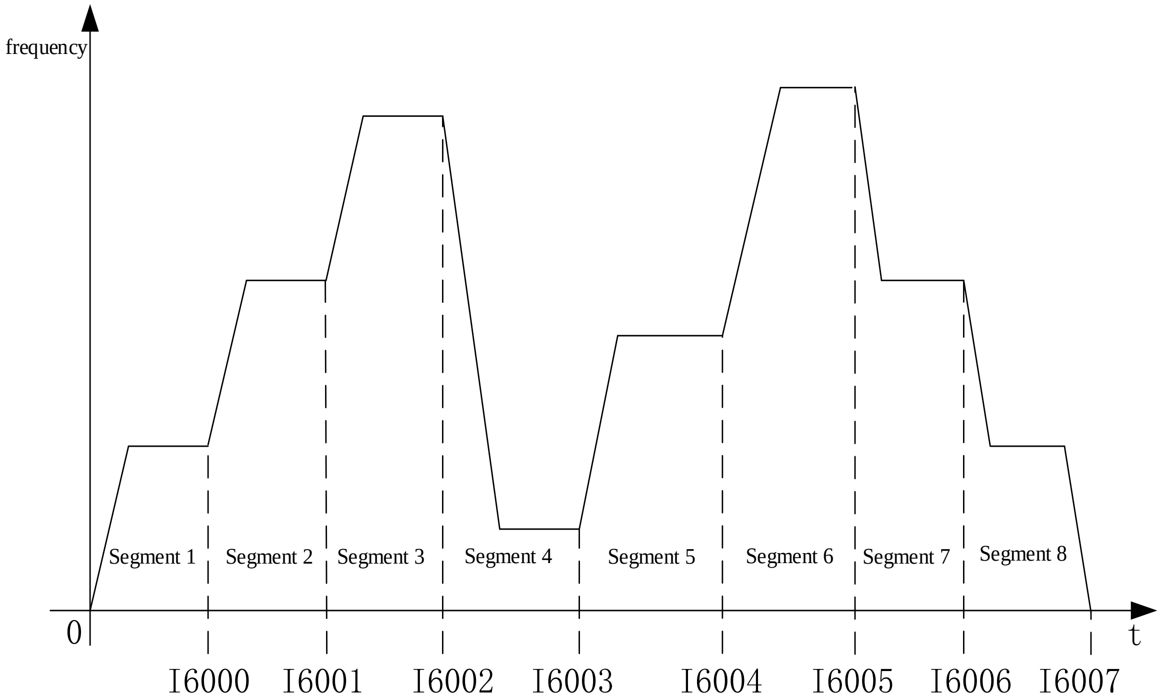
Note: each pulse segment has only one related interruption flag, whatever how is the pulse configuration jump setting, the interruption flag will be executed when this pulse segment is running.

Interruption flag for each pulse segment:

Interruption flag	Pulse axis	Notes
I60**(I6000~I6099)	PLS+0 (pulse)	Y0 axis 100 pulse segments interruption
I61**(I1000~I6199)	PLS+1 (pulse)	Y1 axis 100 pulse segments interruption
I62**(I6200~I6299)	PLS+2 (pulse)	Y2 axis 100 pulse segments interruption
I63**(I6300~I6399)	PLS+3 (pulse)	Y3 axis 100 pulse segments interruption
I64**(I6400~I6499)	PLS+4 (pulse)	Y4 axis 100 pulse segments interruption
I65**(I6500~I6599)	PLS+5 (pulse)	Y5 axis 100 pulse segments interruption
I66**(I6600~I6699)	PLS+6 (pulse)	Y6 axis 100 pulse segments interruption
I67**(I6700~I6799)	PLS+7 (pulse)	Y7 axis 100 pulse segments interruption
I68**(I6800~I6899)	PLS+8 (pulse)	Y8 axis 100 pulse segments interruption
I69**(I6900~I6999)	PLS+9 (pulse)	Y9 axis 100 pulse segments interruption

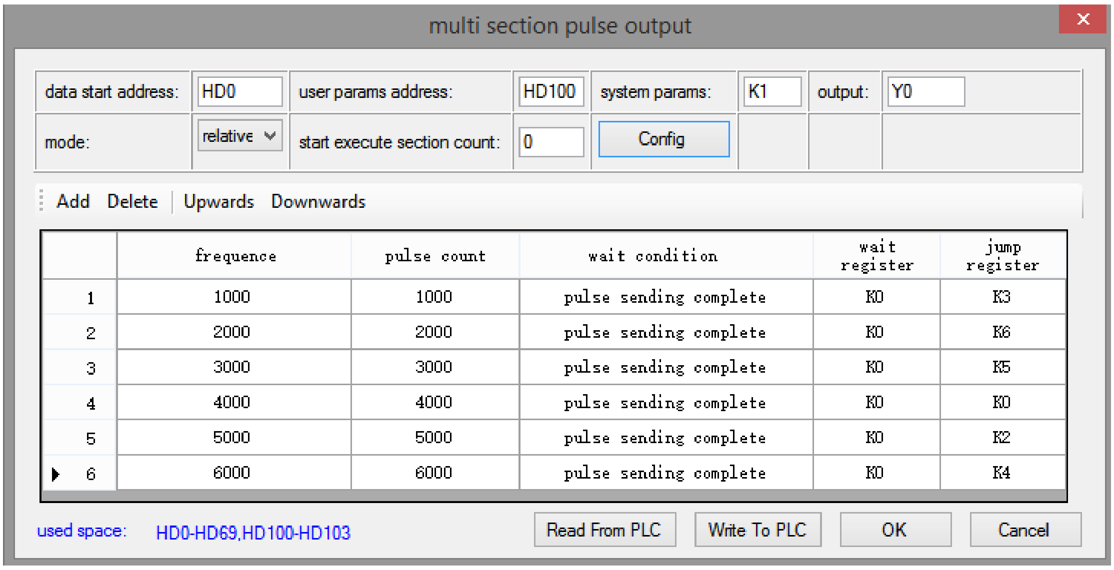
Example 1:

Now PLC has 8 pulse segments and executes from the first segment, the pulse output terminal is Y0, the interruption is shown as below:



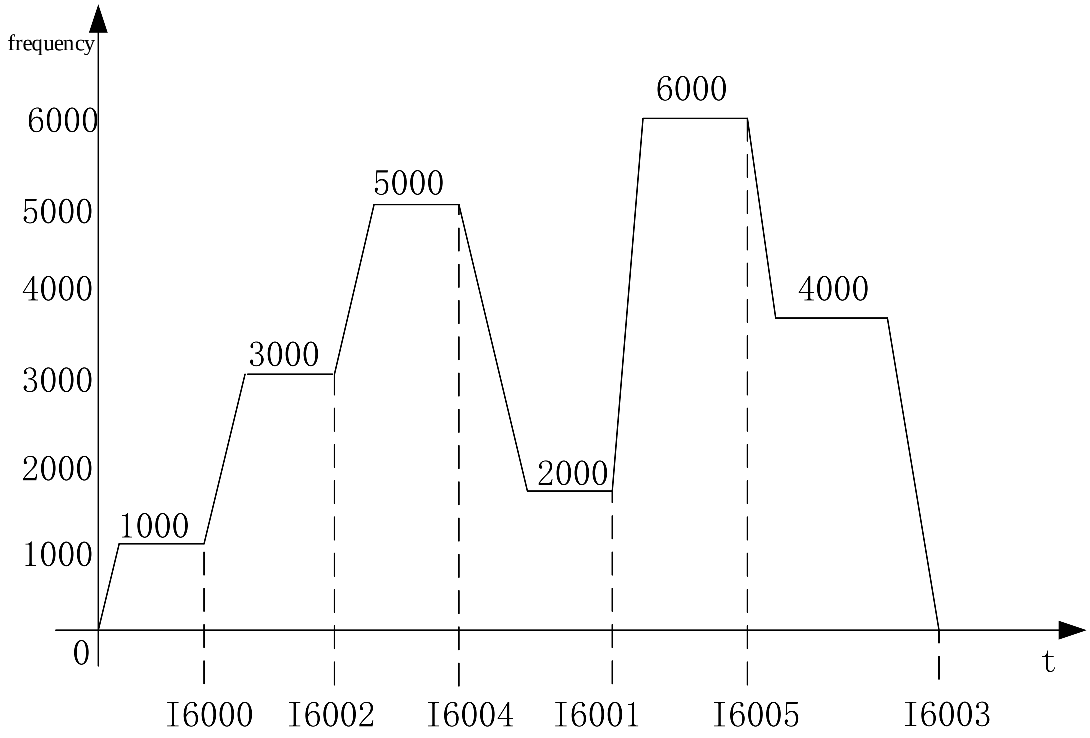
Example 2:

The PLC has 6 pulse segments, the pulse output terminal is Y0, but the pulse is not continuous outputting.

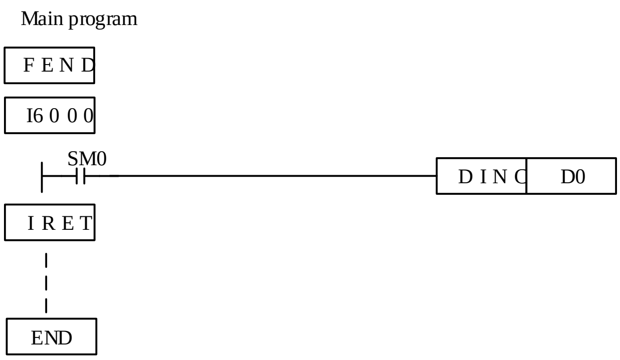


	frequence	pulse count	wait condition	wait register	jump register
1	1000	1000	pulse sending complete	K0	K3
2	2000	2000	pulse sending complete	K0	K6
3	3000	3000	pulse sending complete	K0	K5
4	4000	4000	pulse sending complete	K0	K0
5	5000	5000	pulse sending complete	K0	K2
6	6000	6000	pulse sending complete	K0	K4

As the pulse configuration table, the pulse outputting sequence is segment 1, 3, 5, 2, 6, 4. The interruption flag is I6000, I6002, I6004, I6001, I6005, I6003, please see below diagram:



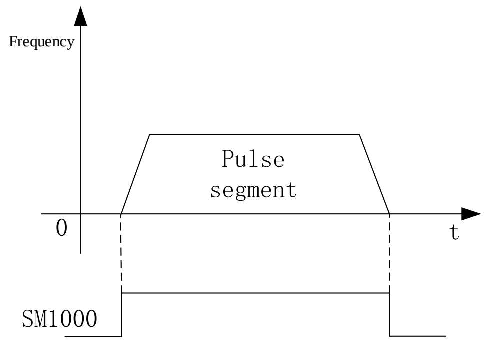
Note: the program format is same for pulse interruption and external interruption.



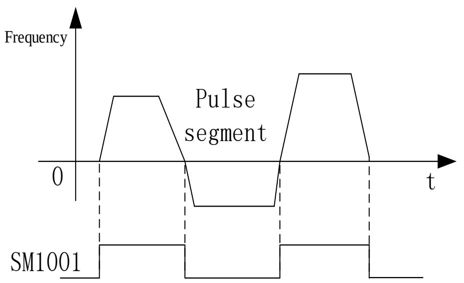
1-2-1-5. Pulse monitoring coil and register

- **Pulse sending flag**

No.	Coil	Axis no.	Note
1	SM1000	PULSE_1	The coil is ON when the pulse is sending, the coil will be OFF when the pulse sending ends. The falling edge of coil can judge whether the pulse sending is completed.
2	SM1020	PULSE_2	
3	SM1040	PULSE_3	
4	SM1060	PULSE_4	
5	SM1080	PULSE_5	
6	SM1100	PULSE_6	
7	SM1120	PULSE_7	
8	SM1140	PULSE_8	

9	SM1160	PULSE_9	
10	SM1180	PULSE_10	

➢ **Pulse sending direction flag**

No.	Coil	Axis no.	Note
1	SM1001	PULSE_1	When the pulse number is positive value and forward direction, the coil is ON, when the pulse number is negative value and reverse direction, the coil is OFF. 
2	SM1021	PULSE_2	
3	SM1041	PULSE_3	
4	SM1061	PULSE_4	
5	SM1081	PULSE_5	
6	SM1101	PULSE_6	
7	SM1121	PULSE_7	
8	SM1141	PULSE_8	
9	SM1161	PULSE_9	
10	SM1181	PULSE_10	

➢ **High speed pulse special regsiter HSD (latched)**

No.	Function	Note	Axis no.
HSD0	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_1
HSD1	Cumulative pulses high 16-bit		
HSD2	Cumulative pulses low 16-bit	The unit is equivalent	
HSD3	Cumulative pulses high 16-bit		
HSD4	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_2
HSD5	Cumulative pulses high 16-bit		
HSD6	Cumulative pulses low 16-bit	The unit is equivalent	
HSD7	Cumulative pulses high 16-bit		
HSD8	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_3
HSD9	Cumulative pulses high 16-bit		
HSD10	Cumulative pulses low 16-bit	The unit is equivalent	
HSD11	Cumulative pulses high 16-bit		
HSD12	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_4
HSD13	Cumulative pulses high 16-bit		
HSD14	Cumulative pulses low 16-bit	The unit is equivalent	

HSD15	Cumulative pulses high 16-bit		
HSD16	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_5
HSD17	Cumulative pulses high 16-bit		
HSD18	Cumulative pulses low 16-bit	The unit is equivalent	
HSD19	Cumulative pulses high 16-bit		
HSD20	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_6
HSD21	Cumulative pulses high 16-bit		
HSD22	Cumulative pulses low 16-bit	The unit is equivalent	
HSD23	Cumulative pulses high 16-bit		
HSD24	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_7
HSD25	Cumulative pulses high 16-bit		
HSD26	Cumulative pulses low 16-bit	The unit is equivalent	
HSD27	Cumulative pulses high 16-bit		
HSD28	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_8
HSD29	Cumulative pulses high 16-bit		
HSD30	Cumulative pulses low 16-bit	The unit is equivalent	
HSD31	Cumulative pulses high 16-bit		
HSD32	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_9
HSD33	Cumulative pulses high 16-bit		
HSD34	Cumulative pulses low 16-bit	The unit is equivalent	
HSD35	Cumulative pulses high 16-bit		
HSD36	Cumulative pulses low 16-bit	The unit is pulse number	PULSE_10
HSD37	Cumulative pulses high 16-bit		
HSD38	Cumulative pulses low 16-bit	The unit is equivalent	
HSD39	Cumulative pulses high 16-bit		

1-2-2. Multi-segment pulse output [PLSR]

◆ Instruction overview

Multi-segment pulse output instruction.

Multi-segment pulse output [PLSR]			
16-bit	-	32-bit	PLSR
Execution condition	Rising /falling edge of the coil	Suitable model	XD, XL (except XD1, XL1)
Hardware	-	Software	-

◆ Operand

Operand	Function	Type
S0	Pulse data start address	32-bit double word
S1	User parameter start address	32-bit double word
S2	System parameter start address (1 to 4)	32-bit double word
D	Pulse output terminal	Bit

◆ Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●	●	●	●	●			
S1	●	●	●	●	●	●	●	●			
S2	●	●							●		

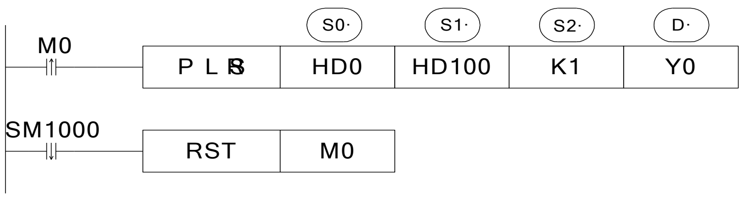
Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●					

*Note: D means D, HD. TD means TD, HTD. CD means CD, HCD, HSCD, HSD. DM means DM, DHM. DS means DS, DHS. M means M, HM, SM. S means S, HS. T means T, HT. C means C, HC.

Function and action

instruction format

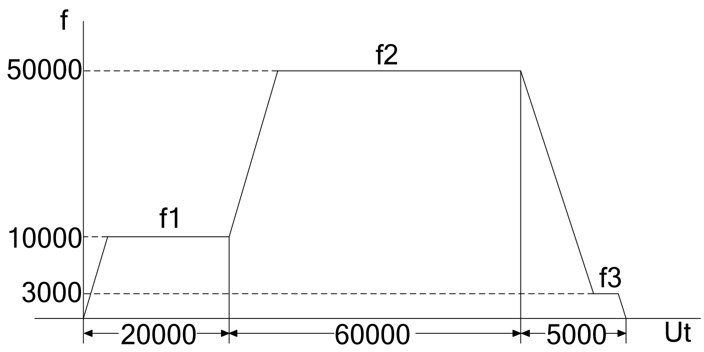


- S0【data start address】refer to chapter 1-2-1-1
- S1【user parameter start address】refer to chapter 1-2-1-2
- S2【system parameter group】K1~K4, refer to 1-2-1-3
- D【pulse output terminal】refer to chapter 1-1
- Pulse frequency range: 1Hz~100KHz. The value increasing means acceleration, the value

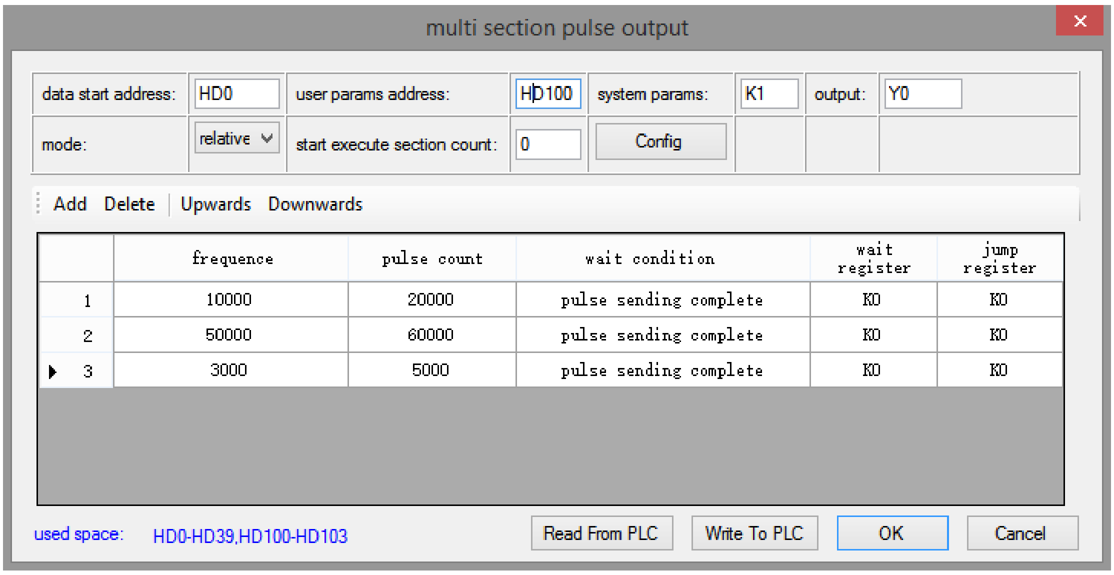
decreasing means deceleration, it is not related to the pulse direction.

- Pulse number: K-2,147,483,648 ~ K2,147,483,647, negative value means reverse direction. The acceleration and deceleration is set in system parameters, refer to chapter 1-2-1-3.
- When M0 is from OFF to ON, PLC executes the instruction PLSR, even M0 is cut off, the pulse will keep sending until end.
- If it needs to stop the pulse outputting, please use the instruction STOP.
- When the pulse is sending, the pulse sending flag of Y0 axis SM1000 is ON, when the pulse sending ends, SM1000 is OFF.
- Y0 cumulative pulse numbers are saved in HSD0(double word), the present pulse numbers are saved in SD1002(double word), more details please refer to chapter 6-5.
- For the instruction PLSR, if the frequency is changed when the pulse is sending, it will be effective at once. Other parameters will not be effective at once after changing, but be effective when the condition triggerring next time.
- **In absolute mode, if the pulse numbers and cumulative pulse numbers(HSD0) is equal, SM1000 has no action, there is no falling edge.**

Instruction mode



Pulse curve

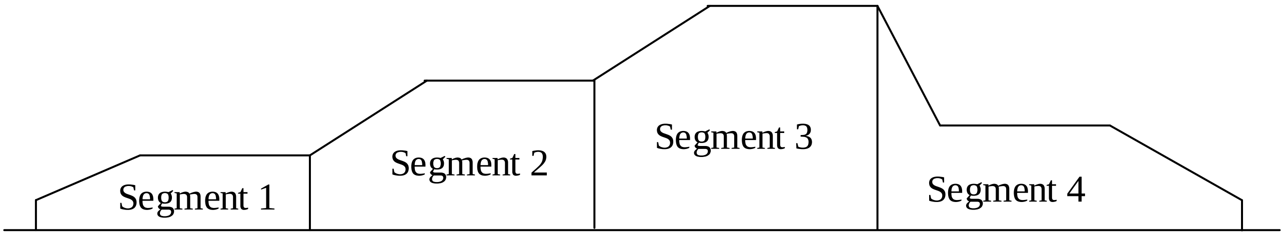


Pulse instruction parameter configuration table

How to do

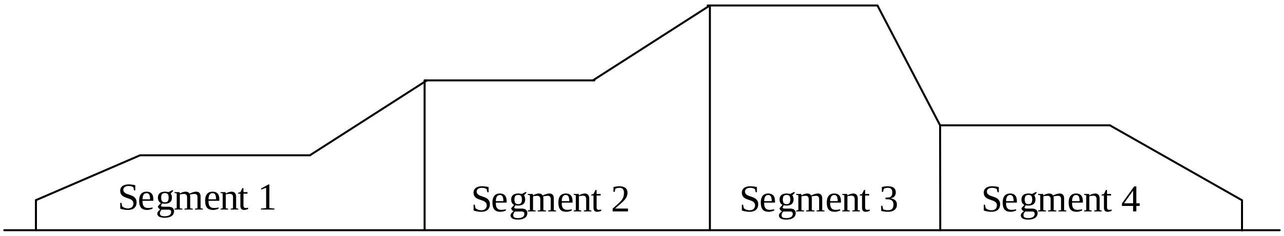
The following curves are set the parameters when the acceleration time is 0.

(1) Pulse segment completion mode division



- The segment are divided as above diagram
- Except the last segment, all the segments include rising, stable and falling part.
- The last segment includes rising or falling, stable and rising or falling part.

(2) Pulse segment subsequent mode division

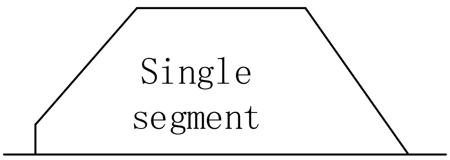


- The segment subsequent mode curve is shown as above diagram.
- It already switched to next segment speed when present segment ends. Except the first segment, other segments include stable part, rising or falling part.
- The first segment includes rising part or falling part, stable part, rising part or falling part.

(3) Single segment pulse curve

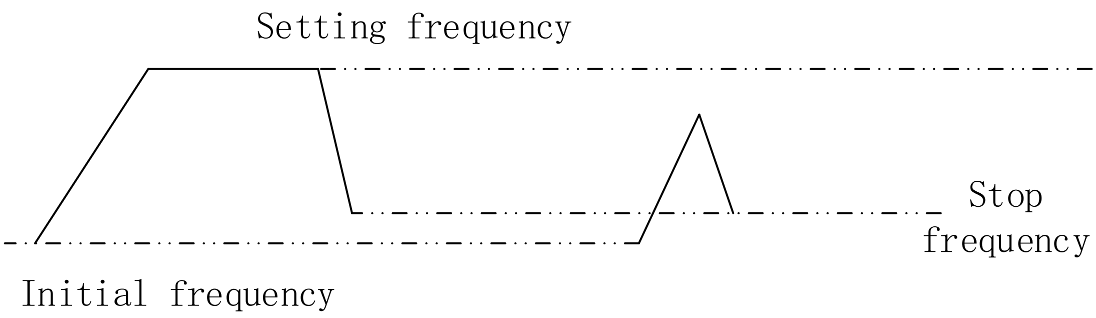
- The pulse numbers are enough

The pulse can reach the setting max frequency, the curve is trapezoid.

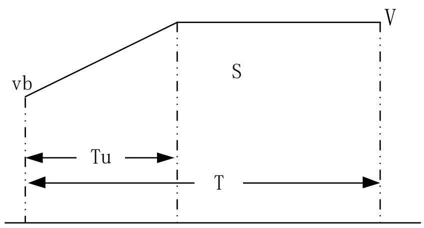


- The pulse numbers are not enough

The pulse curve is triangle.

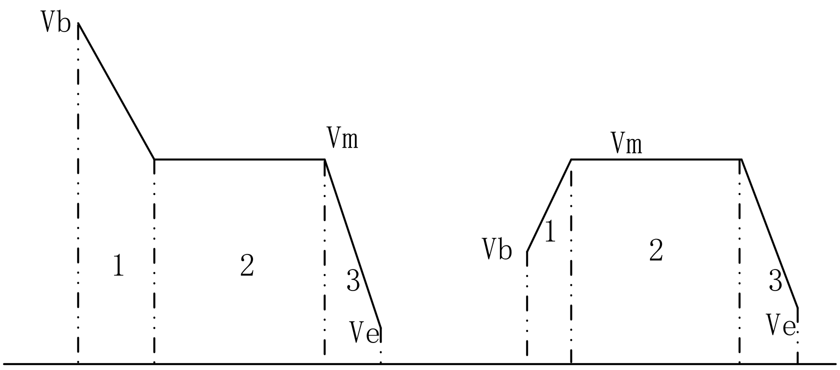


(4) One segment pulse outputting (not the last segment)



- V: setting present segment frequency
- S: present segment pulse numbers
- Vb: present segment initial frequency
- T: present segment pulse sending time
- Tu: pulse rising/falling time (Tu = (V-VB) / K , K is rising or falling slope).

(5) The last segment



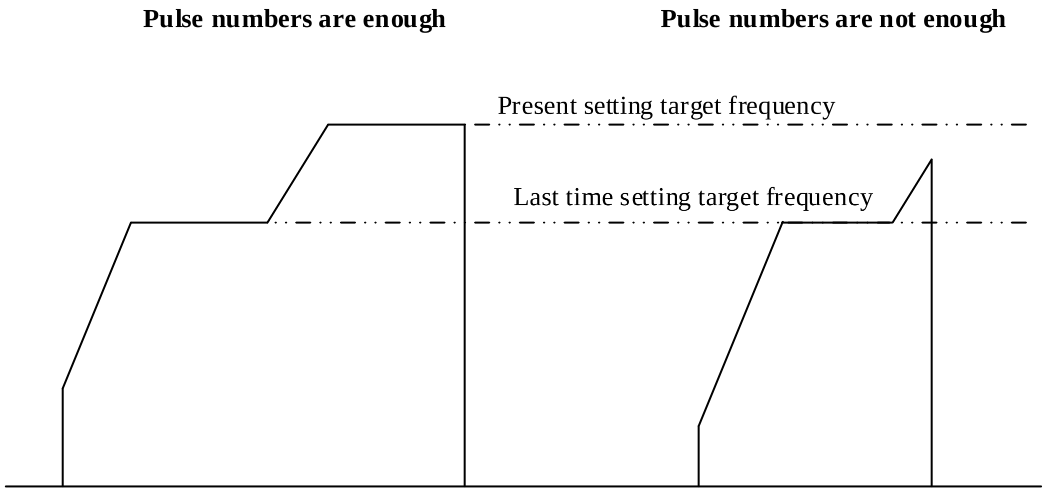
- The last segment includes rising/falling part, stable part, rising/falling part.

(6) the segment which the pulse numbers are 0

- If the present segment pulse frequency or pulse number is 0, it will output pulse as default speed.

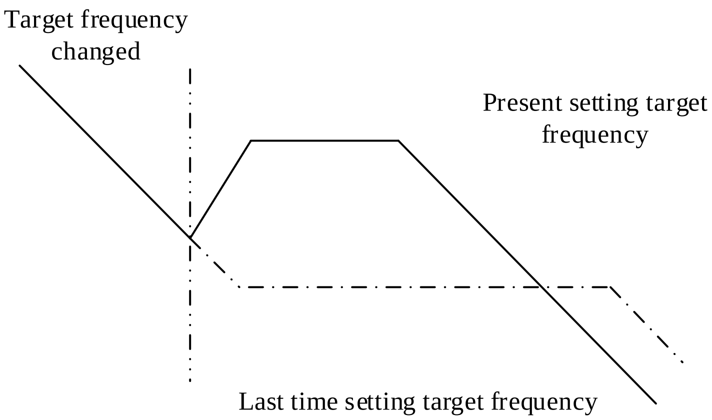
(7) dynamic modify present pulse frequency

- Not the last segment



When the present frequency is changed, it will accelerate/decelerate to target frequency as rising/falling slope.

- The last segment



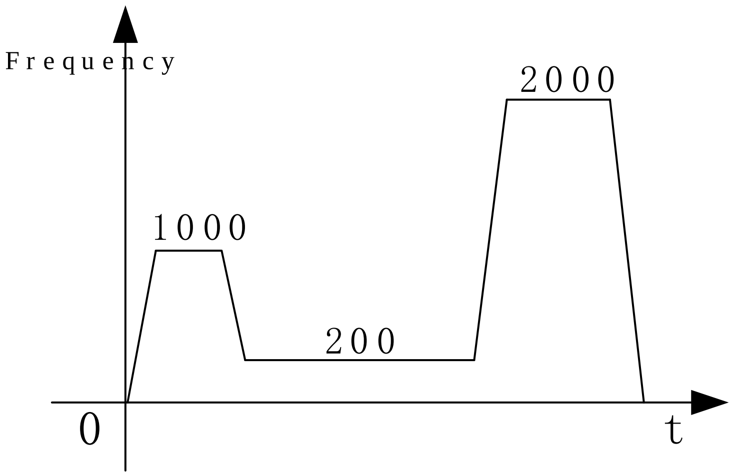
When the present pulse frequency is changed by user, PLC will calcuate the pulse curve again, then output pulse as the new pulse curve.

Example 1

It needs to output 3 continuous segments of pulse, the pulse terminal is Y0, direction terminal is Y2.

Segment	Setting frequency (Hz)	Setting pulse numbers
Segment 1	1000	2000
Segment 2	200	1000
Segment 3	2000	6000
Acceleration/deceleration	The frequency will change 1000Hz every 100ms	

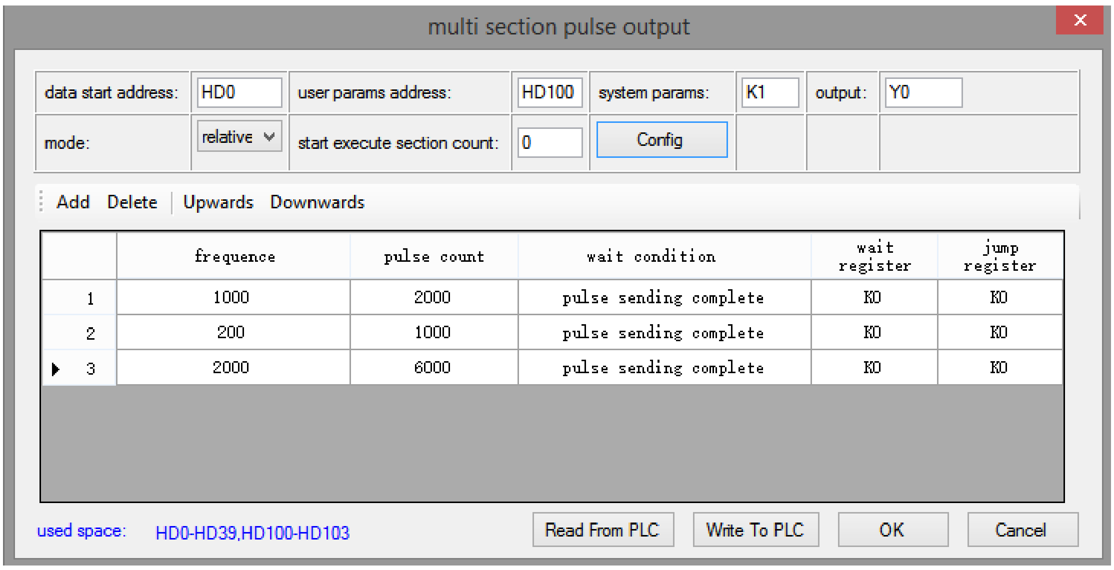
➢ Pulse curve



➢ Pulse instruction

➢ Software configuration

(1) Pulse segment configuration



(2) Pulse configuration parameters

PLC1 - Pulse Set

Config ▾ Delete | init axis | config guide

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	1
Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal
Y0 axis-Common-negative limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	1000
Y0 axis-group 1-Acceleration time of Pulse default s...	100
Y0 axis-group 1-Deceleration time of pulse default s...	100
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

(3) Pulse data address distribution table

Address	Notes	Value
HD0 (double word)	Pulse total segments (1 to 100)	3
HD2 (8 words)	Reserved	0
HD10 (double words)	Pulse frequency (#1)	1000
HD12 (double word)	Pulse number (#1)	2000
HD14	bit15~bit8: waiting condition (#1) H00: pulse sending completion H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse sending completion bit7~bit0: waiting condition register type H00: constant H01: D H02: HD H03: FD H04: X H05: M H06: HM	0
HD15 (double word)	Constant value/ register no. (for waiting condition)(#1)	0
HD17	bit7~bit0: jump register type H00: constant value H01: D H02: HD	0

	H03: FD	
HD+18 (double word)	Constant value/register no. (for jump register)(#1)	0
HD+20 (double word)	Pulse frequency (#2)	200
HD+22 (double word)	Pulse number (#2)	1000
HD+24	Waiting condition, waiting condition register type (#2)	0
HD+25 (double word)	Constant value or register no. (for waiting condition) (#2)	0
HD+27	Jump type, jump register type (#2)	0
HD+28 (double word)	Constant value or register no. (for jump register) (#2)	0
HD+30 (double word)	Pulse frequency (#3)	2000
HD+32 (double word)	Pulse number (#3)	6000
HD+34	Waiting condition, waiting condition register type (#3)	0
HD+35 (double word)	Constant value or register no. (for waiting condition) (#3)	0
HD+37	Jump type, jump register type (for waiting condition) (#3)	0
HD+38 (double word)	Constant value or register no. (for jump register) (#3)	0

(4) System parameters

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	0	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0	0	
SFD902	Pulse number/1 rotation low 16 bits		1	
SFD903	Pulse number/1 rotation high 16 bits		0	
SFD904	Motion quantity/1 rotation low 16 bits		1	
SFD905	Motion quantity/1 rotation high 16 bits		0	
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	2	
SFD907	Direction delay time	Default is 20, unit: ms	20	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	

SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	FFFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0xFF	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		0	
SFD927	CLR signal delay time	Default 20, unit: ms	20	
SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	0	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	0	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	1000	Group 1
SFD951	Pulse default speed high 16 bits		0	
SFD952	Pulse default speed acceleration time		100	

SFD953	Pulse default speed deceleration time		100	
SFD954	Acceleration and deceleration time		0	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved		
SFD956	Max speed limit low 16 bits		3392	
SFD957	Max speed limit high 16 bits		3	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	50	
SFD963	Follow feedforward compensation	0~100, percentage	0	
…				

Note:

※1: As there are many configuration parameters of PLSR, we suggest to use software configuration table to set the parameters.

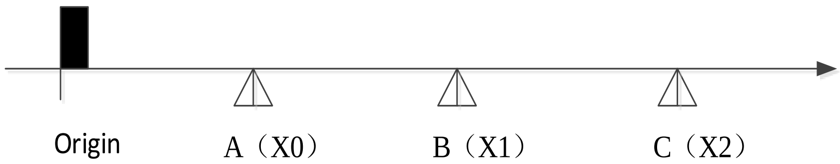
※2: if user needs to set each segment pulse frequency and pulse numbers in the HMI, please configure through the configuration table first, then use instruction DMOV in the program to set the registers (S0+N*10+0, S0+N*10+2).

For example:

```
DMOV  HD200  HD10    //HD200 set segment 1 pulse frequency in HMI
DMOV  HD202  HD12    //HD202 set segment 1 pulse numbers in HMI
DMOV  HD204  HD20    //HD204 set segment 2 pulse frequency in HMI
DMOV  HD206  HD22    //HD206 set segment 2 pulse numbers in HMI
DMOV  HD208  HD30    //HD208 set segment 3 pulse frequency in HMI
DMOV  HD210  HD32    //HD210 set segment 3 pulse numbers in HMI
```

It can also set pulse frequency and numbers in registers HD10, HD12, HD20, HD22, HD30, HD32 directly in the HMI.

Example 2



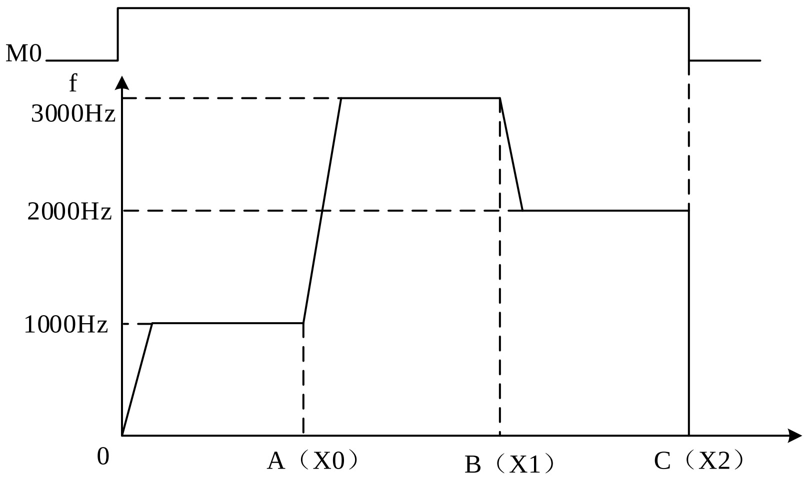
As the above diagram, it needs to move three segments of distance, the position of A, B, C is unknown and the moving speed is different for each segment. We can configure the PLSR to do it. First we install proximity switch at point A, B, C and connect to PLC input X0, X1, X2. The pulse output terminal is Y0, the direction terminal is Y2.

Each segment pulse frequency and numbers:

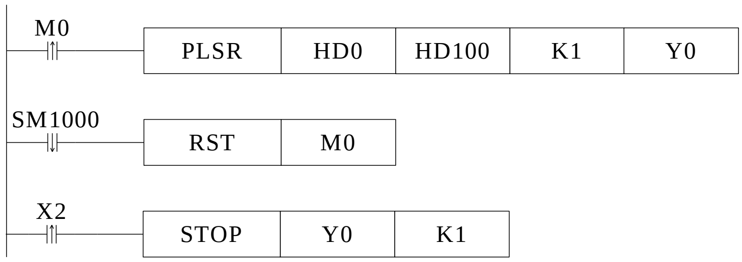
Segment	Frequency setting (Hz)	Pulse number setting
Origin to A	1000	999999999
A to B	3000	999999999
B to C	2000	999999999
Acceleration/deceleration time	The frequency will change 1000Hz every 100ms	

Note:

As the pulse numbers are unknown for each segment, we set a very large pulse numbers to ensure it can reach the proximity switch. When it reaches point C, the pulse will urgent stop by instruction STOP.

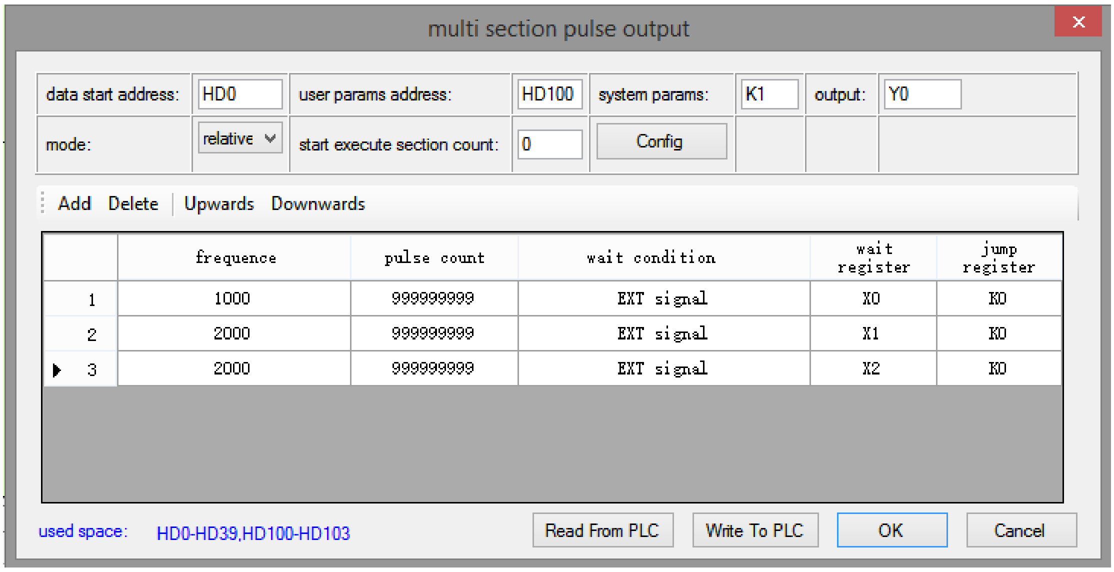


➢ Pulse instructions



- Software configuration

(1) Pulse segment configuration



(2) Pulse configuration parameters

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to th...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coordin...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	1
Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-Far...	normally on
Y0 axis-Common-signal terminal switch state setting-Z p...	normally on
Y0 axis-Common-signal terminal switch state setting-pos...	normally on
Y0 axis-Common-signal terminal switch state setting-neg...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	1000
Y0 axis-group 1-Acceleration time of Pulse default spee...	100
Y0 axis-group 1-Deceleration time of pulse default spee...	100
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

(3) Pulse data address distribution table

Address	Notes	Value
HD0 (double word)	Pulse total segments (1 to 100)	3
HD2 (8 words)	Reserved	0
HD10 (double words)	Pulse frequency (#1)	1000
HD12 (double word)	Pulse number (#1)	999999999
HD14	bit15~bit8: waiting condition (#1) H00: pulse sending completion H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse sending completion	1028

	bit7~bit0: waiting condition register type H00: constant H01: D H02: HD H03: FD H04: X H05: M H06: HM	
HD15 (double word)	Constant value/ register no. (for waiting condition)(#1)	0
HD17	bit7~bit0: jump register type H00: constant value H01: D H02: HD H03: FD	0
HD+18 (double word)	Constant value/register no. (for jump register)(#1)	0
HD+20 (double word)	Pulse frequency (#2)	3000
HD+22 (double word)	Pulse number (#2)	999999999
HD+24	Waiting condition, waiting condition register type (#2)	1028
HD+25 (double word)	Constant value or register no. (for waiting condition) (#2)	1
HD+27	Jump type, jump register type (#2)	0
HD+28 (double word)	Constant value or register no. (for jump register) (#2)	0
HD+30 (double word)	Pulse frequency (#3)	2000
HD+32 (double word)	Pulse number (#3)	999999999
HD+34	Waiting condition, waiting condition register type (#3)	1028
HD+35 (double word)	Constant value or register no. (for waiting condition) (#3)	2
HD+37	Jump type, jump register type (for waiting condition) (#3)	0
HD+38 (double word)	Constant value or register no. (for jump register) (#3)	0

(4) System parameters

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	0	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0	1	
SFD902	Pulse number/1 rotation low 16 bits		0	
SFD903	Pulse number/1 rotation high 16 bits		1	
SFD904	Motion quantity/1 rotation low 16 bits		0	
SFD905	Motion quantity/1 rotation high 16 bits		2	
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	20	
SFD907	Direction delay time	Default is 20, unit: ms	0	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	

SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0xFF	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	FFFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	0xFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		20	
SFD927	CLR signal delay time	Default 20, unit: ms	0	
SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	0	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	1	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	1000	Group 1
SFD951	Pulse default speed high 16 bits		0	
SFD952	Pulse default speed acceleration time		100	

SFD953	Pulse default speed deceleration time		100	
SFD954	Acceleration and deceleration time		0	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved	0	
SFD956	Max speed limit low 16 bits		3392	
SFD957	Max speed limit high 16 bits		3	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	50	
SFD963	Follow feedforward compensation	0~100, percentage	0	
…				

Note:

※1: As there are many configuration parameters of PLSR, we suggest to use software configuration table to set the parameters.

※2: if user needs to set each segment pulse frequency and pulse numbers in the HMI, please configure through the configuration table first, then use instruction DMOV in the program to set the registers (S0+N*10+0, S0+N*10+2).

For example:

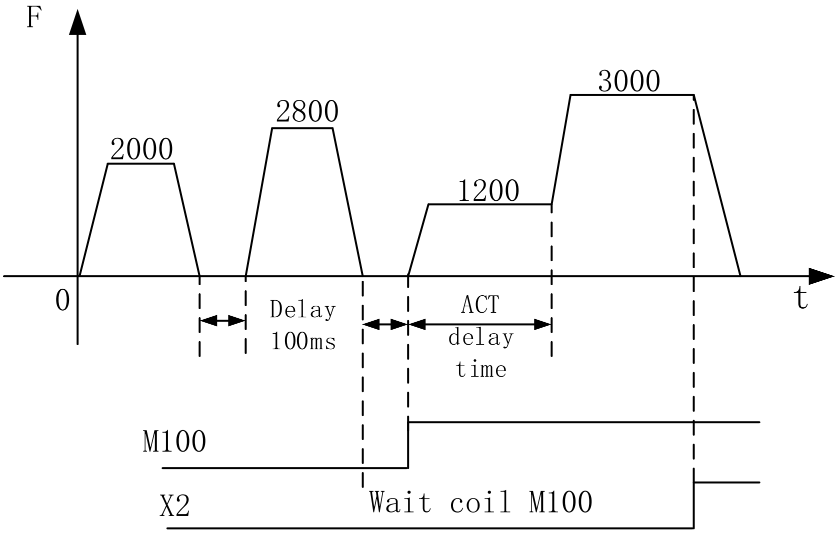
```
DMOV   HD200   HD10        //HD200 set segment 1 pulse frequency in HMI
DMOV   HD202   HD12        //HD202 set segment 1 pulse numbers in HMI
DMOV   HD204   HD20        //HD204 set segment 2 pulse frequency in HMI
DMOV   HD206   HD22        //HD206 set segment 2 pulse numbers in HMI
DMOV   HD208   HD30        //HD208 set segment 3 pulse frequency in HMI
DMOV   HD210   HD32        //HD210 set segment 3 pulse numbers in HMI
```

It can also set pulse frequency and numbers in registers HD10, HD12, HD20, HD22, HD30, HD32 directly in the HMI.

Example 3

It needs to execute 4 segments of pulse: segment 1 pulse frequency is 2000Hz, pulse number is 3000, it will delay 100ms then segment 2 is executed. Segment 2 pulse frequency is 2800Hz, pulse number is 4000. It will wait for M100, when M100 is ON, the segment 3 starts to run. Segment 3 pulse frequency is 1200Hz, pulse number is 999999999. It will delay ACT time 2s after the pulse is outputting then switch to segment 4 at once. Segment 4 pulse frequency is 3000Hz, pulse number is 999999999. When the external signal X2 is ON, it will decelerate and stop the pulse. Pulse acceleration slope is 80ms every 1000Hz, deceleration slope is 120ms every 1000Hz. The pulse direction terminal is Y2.

- Pulse curve:

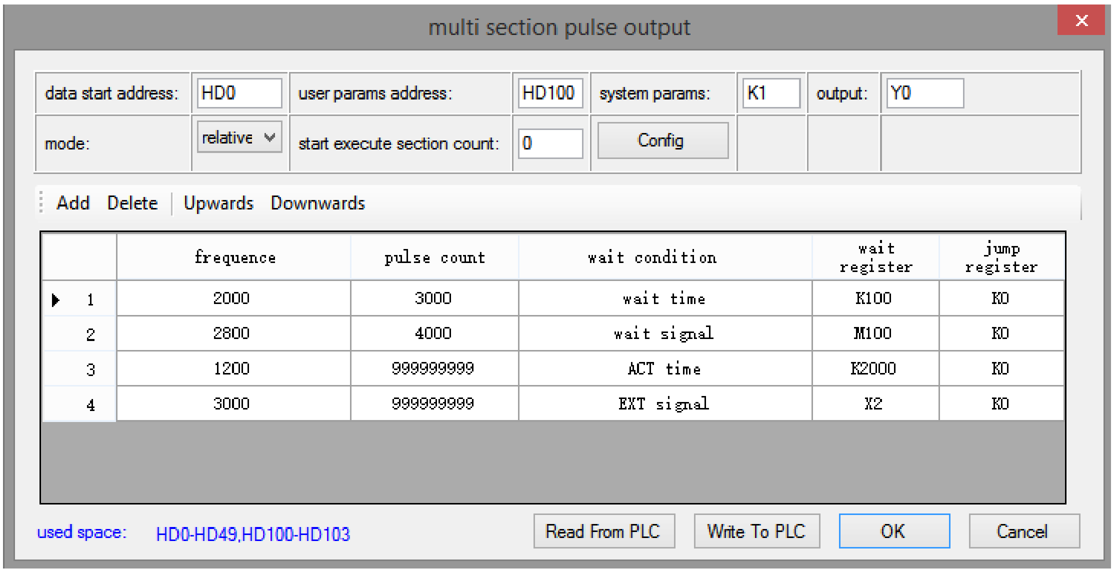


- Pulse instruction

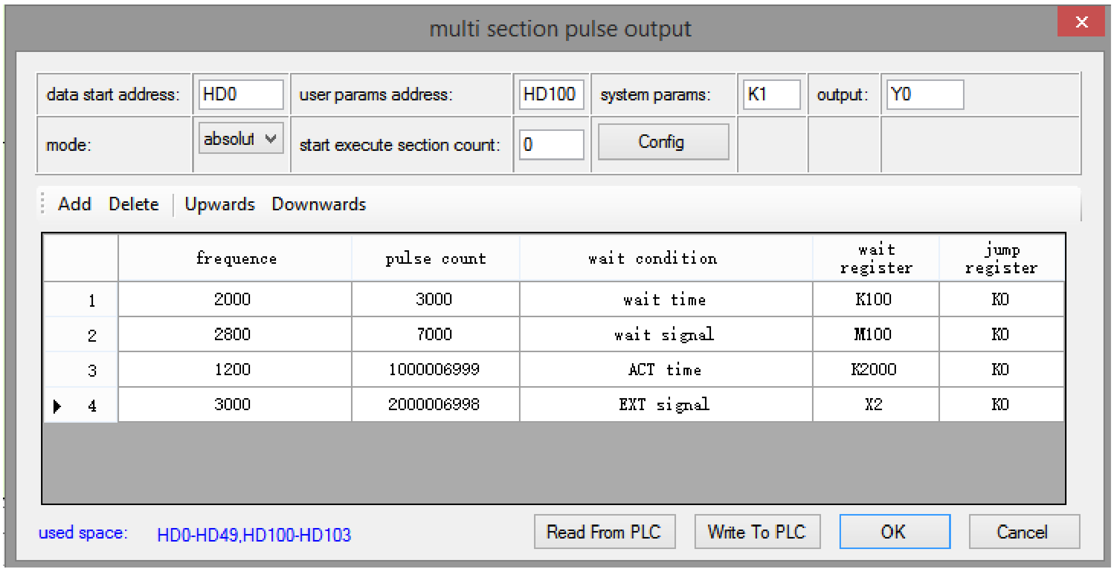
M0					
─┤↑├─	PLSR	HD0	HD100	K1	Y0

- Pulse data configuration

(1) Pulse segment configuration



Pulse data configuration (relative mode)



Pulse data configuration (absolute mode)

(2) System parameters

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	1
Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	1000
Y0 axis-group 1-Acceleration time of Pulse default s...	80
Y0 axis-group 1-Deceleration time of pulse default s...	120
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

(3) Pulse data address distribution table

Address	Notes	Value
HD0 (double word)	Pulse total segments (1 to 100)	4

HD2 (8 words)	Reserved	0
HD10 (double words)	Pulse frequency (#1)	2000
HD12 (double word)	Pulse number (#1)	3000
HD14	bit15~bit8: waiting condition (#1) H00: pulse sending completion H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse sending completion bit7~bit0: waiting condition register type H00: constant H01: D H02: HD H03: FD H04: X H05: M H06: HM	256
HD15 (double word)	Constant value/ register no. (for waiting condition)(#1)	100
HD17	bit7~bit0: jump register type H00: constant value H01: D H02: HD H03: FD	0
HD+18 (double word)	Constant value/register no. (for jump register)(#1)	0
HD+20 (double word)	Pulse frequency (#2)	2800
HD+22 (double word)	Pulse number (#2)	7000
HD+24	Waiting condition, waiting condition register type (#2)	517
HD+25 (double word)	Constant value or register no. (for waiting condition) (#2)	100
HD+27	Jump type, jump register type (#2)	0
HD+28 (double word)	Constant value or register no. (for jump register) (#2)	0
HD+30 (double word)	Pulse frequency (#3)	1200
HD+32 (double word)	Pulse number (#3)	999999999

HD+34	Waiting condition, waiting condition register type (#3)	768
HD+35 (double word)	Constant value or register no. (for waiting condition) (#3)	2000
HD+37	Jump type, jump register type (for waiting condition) (#3)	0
HD+38 (double word)	Constant value or register no. (for jump register) (#3)	0
HD+40 (double word)	Pulse frequency (#4)	3000
HD+42 (double word)	Pulse number (#4)	999999999
HD+44	Waiting condition, waiting condition register type (#4)	1028
HD+45 (double word)	Constant value or register no. (for waiting condition) (#4)	2
HD+47	Jump type, jump register type (for waiting condition) (#4)	0
HD+48 (double word)	Constant value or register no. (for jump register) (#4)	0

(4) System parameters

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	0	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0	0	
SFD902	Pulse number/1 rotation low 16 bits		1	

SFD903	Pulse number/1 rotation high 16 bits		0	
SFD904	Motion quantity/1 rotation low 16 bits		1	
SFD905	Motion quantity/1 rotation high 16 bits		0	
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	2	
SFD907	Direction delay time	Default is 20, unit: ms	20	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	
SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	FFFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0xFF	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		0	
SFD927	CLR signal delay time	Default 20, unit: ms	20	

SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	0	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	0	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	1000	Group 1
SFD951	Pulse default speed high 16 bits		0	
SFD952	Pulse default speed acceleration time		100	
SFD953	Pulse default speed deceleration time		100	
SFD954	Acceleration and deceleration time		0	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved	0	
SFD956	Max speed limit low 16 bits		3392	
SFD957	Max speed limit high 16 bits		3	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	50	
SFD963	Follow feedforward compensation	0~100, percentage	0	
…				

Note:

※1: As there are many configuration parameters of PLSR, we suggest to use software configuration table to set the parameters.

※2: if user needs to set each segment pulse frequency and pulse numbers in the HMI, please configure through the configuration table first, then use instruction DMOV in the program to set the registers (S0+N*10+0, S0+N*10+2).

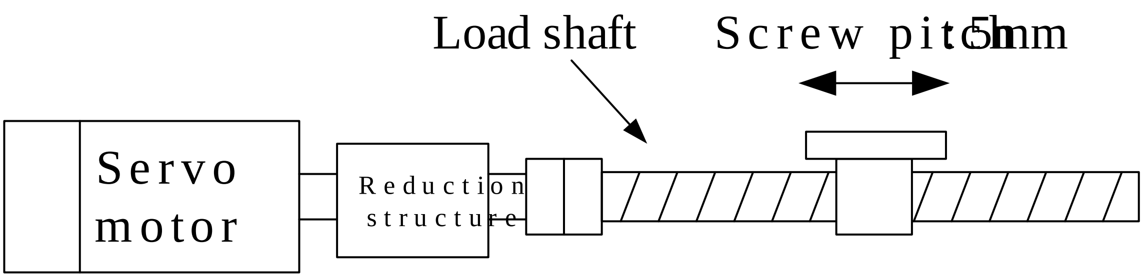
For example:

```
DMOV   HD200   HD10      //HD200 set segment 1 pulse frequency in HMI
DMOV   HD202   HD12      //HD202 set segment 1 pulse numbers in HMI
DMOV   HD204   HD20      //HD204 set segment 2 pulse frequency in HMI
DMOV   HD206   HD22      //HD206 set segment 2 pulse numbers in HMI
DMOV   HD208   HD30      //HD208 set segment 3 pulse frequency in HMI
DMOV   HD210   HD32      //HD210 set segment 3 pulse numbers in HMI
DMOV   HD212   HD40      //HD212 set segment 4 pulse frequency in HMI
DMOV   HD214   HD42      //HD214 set segment 4 pulse numbers in HMI
```

It can also set pulse frequency and numbers in registers HD10, HD12, HD20, HD22, HD30, HD32, HD40, HD42 directly in the HMI.

Example 4

There is a transmission mechanism which includes one servo drive (electronic gear ratio is 1:1), one servo motor (encoder is 2500ppr), it connects the ball screw through a reducer (the reduction ratio is 1:2), the ball screw pitch is 10mm, the ball screw drives a working table which can move left and right. Now it needs to move the table from left to right for 200mm, then move in reverse direction for 200mm, the speed is 20mm/s, acceleration time is 100ms, deceleration time is 200ms, the pulse direction terminal is Y2.



Mechanical structure

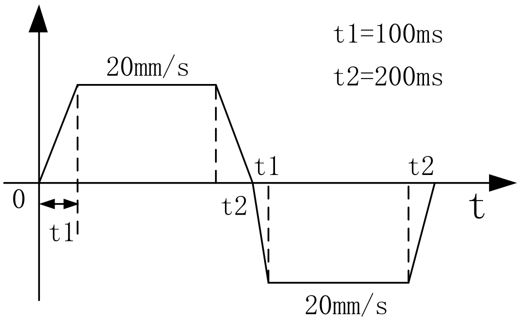
$$\text{Pulse number per rotate} = 20000 = 2500 * 4 * \frac{2}{1}$$

$$\text{Motion quantity per rotate} = \text{pitch} = 10\text{mm}$$

$$20\text{mm/s} = \frac{20\text{mm}}{10\text{mm}} * 20000 = 40000 \text{ pulse/s}$$

The max pulse output frequency is 40K/s, less than 200K/s, the PLC can run well.

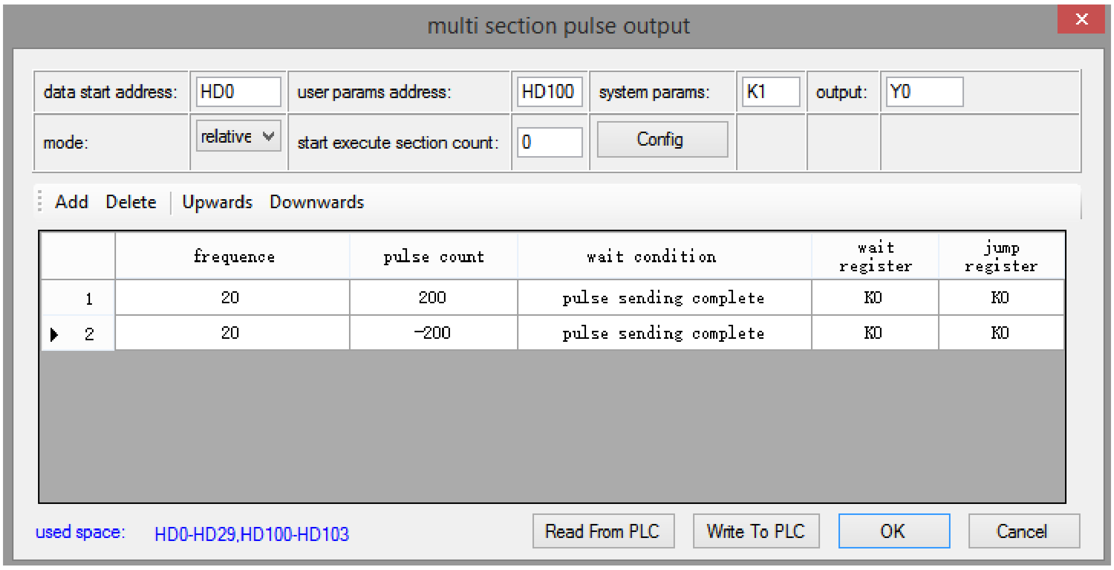
➢ Pulse curve



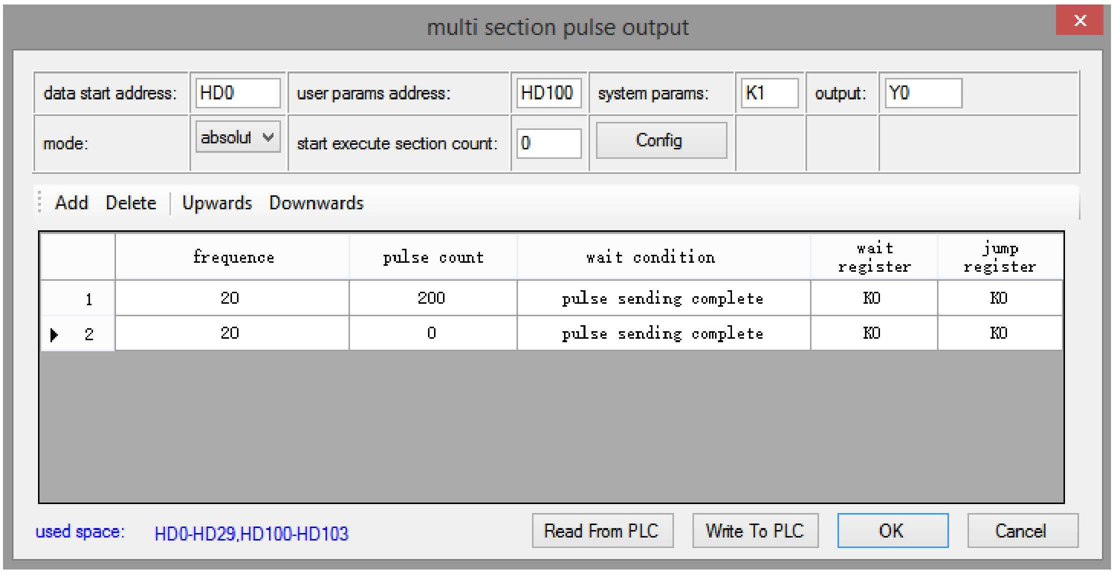
➢ Pulse instruction

➢ Pulse configuration

(1) Pulse segment configuration



Relative mode



Absolute mode

(2) System parameters (relative mode)

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	1mm
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	20000
Y0 axis-Common-1mm(revolve)	10
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	20
Y0 axis-group 1-Acceleration time of Pulse default s...	100
Y0 axis-group 1-Deceleration time of pulse default s...	200
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	100
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

(3) Pulse data address distribution table

Address	Notes	Value
HD0 (double word)	Pulse total segments (1 to 100)	2
HD2 (8 words)	Reserved	0
HD10 (double words)	Pulse frequency (#1)	20
HD12 (double word)	Pulse number (#1)	200
HD14	bit15~bit8: waiting condition (#1) H00: pulse sending completion H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse sending completion bit7~bit0: waiting condition register type	0

	H00: constant H01: D H02: HD H03: FD H04: X H05: M H06: HM	
HD15 (double word)	Constant value/ register no. (for waiting condition)(#1)	0
HD17	bit7~bit0: jump register type H00: constant value H01: D H02: HD H03: FD	0
HD+18 (double word)	Constant value/register no. (for jump register)(#1)	0
HD+20 (double word)	Pulse frequency (#2)	20
HD+22 (double word)	Pulse number (#2)	-200
HD+24	Waiting condition, waiting condition register type (#2)	0
HD+25 (double word)	Constant value or register no. (for waiting condition) (#2)	0
HD+27	Jump type, jump register type (#2)	0
HD+28 (double word)	Constant value or register no. (for jump register) (#2)	0

(4) System parameters

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	1792	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0	0	
SFD902	Pulse number/1 rotation low 16 bits		20000	
SFD903	Pulse number/1 rotation high 16 bits			
SFD904	Motion quantity/1 rotation low 16 bits		10	
SFD905	Motion quantity/1 rotation high 16 bits			
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	2	
SFD907	Direction delay time	Default is 20, unit: ms	20	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	

SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	FFFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0xFF	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		0	
SFD927	CLR signal delay time	Default 20, unit: ms	20	
SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	0	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	0	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	20	Group 1
SFD951	Pulse default speed high 16 bits		0	
SFD952	Pulse default speed acceleration time		100	

SFD953	Pulse default speed deceleration time		200	
SFD954	Acceleration and deceleration time		0	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved	0	
SFD956	Max speed limit low 16 bits		100	
SFD957	Max speed limit high 16 bits		0	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	50	
SFD963	Follow feedforward compensation	0~100, percentage	0	
…				

Note:

※1: As there are many configuration parameters of PLSR, we suggest to use software configuration table to set the parameters.

※2: if user needs to set each segment pulse frequency and pulse numbers in the HMI, please configure through the configuration table first, then use instruction DMOV in the program to set the registers (S0+N*10+0, S0+N*10+2).

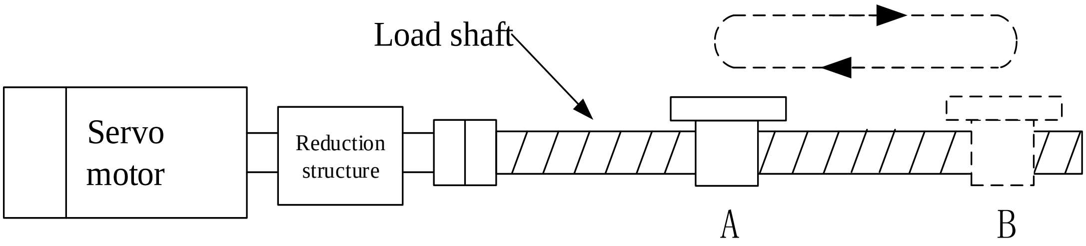
For example:

```
DMOV   HD200   HD10        //HD200 set segment 1 pulse frequency in HMI
DMOV   HD202   HD12        //HD202 set segment 1 pulse numbers in HMI
DMOV   HD204   HD20        //HD204 set segment 2 pulse frequency in HMI
DMOV   HD206   HD22        //HD206 set segment 2 pulse numbers in HMI
```

It can also set pulse frequency and numbers in registers HD10, HD12, HD20, HD22 directly in the HMI.

Example 5

There is a transmission mechanism which includes one servo drive (electronic gear ratio is 1:1), one servo motor (encoder is 2500ppr), it connects the ball screw through a reducer (the reduction ratio is 1:2), the ball screw pitch is 5mm, the ball screw drives a working table which can move left and right. Now it needs to move forth and back on the table, A to B distance is 200mm, A to B speed is 20mm/s, B to A speed is 30mm/s, acceleration time is 100ms, deceleration time is 200ms, the pulse direction terminal is Y2, the mechanical clearance of A to B to A is 3mm, B to A to B is 2mm.



Mechanical structure

We can calculate the following things:

$$\text{Pulse number per rotate} = 20000 = 2500 * 4 * \frac{2}{1}$$

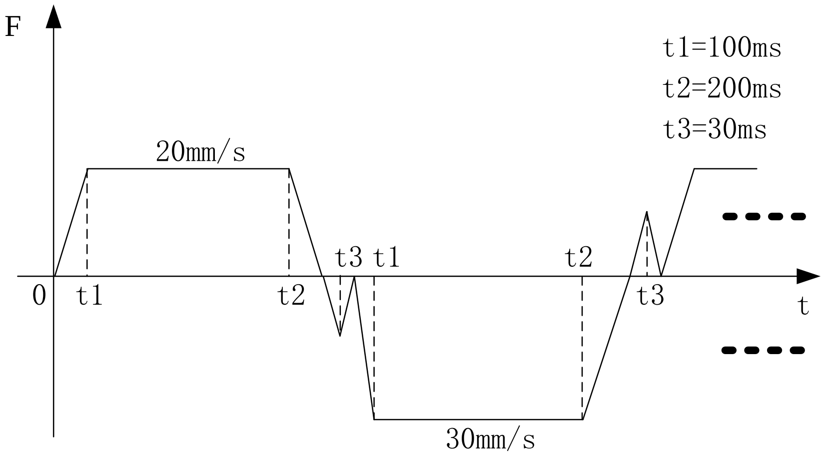
$$\text{Moving quantity} = \text{pitch} = 5\text{mm}$$

$$20\text{mm/s} = \frac{20\text{mm}}{5\text{mm}} * 20000 = 80000\text{pulse/s}$$

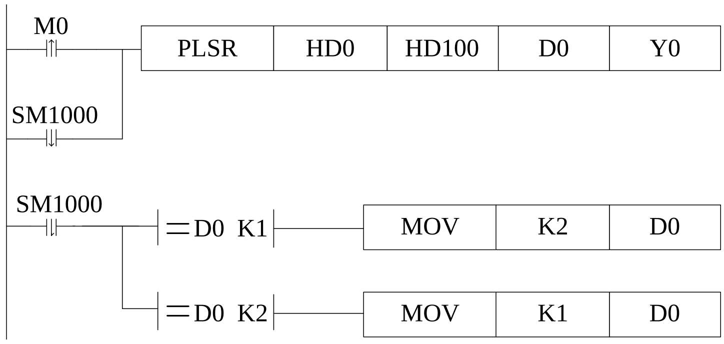
$$30\text{mm/s} = \frac{30\text{mm}}{5\text{mm}} * 20000 = 120000\text{pulse/s}$$

As the acceleration and deceleration time for forward motion and reverse motion is same, but the max frequency is different, so their acceleration and deceleration slope is different.
Forward acceleration slope: 80000Hz/100ms, forward deceleration slope: 80000Hz/200ms.
Reverse acceleration slope: 120000Hz/100ms, reverse deceleration slope: 120000Hz/200ms.
We needs to set two groups of parameter as there are two groups of acc/dec slope.
The max frequency is 40K/s and 120K/s, less than 200K/s, so PLC can work normally.

- Pulse curve

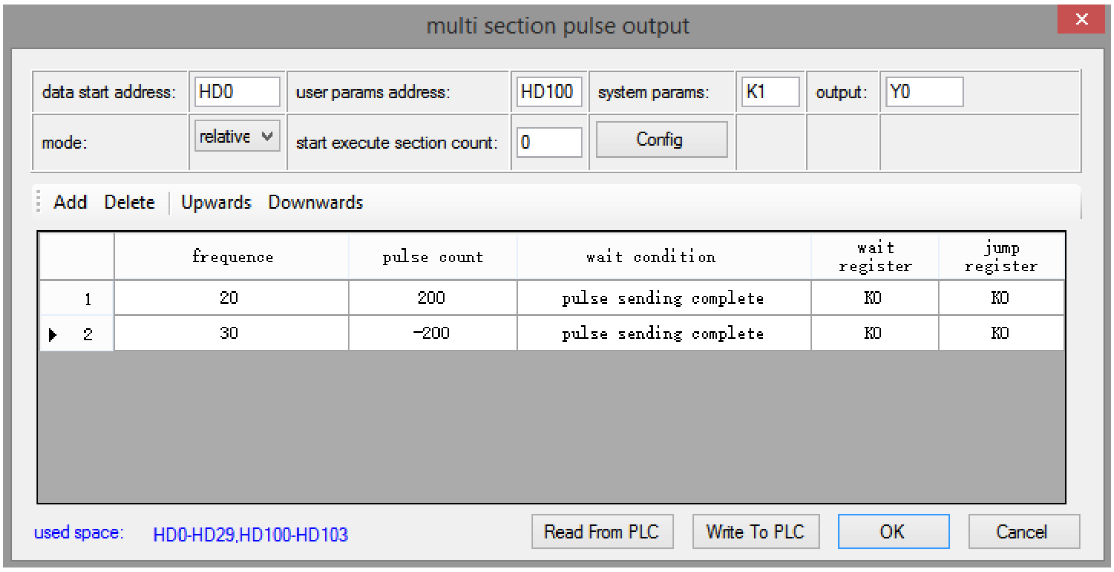


➢ Pulse instruction

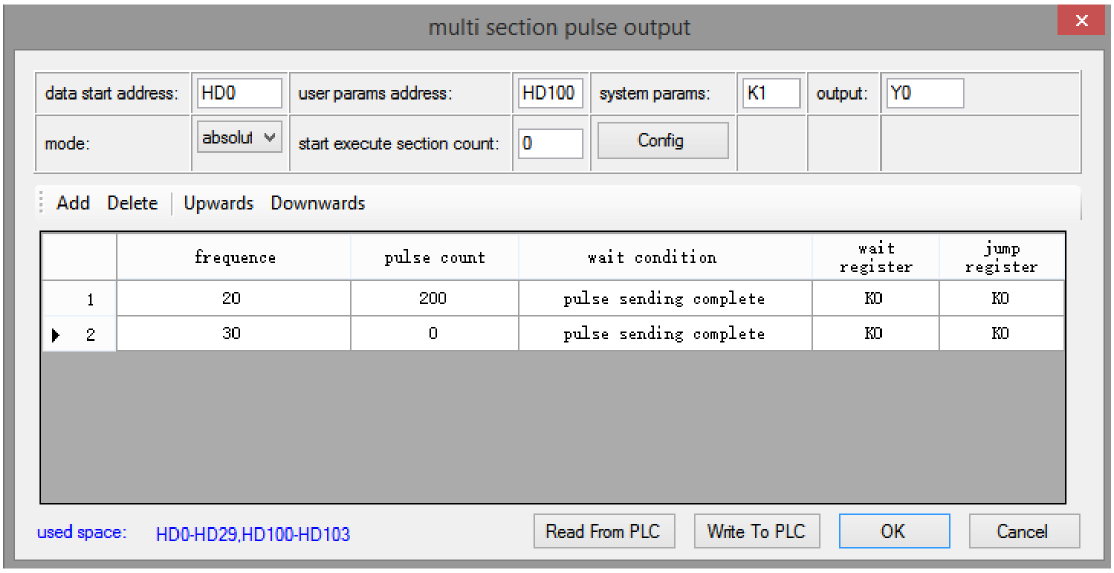


➢ Pulse data configuration

(1) Pulse segment configuration



Relative mode



Absolute mode

(2) System parameters

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	1mm
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	20000
Y0 axis-Common-1mm(revolve)	5
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	3
Y0 axis-Common-Gear clearance negative compensation	2
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	20
Y0 axis-group 1-Acceleration time of Pulse default s...	100
Y0 axis-group 1-Deceleration time of pulse default s...	200
Y0 axis-group 1-Acceleration and deceleration time (ms)	30
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	50
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

Param	Value
Y0 axis-group 2-Pulse default speed	30
Y0 axis-group 2-Acceleration time of Pulse default s...	100
Y0 axis-group 2-Deceleration time of pulse default s...	200
Y0 axis-group 2-Acceleration and deceleration time (ms)	30
Y0 axis-group 2-pulse acc/dec mode	linear acc/dec
Y0 axis-group 2-Max speed	50
Y0 axis-group 2-Initial speed	0
Y0 axis-group 2-stop speed	0
Y0 axis-group 2-FOLLOW performance param(1-100)	50
Y0 axis-group 2-FOLLOW forward compensation(0-100)	0

(3) Pulse data address distribution table(relative mode)

Address	Notes	Value
HD0 (double word)	Pulse total segments (1 to 100)	2
HD2 (8 words)	Reserved	0
HD10 (double words)	Pulse frequency (#1)	20
HD12 (double word)	Pulse number (#1)	200
HD14	bit15~bit8: waiting condition (#1) H00: pulse sending completion H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse sending completion bit7~bit0: waiting condition register type H00: constant H01: D H02: HD H03: FD H04: X H05: M H06: HM	0
HD15 (double word)	Constant value/ register no. (for waiting condition)(#1)	0
HD17	bit7~bit0: jump register type H00: constant value H01: D H02: HD H03: FD	0
HD+18 (double word)	Constant value/register no. (for jump register)(#1)	0
HD+20 (double word)	Pulse frequency (#2)	20
HD+22 (double word)	Pulse number (#2)	-200
HD+24	Waiting condition, waiting condition register type (#2)	0
HD+25 (double word)	Constant value or register no. (for waiting condition) (#2)	0
HD+27	Jump type, jump register type (#2)	0
HD+28 (double word)	Constant value or register no. (for jump register) (#2)	0

(4) System parameters

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	1792	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0	0	
SFD902	Pulse number/1 rotation low 16 bits		20000	
SFD903	Pulse number/1 rotation high 16 bits		0	
SFD904	Motion quantity/1 rotation low 16 bits		5	
SFD905	Motion quantity/1 rotation high 16 bits		0	
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	2	
SFD907	Direction delay time	Default is 20, unit: ms	20	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	

SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	FFFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0xFF	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		0	
SFD927	CLR signal delay time	Default 20, unit: ms	20	
SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	0	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	0	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	20	Group 1
SFD951	Pulse default speed high 16 bits		0	
SFD952	Pulse default speed acceleration time		100	

SFD953	Pulse default speed deceleration time		200	
SFD954	Acceleration and deceleration time		30	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved	0	
SFD956	Max speed limit low 16 bits		50	
SFD957	Max speed limit high 16 bits		0	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	50	
SFD963	Follow feedforward compensation	0~100, percentage	0	
…				
SFD970	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	30	Group 2
SFD971	Pulse default speed high 16 bits		0	
SFD972	Pulse default speed acceleration time		100	
SFD973	Pulse default speed deceleration time		200	
SFD974	Acceleration and deceleration time		30	
SFD975	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved	0	
SFD976	Max speed limit low 16 bits		50	
SFD977	Max speed limit high 16 bits		0	
SFD978	Initial speed low 16 bits		0	
SFD979	Initial speed high 16 bits		0	
SFD980	Stop speed low 16 bits		0	

SFD981	Stop speed high 16 bits		0	
SFD982	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	50	
SFD983	Follow feedforward compensation	0~100, percentage	0	
…				

Note:

※1: As there are many configuration parameters of PLSR, we suggest to use software configuration table to set the parameters.

※2: if user needs to set each segment pulse frequency and pulse numbers in the HMI, please configure through the configuration table first, then use instruction DMOV in the program to set the registers (S0+N*10+0, S0+N*10+2).

For example:

```
DMOV   HD200   HD10        //HD200 set segment 1 pulse frequency in HMI
DMOV   HD202   HD12        //HD202 set segment 1 pulse numbers in HMI
DMOV   HD204   HD20        //HD204 set segment 2 pulse frequency in HMI
DMOV   HD206   HD22        //HD206 set segment 2 pulse numbers in HMI
```

It can also set pulse frequency and numbers in registers HD10, HD12, HD20, HD22 directly in the HMI.

1-2-3. Variable frequency pulse output [PLSF]

■ Instruction summarization

Variable frequency pulse output instruction.

Variable frequency pulse output [PLSF]			
16-bit	-	32-bit instruction	PLSF
Execution condition	Normally open/close coil	Suitable mode	XD, XL (except XD1, XL1)
Hardware	-	Software	-

■ Operand

Operand	Function	Type
S0	Pulse frequency	32-bit, double word
S1	System parameters (1 to 4)	32-bit, double word
D	Pulse output terminal	Bit

■ Suitable soft component

word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	•	•	•	•	•	•	•	•	•		
S1	•	•							•		

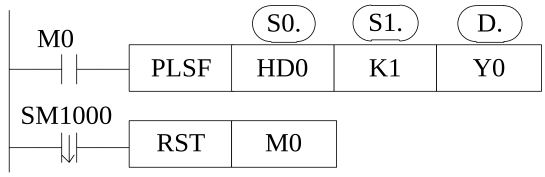
bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		•					

*Note: D means D, HD. TD means TD, HTD. CD means CD, HCD, HSCD, HSD. DM means DM, DHM.
DS means DS, DHS. M means M, HM, SM. S means S, HS. T means T, HT. C means C, HC.

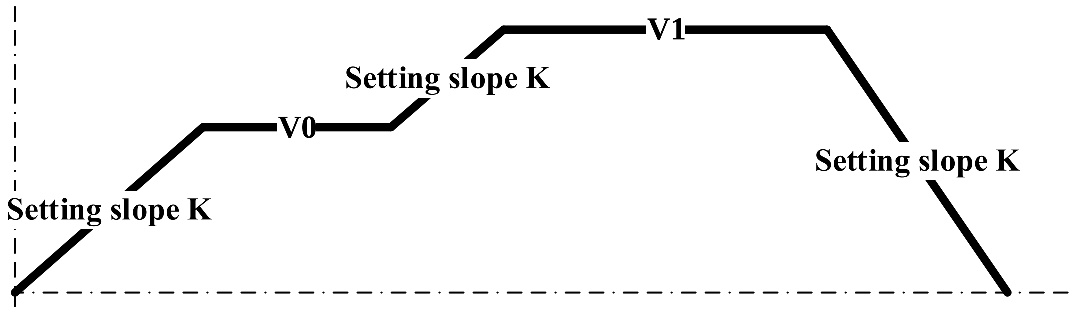
Function and action

Instruction mode:



- Frequency range: 1Hz ~100KHz or -100KHz ~ -1Hz (note: PLC can output 100~200KHz pulse, but we cannot ensure all the servo drive can work fine, please connect 500Ω resistor between output terminal and 24V power supply)
- When the frequency is positive, it outputs pulse in forward direction, when the frequency is negative, it outputs pulse in reverse direction
- Pulse direction terminal is set in system parameters
- The pulse frequency outputting from Y terminal will change as the S0 value
- HSD0 (double word) is cumulative pulse numbers, HSD2 (double word) is cumulative equivalents
- The frequency jump (acceleration/deceleration) will dynamic adjust as pulse rising or falling slope (refer to chapter 1-2-1-3)
- The system parameters are same to PLSR, refer to chapter 1-2-1-3

Output mode

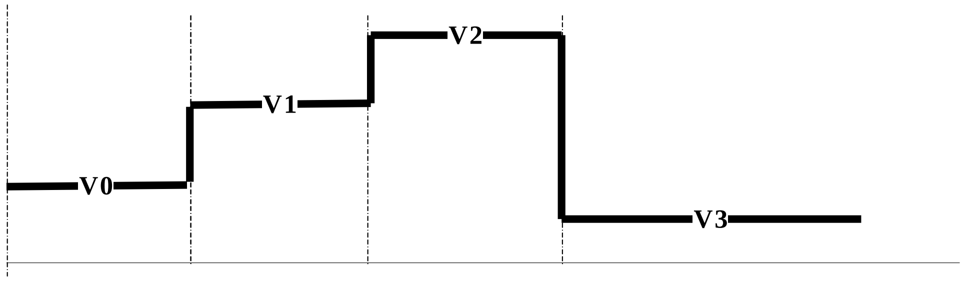


- The pulse output terminal is set in system parameters (refer to chapter 6-2-1-3)
- When the frequency is positive, it outputs pulse in forward direction, when the frequency is negative, it outputs pulse in reverse direction
- When S0 is 0, PLSF stop pulse outputting.
- It will dynamic adjust pulse curve according to pulse slope and setting frequency. If the setting frequency is 0, pulse will stop outputting. And it will output pulses when setting frequency is non-zero value.

Switching mode analysis

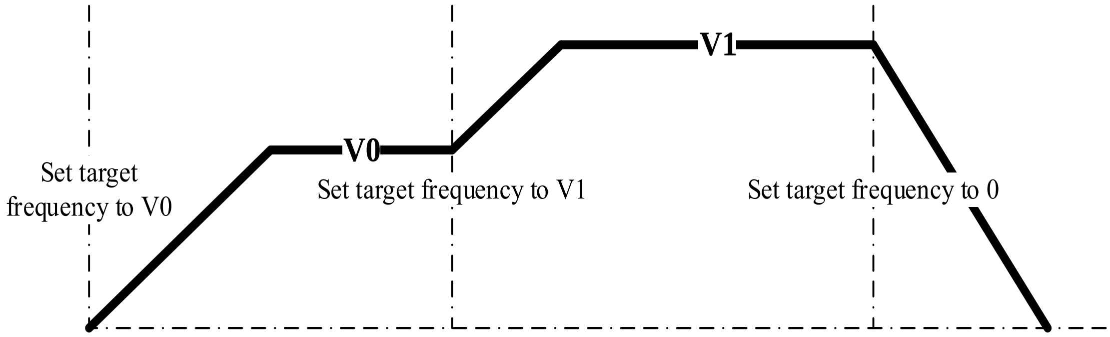
(A) Pulse default speed acceleration deceleration time is 0

The pulse frequency will change as setting frequency.

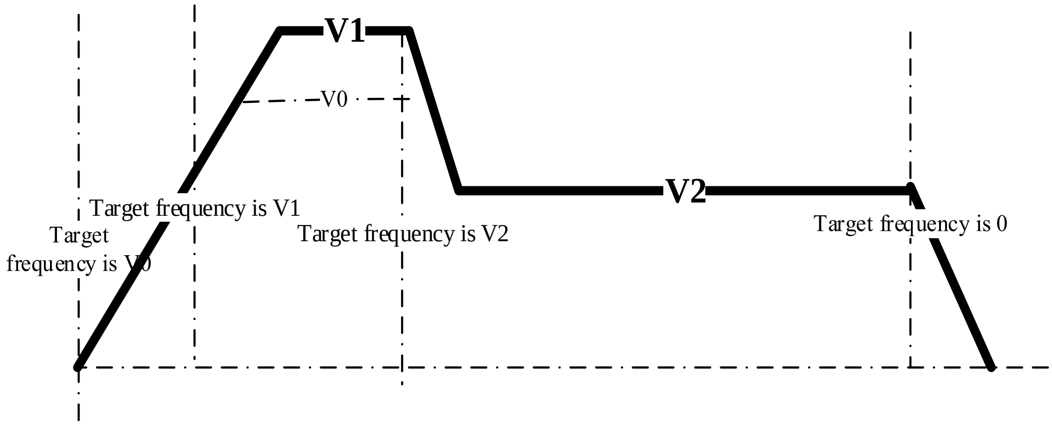


(B) Pulse default speed acceleration deceleration time is not 0

(1) the pulse is in stable segment when user setting new frequency, it will switch to setting frequency through the slope.

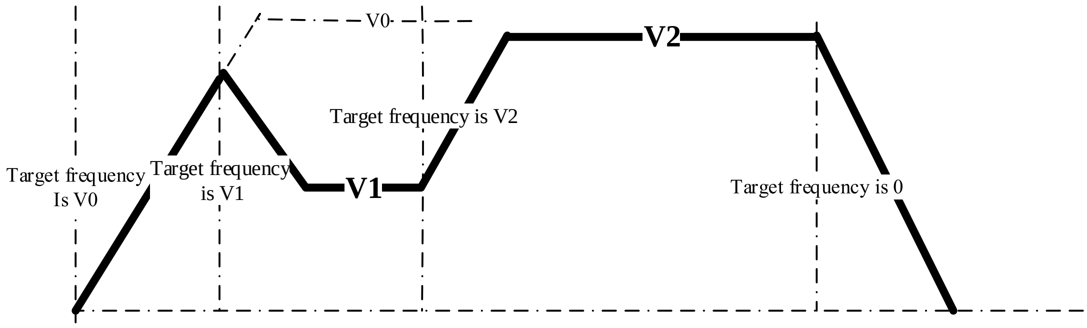


(2) the pulse is not in stable segment when user setting new frequency, it will switch to setting frequency through the slope. (present setting frequency > last time setting frequency, takes present setting frequency as target).



User set target frequency V1 (V1>V0) before reaching setting frequency V0, at this time, it will go to new setting frequency V1 as the slope.

(3) the pulse is not in stable segment when user setting new frequency, it will switch to setting frequency through the slope. (present setting frequency < last time setting frequency, and present setting frequency < present frequency). setting frequency as target).

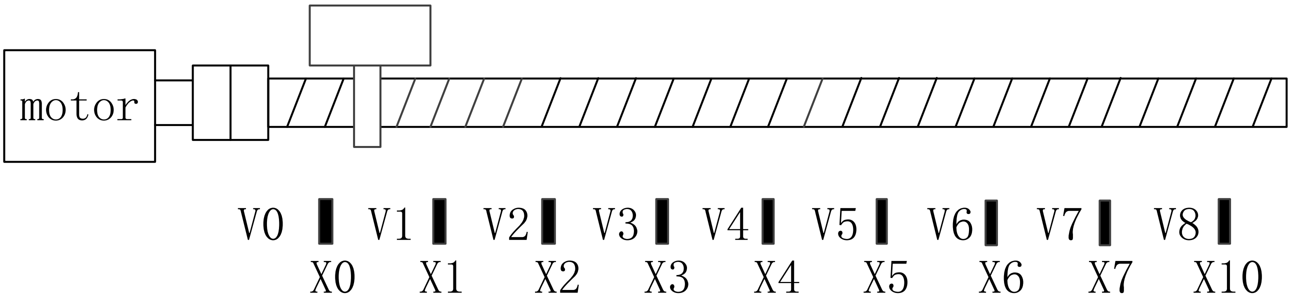


User set target frequency V1 (V1<V0, V1<present frequency) before reaching setting frequency V0, at this time, it will go to new setting frequency V1 as the down slope.

Example 1

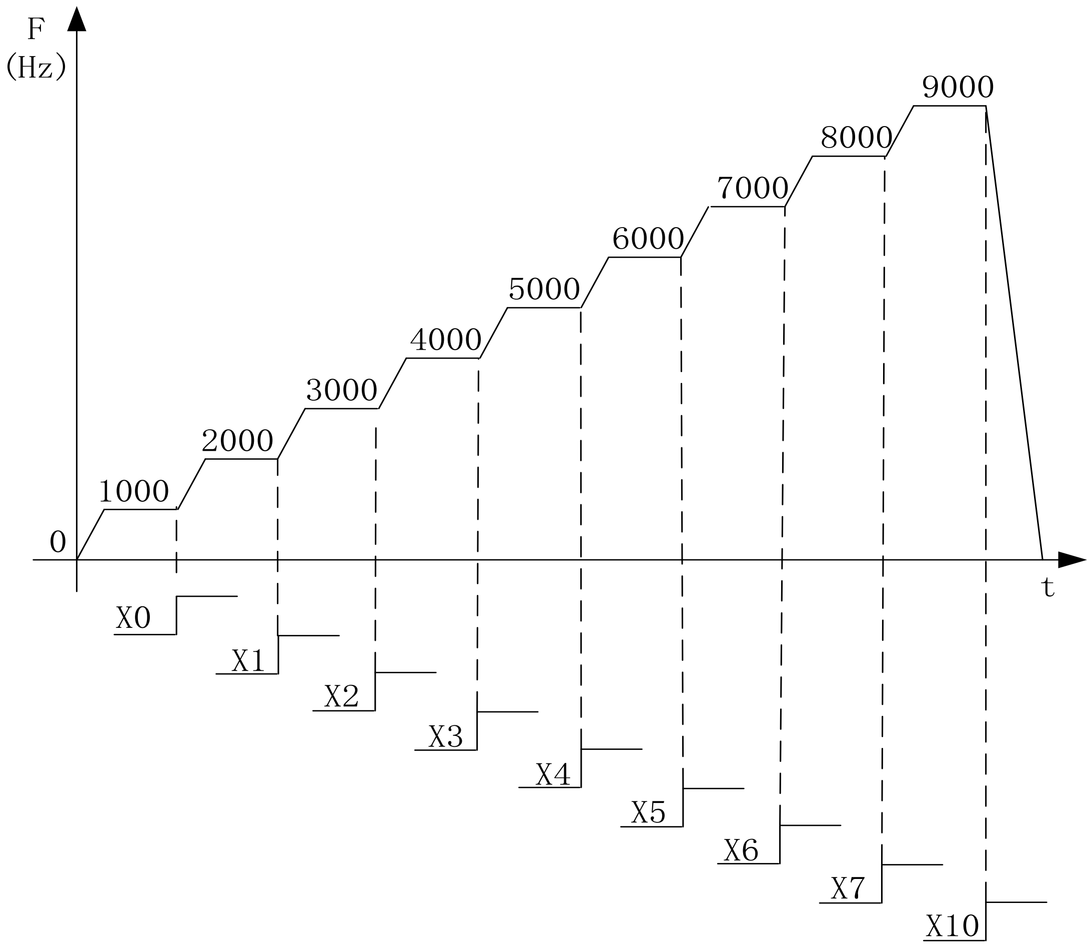
As below diagram, the working table needs to move from left to right position X10. Now the position X0 to X10 all installed proximity switch. The speed from left to X0 is V0, X0 to X1 speed is V1, X1 to X2 speed is V2, X2 to X3 speed is V3, X3 to X4 speed is V4, X4 to X5 speed is V5, X5 to X6 speed is V6, X6 to X7 speed is V7, X7 to X10 speed is V8. Acceleration/deceleration slope is 1000Hz/100ms. Pulse direction terminal is Y2.

No.	Speed name	Speed	No.	Speed name	speed
1	V0	1000Hz	6	V5	6000Hz
2	V1	2000Hz	7	V6	7000Hz
3	V2	3000Hz	8	V7	8000Hz
4	V3	4000Hz	9	V8	9000Hz
5	V4	5000Hz			

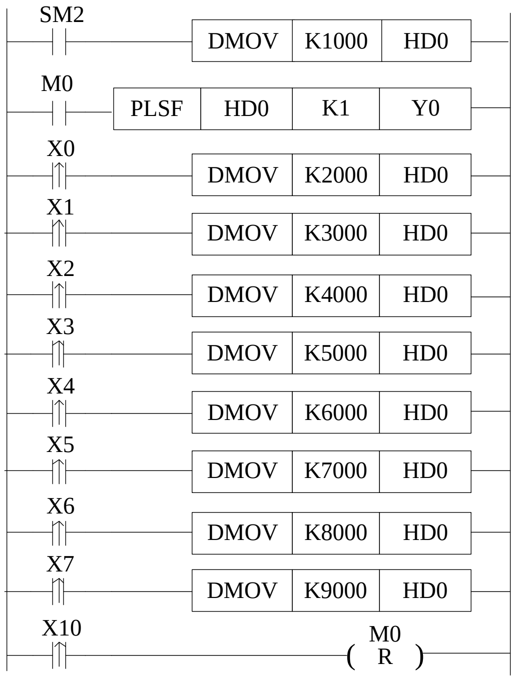


Mechanical structure

- Pulse curve

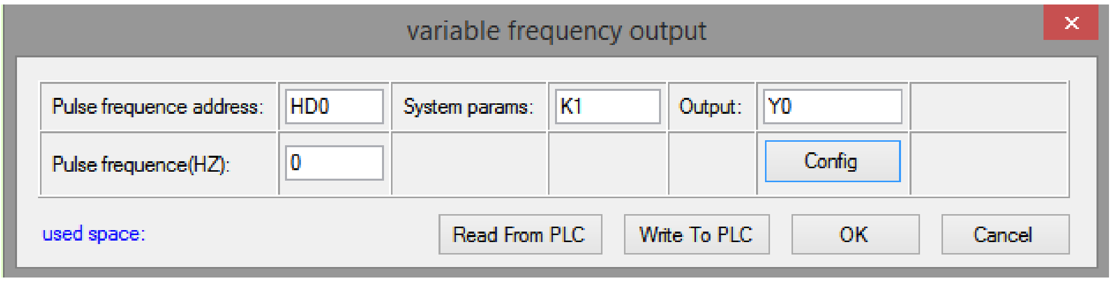


- Pulse instruction



- Software configuration

(1) Pulse segment configuration



(2) System parameter configuration (relative mode)

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	1
Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	1000
Y0 axis-group 1-Acceleration time of Pulse default s...	100
Y0 axis-group 1-Deceleration time of pulse default s...	100
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

(3) System parameters address:

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	0	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0		
SFD902	Pulse number/1 rotation low 16 bits		0	
SFD903	Pulse number/1 rotation high 16 bits		0	
SFD904	Motion quantity/1 rotation low 16 bits		0	
SFD905	Motion quantity/1 rotation high 16 bits		0	
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	2	
SFD907	Direction delay time	Default is 20, unit: ms	20	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	

SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	FFFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0xFF	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		0	
SFD927	CLR signal delay time	Default 20, unit: ms	20	
SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	2	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	0	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	1000	Group 1
SFD951	Pulse default speed high 16 bits		0	
SFD952	Pulse default speed acceleration time		100	

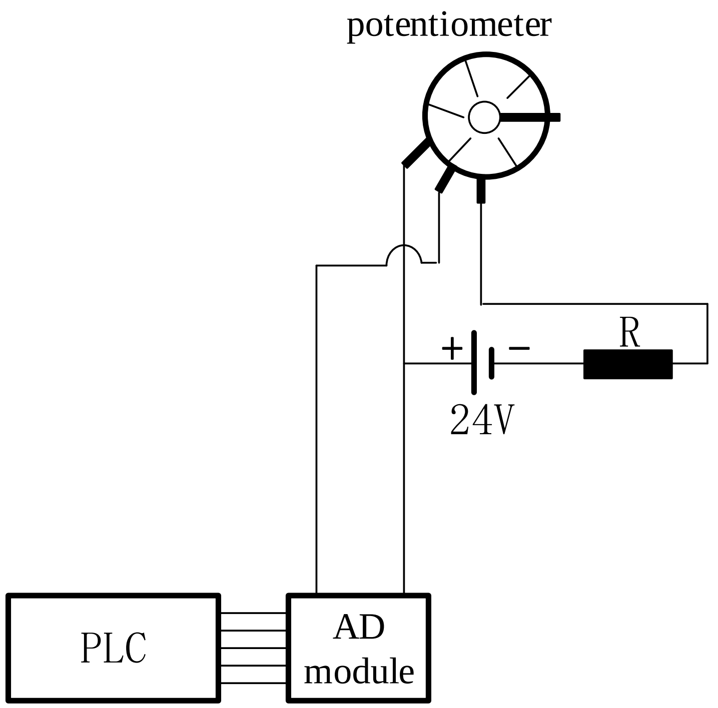
SFD953	Pulse default speed deceleration time		100	
SFD954	Acceleration and deceleration time		0	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved	0	
SFD956	Max speed limit low 16 bits		3392	
SFD957	Max speed limit high 16 bits		3	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.	0	
SFD963	Follow feedforward compensation	0~100, percentage	0	
…				

Note:

※1: As there are many configuration parameters of PLSF, we suggest to use software configuration table to set the parameters.

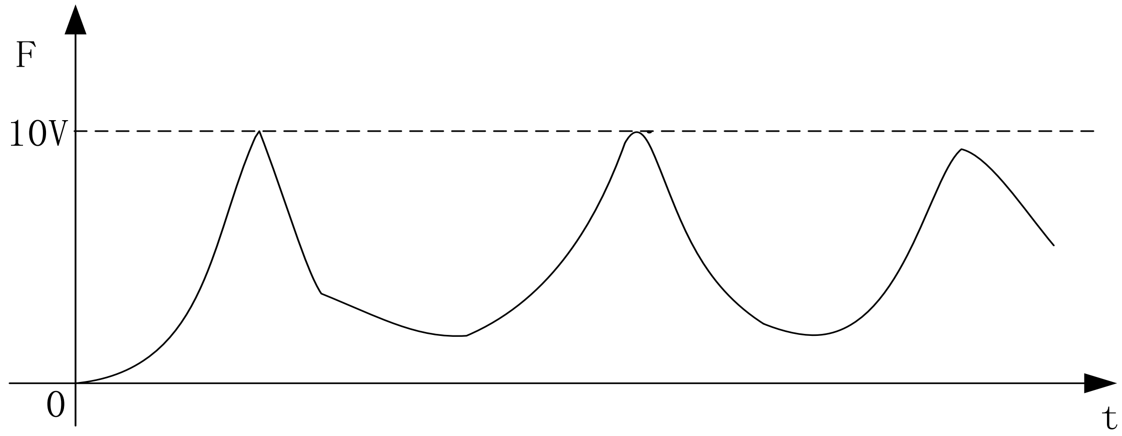
Example 2

As below diagram, the AD module collects 0-10V voltage signal and transforms to digital value 0-16383, this value will be sent to PLSF pulse frequency register, and PLC will output the pulse curve changing as the voltage signal.



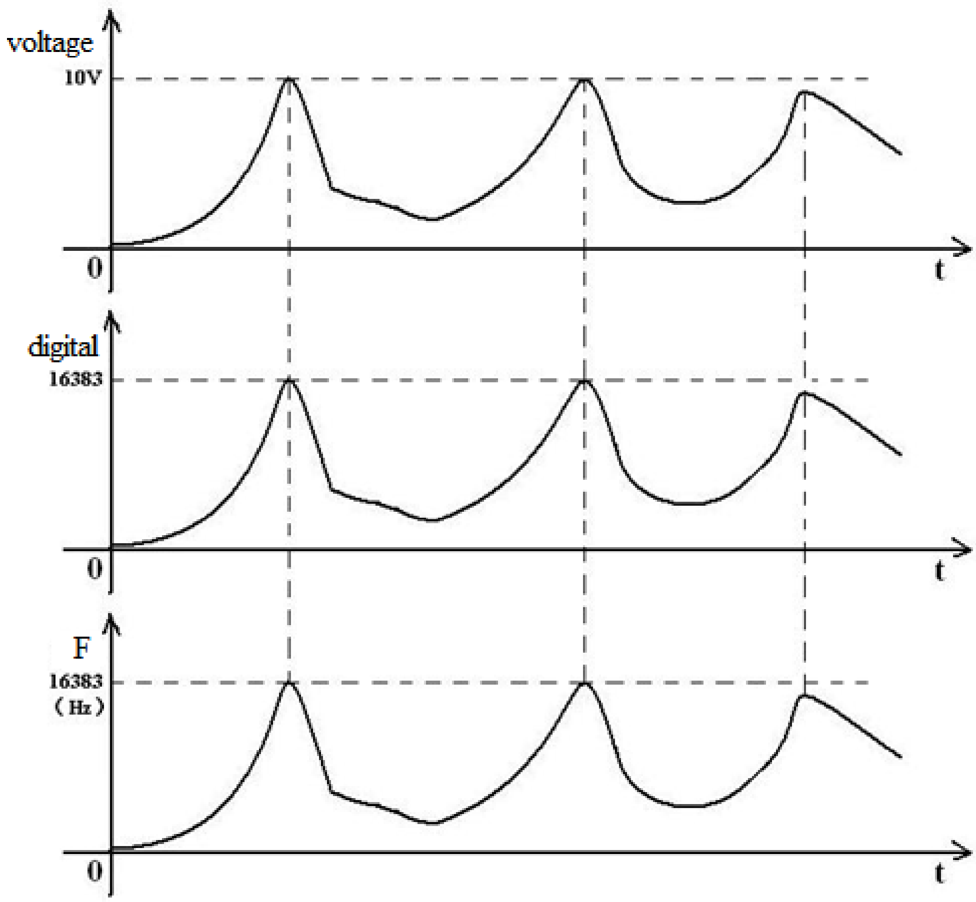
Structure diagram

For example: the output signal of potentiometer is shown as below:



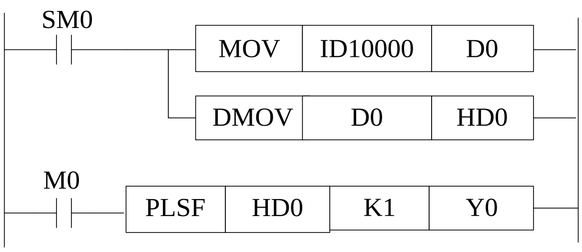
voltage signal diagram

The transformed digital value is 0 to 16383 of 0-10V voltage signal, which means the pulse frequency is 0~16383Hz (because of the response problem, PLSF acceleration deceleration time is 0). The relationship of voltage signal, digital value and pulse output frequency is shown as below diagram:



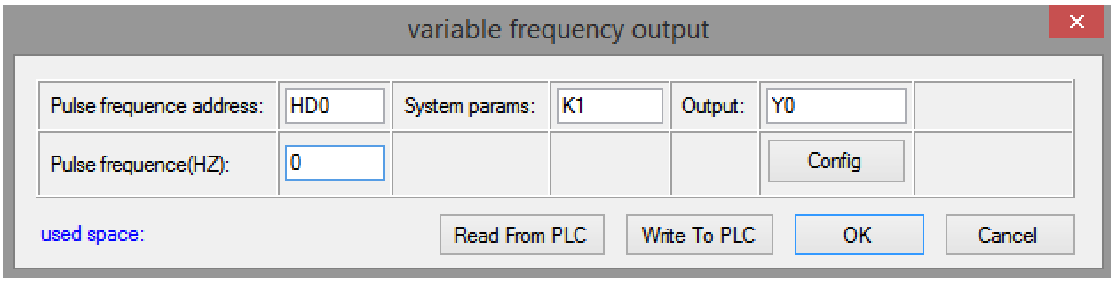
Relationship of voltage signal/digital value/pulse frequency

- Pulse instruction



- Software configuration

(1) Pulse segment configuration



(2) System parameters (relative mode)

Param	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete
Y0 axis-Common-Pulse num (1)	1
Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10

Param	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal

Param	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar)	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0

Param	Value
Y0 axis-group 1-Pulse default speed	0
Y0 axis-group 1-Acceleration time of Pulse default s...	0
Y0 axis-group 1-Deceleration time of pulse default s...	0
Y0 axis-group 1-Acceleration and deceleration time (ms)	0
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0

Note:

※1: As there are many configuration parameters of PLSF, we suggest to use software configuration table to set the parameters.

1-2-4. Relative single segment positioning [DRVI]

■ Instruction overview

Relative single segment positioning pulse instruction.

Relative single segment positioning [DRVI]			
16-bit instruction	-	32-bit instruction	DRVI
Execution condition	Rising/falling edge coil	Suitable model	XD, XL (except XD1, XL1)
Hardware	V3.3.1 and up	Software	V3.3 and up

■ Operand

Operand	Function	Type
S0	Pulse numbers or soft component address	32-bit, BIN
S1	Pulse frequency or soft component address	32-bit, BIN
S2	Pulse acceleration/deceleration time or soft component address	32-bit, BIN
D0	Pulse output terminal	Bit
D1	Pulse direction terminal	Bit

■ Suitable soft component

Word

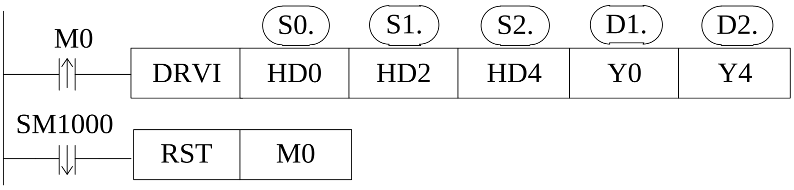
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●	●	●	●	●	●		
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		

Bit

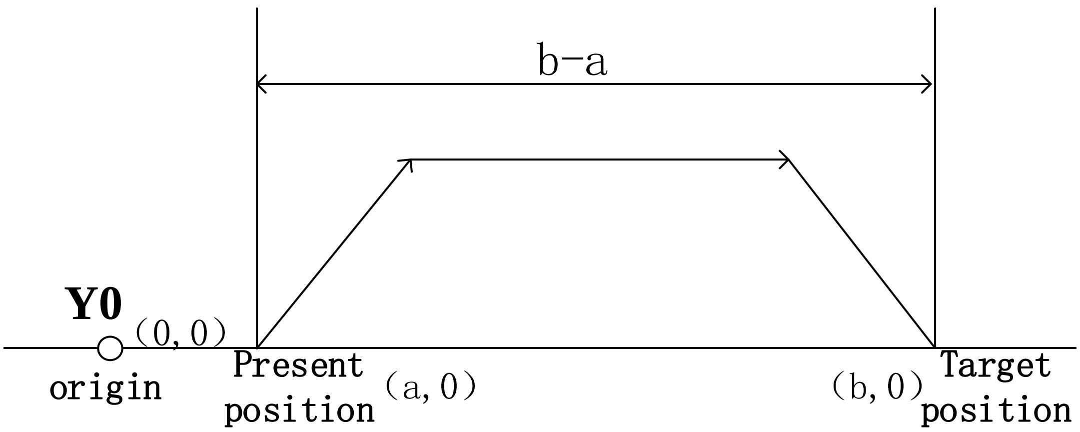
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D1		●					
D2		●					

*Note: D means D, HD. TD means TD, HTD. CD means CD, HCD, HSCD, HSD. DM means DM, DHM. DS means DS, DHS. M means M, HM, SM. S means S, HS. T means T, HT. C means C, HC.

Function and action



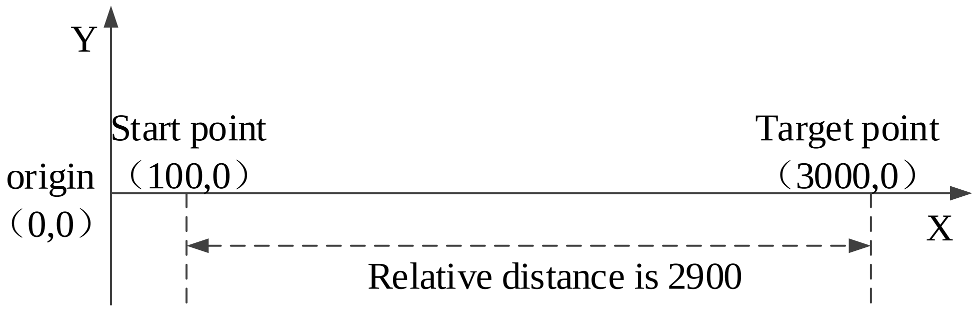
- Pulse frequency output range:1Hz ~100KHz (note: PLC can output 100~200KHz pulse, but we cannot ensure all the servo drive can work fine, please connect 500Ω resistor between output terminal and 24V power supply)
- Pulse numbers: K-2,147,483,648 ~ K2,147,483,647; negative value means output pulse in reverse direction.
- Relative driving mode: move from the present position (the distance between present position and target position), HSD0, HSD2, HSD4, HSD6…… are the reference point.



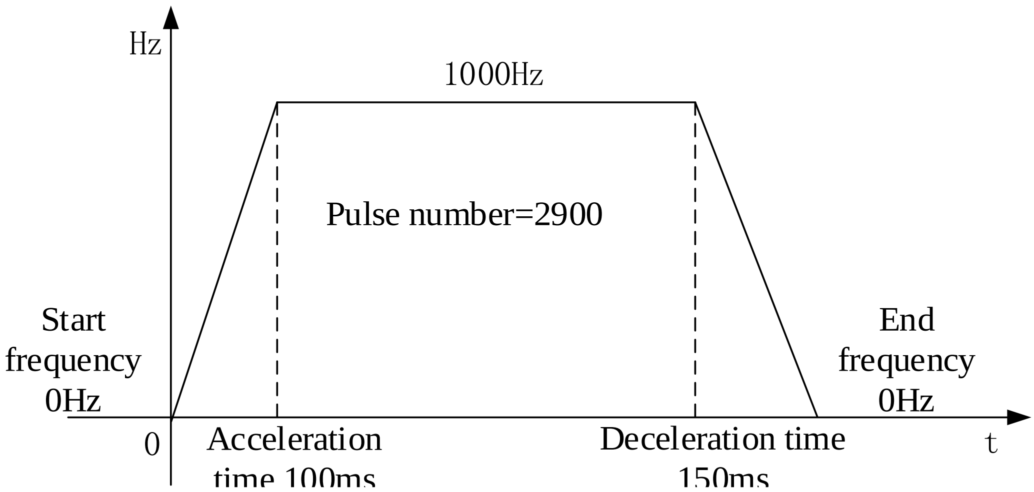
- The pulse number is accumulated in register HSD0 (double word).
- The pulse frequency can be real-time changed when the instruction is executing, the new frequency will be effective at once. (PLC firmware v3.4.5 and up can support)
- The acceleration and deceleration time is same for DRVI instruction.
- The direction of relative positioning instruction depends on S0 (pulse number), if the number of pulses is set to a positive value, the pulse is sent in forward direction and the accumulative pulse register (HSD0, HSD4...) value increases; if the number of pulses is set to a negative value, the pulse is sent in reverse direction and the accumulative pulse register (HSD0, HSD4...) value decreases.
- DRVI does not use the system parameter block configuration mode, if the public and the first set of parameters (except the deceleration parameters) are configured, they will be effective for DRVI.

Example 1

X axis present coordinates is (100, 0), it needs to move to target position (3000, 0) with the speed 1000Hz, start frequency and end frequency is 0Hz, pulse output terminal is Y0, direction terminal is Y4. As HSD0(dword) present value is 100, the relative distance from target position 3000 to present position 100 is 3000-100=2900. The execution diagram of DRVI is shown as below:

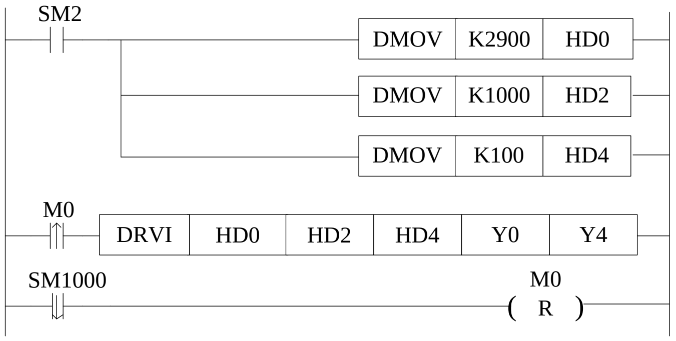


Pulse coordinates diagram



Pulse curve diagram

➢ **Program:**



//set pulse number to 2900

//set pulse frequency to 1000

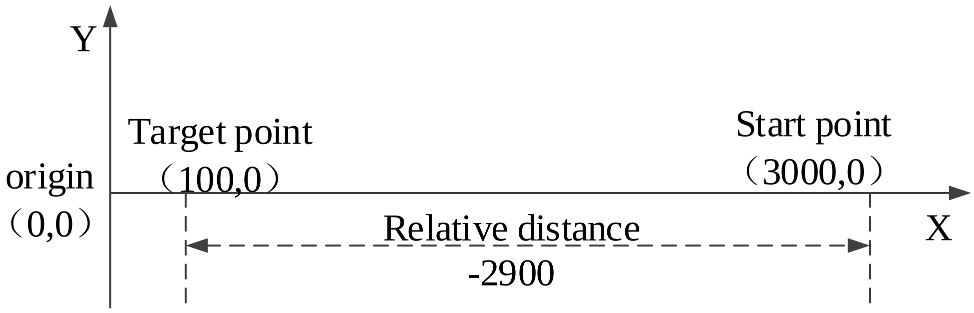
//set acceleration/deceleration to 100

//execute the DRVI instruction

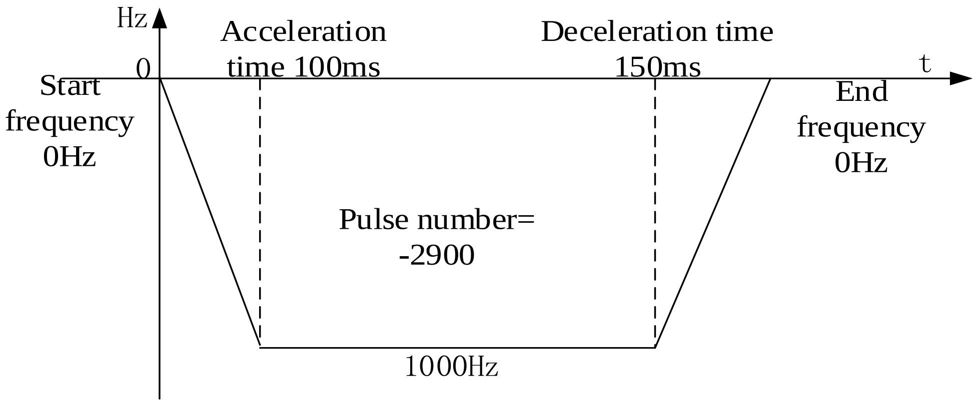
//reset M0 at the falling edge of pulse outputting end flag

Example 2

X axis present coordinates is (3000, 0), it needs to move to target position (100, 0) with the speed 1000Hz, start frequency and end frequency is 0Hz, pulse output terminal is Y0, direction terminal is Y4. As HSD0(dword) present value is 3000, the relative distance from target position 100 to present position 3000 is 100-3000=-2900. The execution diagram of DRVI is shown as below:

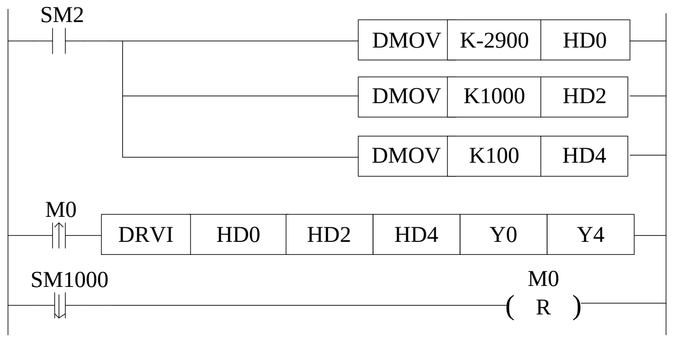


Pulse coordinate diagram



Pulse curve diagram

➢ **Program:**



//set pulse number to -2900

//set pulse frequency to 1000

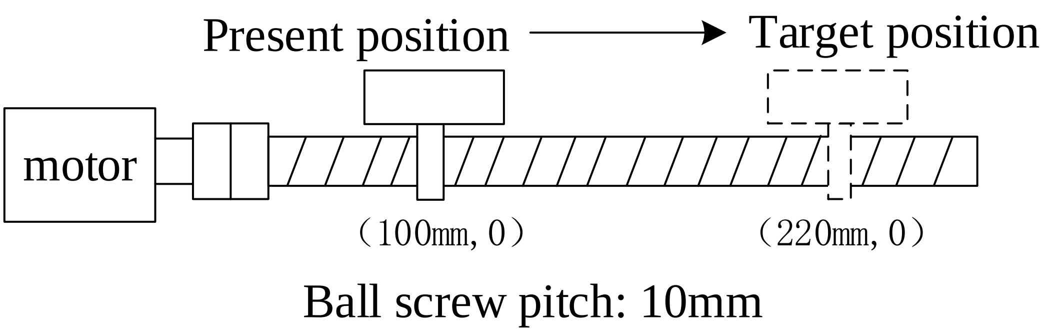
//set acceleration/deceleration to 100

//execute the DRVI instruction

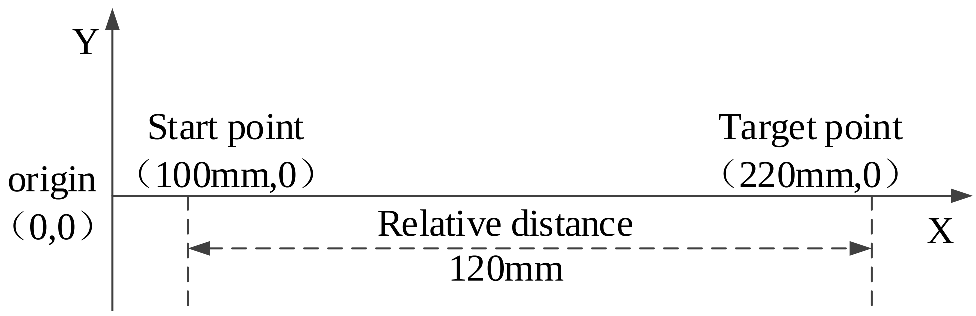
//reset M0 at the falling edge of pulse outputting end flag

Example 3

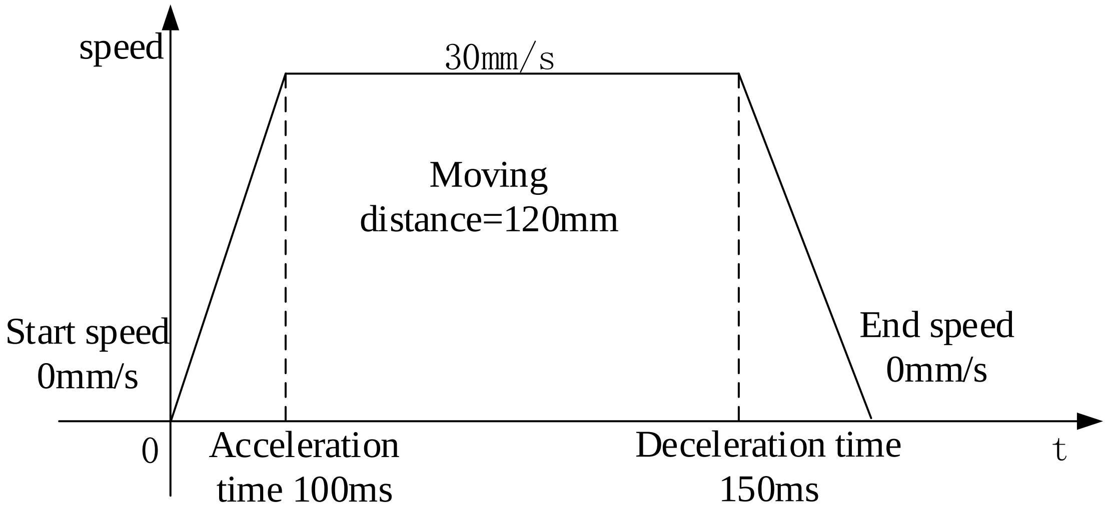
There is a ball screw workbench, the motor has 5000 pulses per circle, X axis present coordinate is (100mm, 0), start speed and end speed is 0mm/s, it needs to reach the target position (220mm, 0) with the speed 15000（30mm/s）, the pulse output terminal is Y0, pulse direction terminal is Y4,as the accumulated pulse number register HSD0 present value is 50000 (100mm), the relative distance from target position 110000 (220mm) to present position 50000 (100mm) is 60000=110000-50000. The execution diagram of DRVI is shown as below:



Ball srew diagram

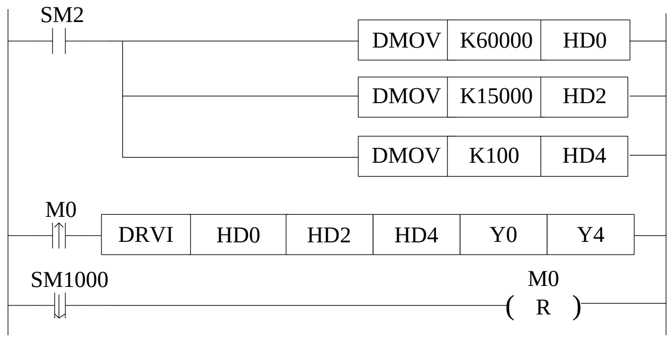


Pulse coordinate diagram



pulse curve diagram

- **Program:**



//set moving distance to 60000

//set speed to 15000

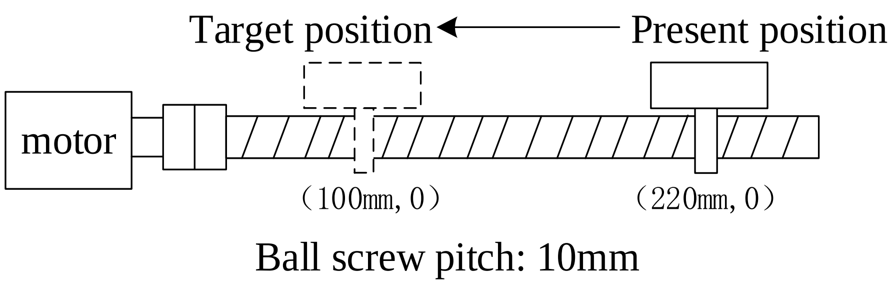
//set acceleration/deceleration to 100

//execute the DRVI instruction

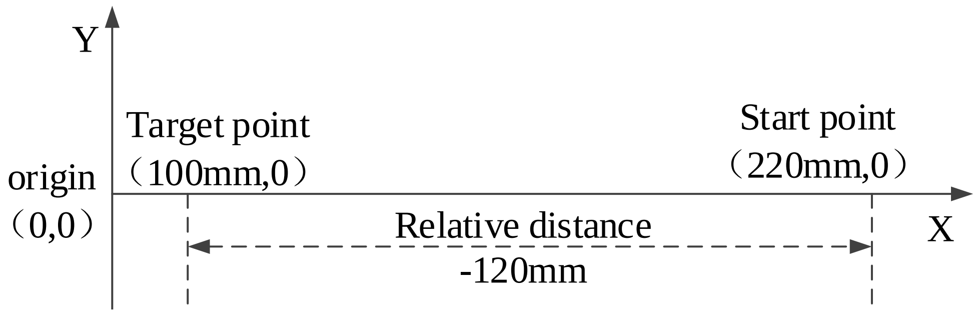
//reset M0 at the falling edge of pulse outputting end flag

Example 4

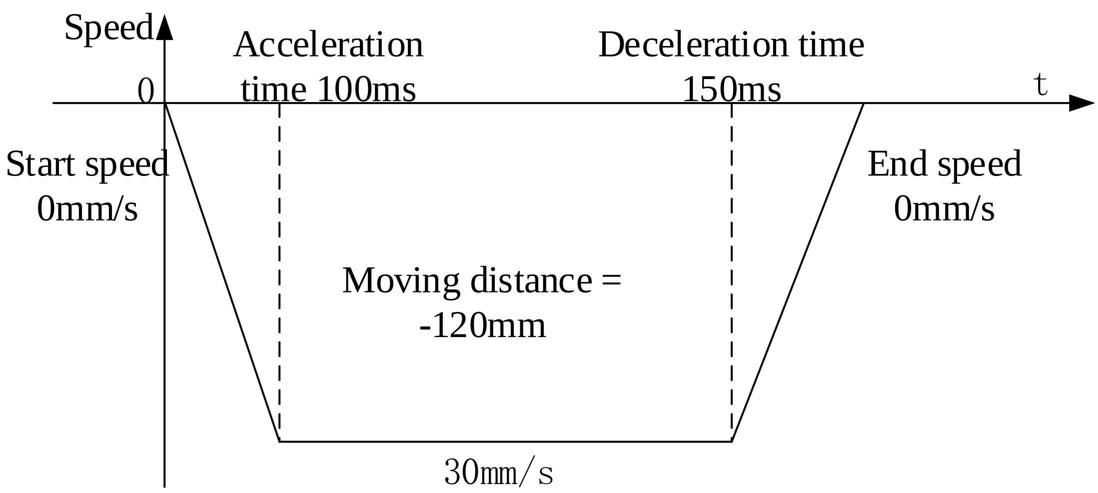
There is a ball screw workbench, the motor has 5000 pulses per circle, X axis present coordinate is (220mm, 0), start speed and end speed is 0mm/s, it needs to reach the target position (100mm, 0) with the speed 15000（30mm/s）, the pulse output terminal is Y0, pulse direction terminal is Y4,as the accumulated pulse number register HSD0 present value is 110000 (220mm), the relative distance from target position 50000 (100mm) to present position 110000 (220mm) is -60000=50000-110000. The execution diagram of DRVI is shown as below:



Ball screw diagram

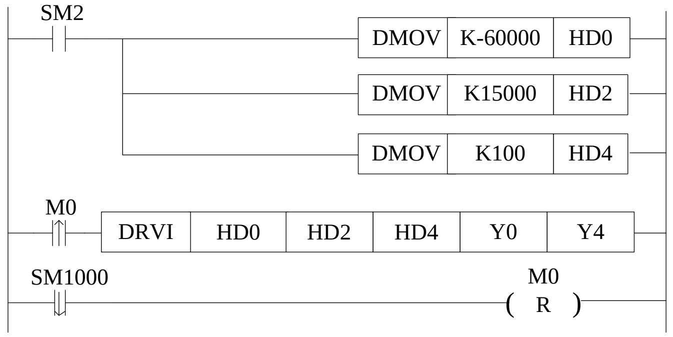


Pulse coordinate diagram



Pulse curve diagram

- **Program:**



//set moving distance to -60000

//set speed to 15000

//set acceleration/deceleration to 100

//execute the DRVI instruction

//reset M0 at the falling edge of pulse outputting end flag

1-2-5. Absolute single-segment positioning [DRVA]

1. Instruction summarization

Absolute single-segment positioning instruction.

Absolute single-segment positioning [DRVA]			
16-bit instruction	-	32-bit instruction	DRVA
Execution condition	Rising/falling edge of the coil	Suitable model	XD, XL (except XD1, XL1)
Hardware	V3.3.1 and up	Software	V3.3 and up

2. operand

Operand	Function	Type
S0	Output pulse numbers register address	32-bit, BIN
S1	Output pulse frequency register address	32-bit, BIN
S2	Pulse acceleration/deceleration time register address	32-bit, BIN
D0	Pulse output terminal	Bit
D1	Pulse output direction	Bit

3. Suitable soft component

word

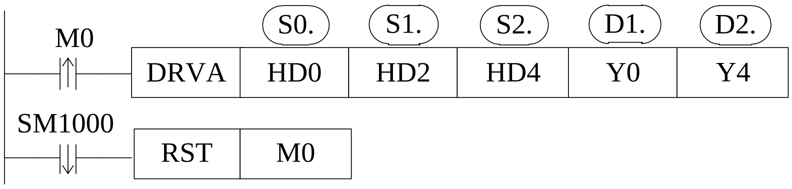
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●	●	●	●	●	●		
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		

Bit

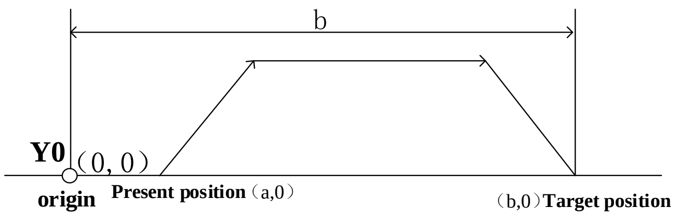
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*Note: D means D, HD. TD means TD, HTD. CD means CD, HCD, HSCD, HSD. DM means DM, DHM.
DS means DS, DHS. M means M, HM, SM. S means S, HS. T means T, HT. C means C, HC.

Function and action



- Pulse frequency output range:1Hz ~100KHz (note: PLC can output 100~200KHz pulse, but we cannot ensure all the servo drive can work fine, please connect 500Ω resistor between output terminal and 24V power supply)
- Pulse numbers: K-2,147,483,648 ~ K2,147,483,647; negative value means output pulse in reverse direction.
- Absolute driving mode: move from the origin point (the distance between origin position and target position), origin point is the reference point.

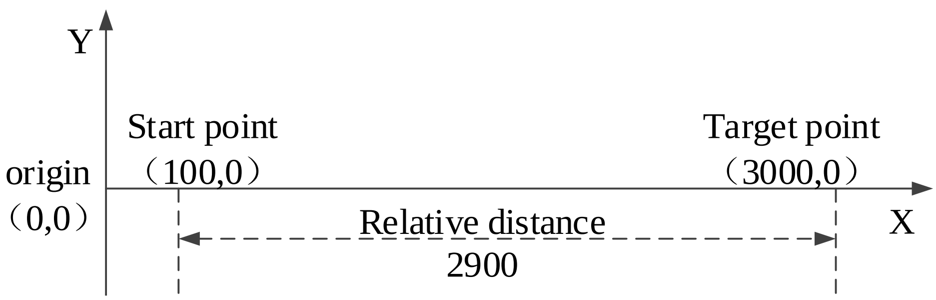


- DRVA does not use the system parameter block configuration mode, if the public and the first set of parameters (except the deceleration parameters) are configured, they will be effective for DRVA.
- The pulse number is accumulated in register HSD0 (double word).
- The pulse frequency can be real-time changed when the instruction is executing, the new frequency will be effective at once. (PLC firmware v3.4.5 and up can support)

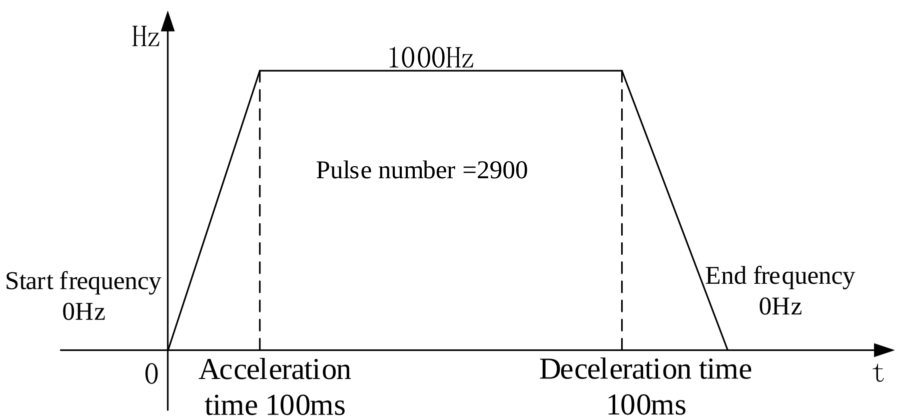
- The acceleration and deceleration time is same for DRVA instruction.
- The direction of absolute positioning instruction depends on whether the target position is larger than present position, if the target position is larger than present position(the target position is on the right of present position on the axis), the pulse is sent in forward direction and the accumulative pulse register (HSD0, HSD4...) value increases; if the target position is smaller than present position(the target position is on the left of present position on the axis), the pulse is sent in reverse direction and the accumulative pulse register (HSD0, HSD4...) value decreases, if the target position is equal to present position(the target position overlaps present position on the axis), it will not send pulse.
- When S0 parameters are same to pulse accumulated register HSD0, SM1000 will not act, no falling edge.

Example 1

X axis present coordinates is (100, 0), it needs to move to target position (3000, 0) with the speed 1000Hz, start frequency and end frequency is 0Hz, pulse output terminal is Y0, direction terminal is Y4. As HSD0(dword) present value is 100, the target position is 3000, target position is larger than present position, send forward direction pulse, the execution diagram of DRVA is shown as below:

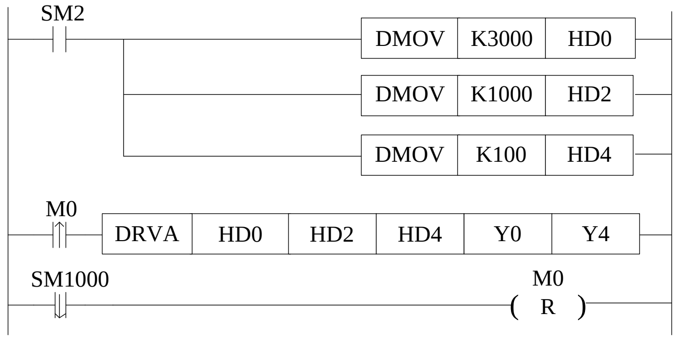


Pulse coordinate diagram



Pulse curve diagram

- **Program:**



//set pulse number to 3000

//set pulse frequency to 1000

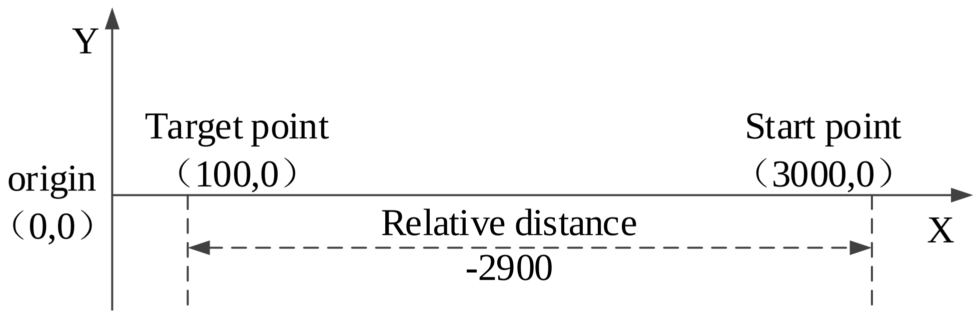
//set acceleration/deceleration to 100

//execute the DRVA instruction

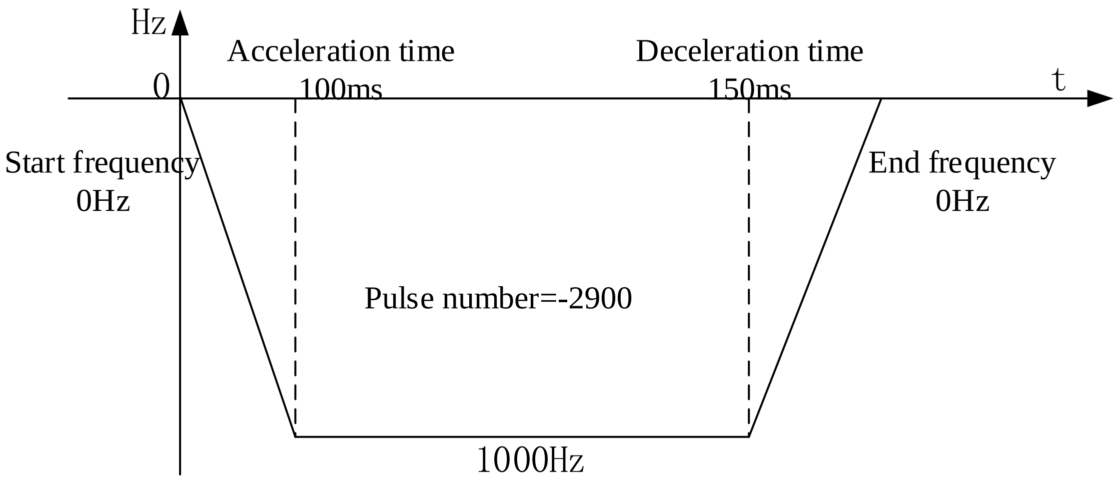
//reset M0 at the falling edge of pulse outputting end flag

Example 2

X axis present coordinates is (3000, 0), it needs to move to target position (100, 0) with the speed 1000Hz, start frequency and end frequency is 0Hz, pulse output terminal is Y0, direction terminal is Y4. As HSD0(dword) present value is 3000, the target position is 100, present position is 3000, the relative ditance is 100-3000=-2900, the execution diagram of DRVA is shown as below:

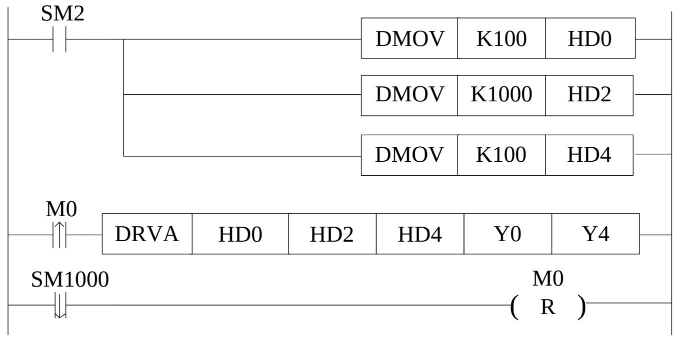


Pulse coordinate diagram



Pulse curve diagram

➢ **Program:**



//set pulse number to 100

//set pulse frequency to 1000

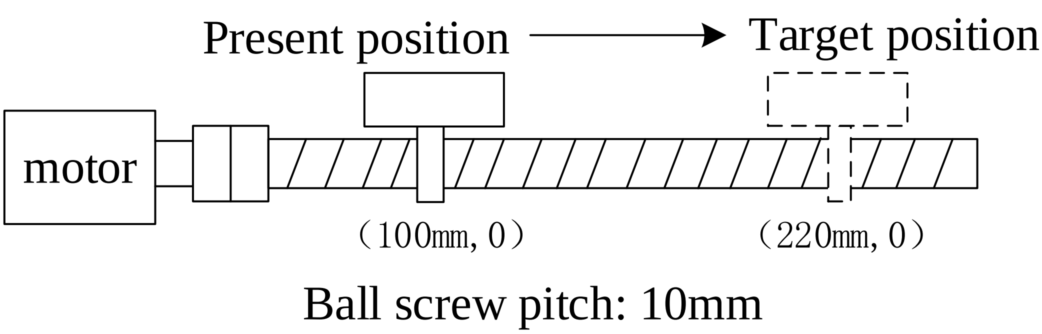
//set acceleration/deceleration to 100

//execute the DRVA instruction

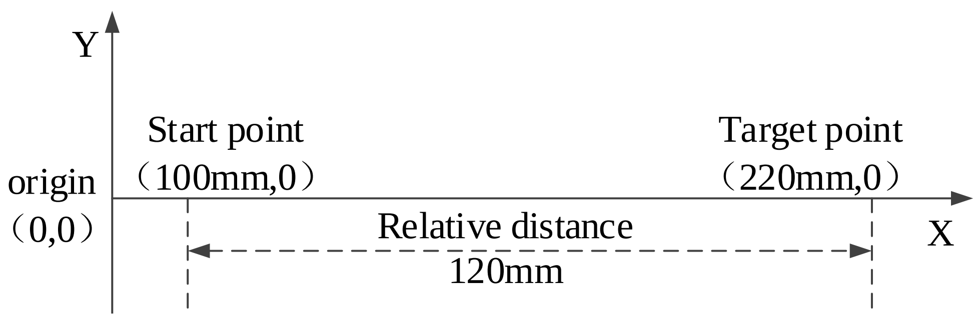
//reset M0 at the falling edge of pulse outputting end flag

Example 3

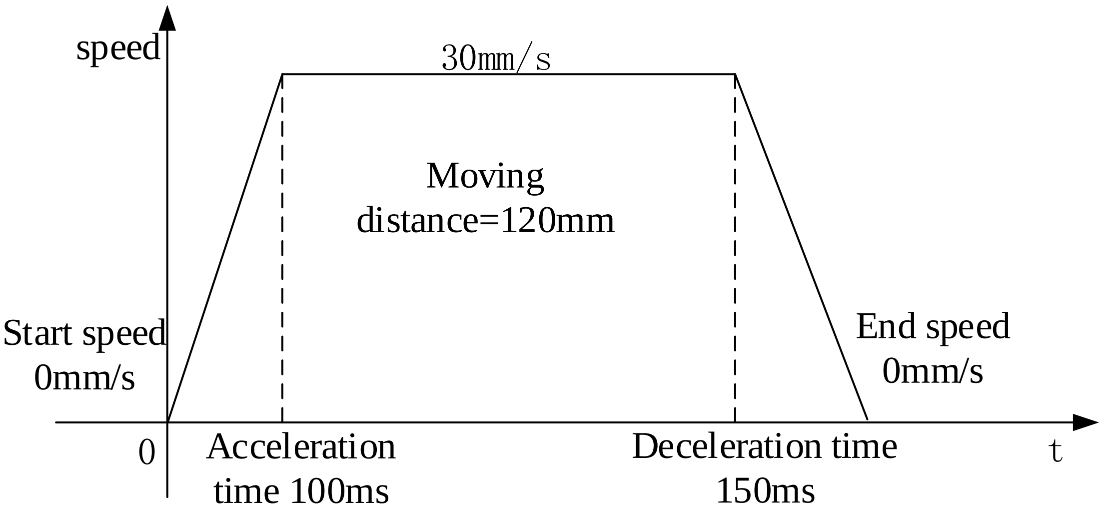
There is a ball screw workbench, the motor has 5000 pulses per circle, X axis present coordinate is (100mm, 0), start speed and end speed is 0mm/s, it needs to reach the target position (220mm, 0) with the speed 15000 （30mm/s）, the pulse output terminal is Y0, pulse direction terminal is Y4, as the accumulated pulse number register HSD0 present value is 50000 (100mm), the relative distance from target position 110000 (220mm) to present position 50000 (100mm) is 60000=110000-50000. The execution diagram of DRVA is shown as below:



Ball srew diagram

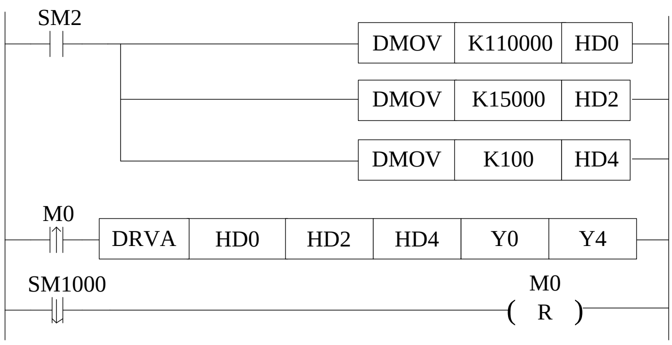


Pulse coordinate diagram



pulse curve diagram

➢ **Program:**



//set moving distance to 110000

//set speed to 15000

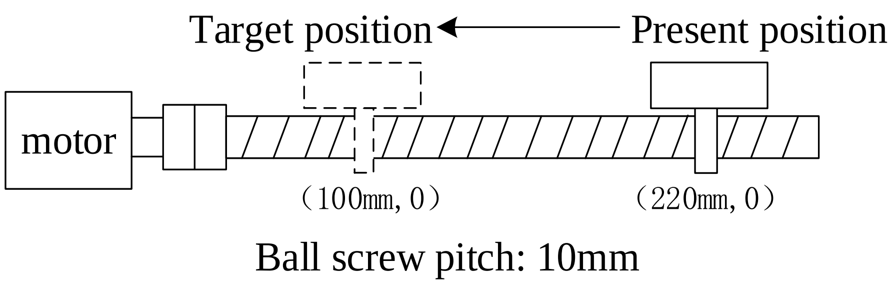
//set acceleration/deceleration to 100

//execute the DRVA instruction

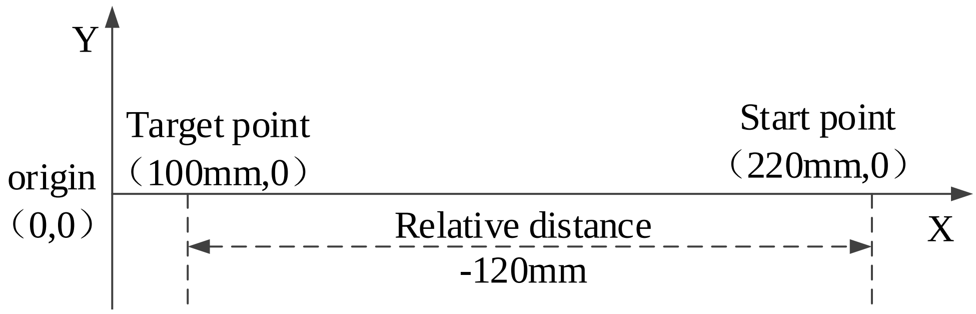
//reset M0 at the falling edge of pulse outputting end flag

Example 4

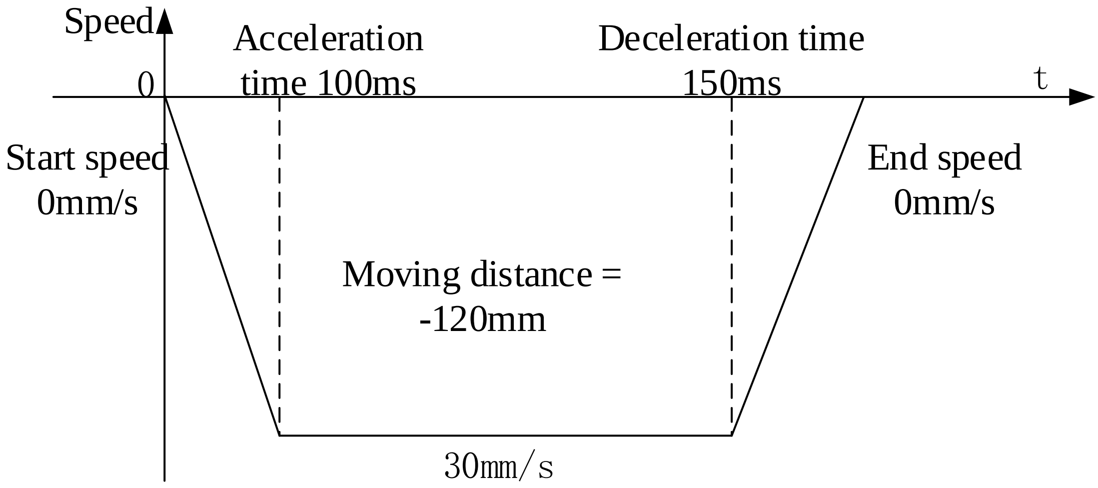
There is a ball screw workbench, the motor has 5000 pulses per circle, X axis present coordinate is (220mm, 0), start speed and end speed is 0mm/s, it needs to reach the target position (100mm, 0) with the speed 15000（30mm/s）, the pulse output terminal is Y0, pulse direction terminal is Y4, as the accumulated pulse number register HSD0 present value is 110000 (220mm), the relative distance from target position 50000 (100mm) to present position 110000 (220mm) is -60000=50000-110000. The execution diagram of DRVA is shown as below:



Ball screw diagram

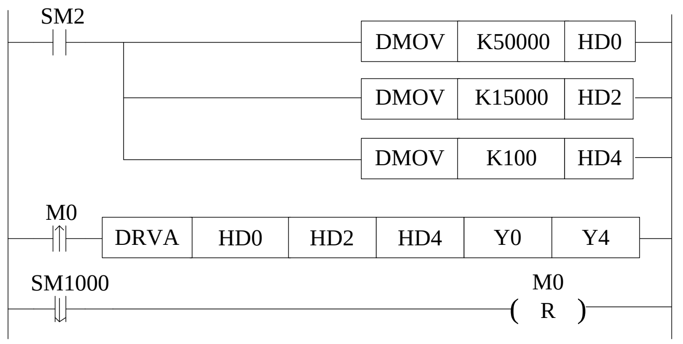


Pulse coordinate diagram



Pulse curve diagram

➢ **Program:**



//set moving distance to 50000

//set speed to 15000

//set acceleration/deceleration to 100

//execute the DRVA instruction

//reset M0 at the falling edge of pulse outputting end flag

1-2-6. Mechanical origin return[ZRN]

1. Instruction overview

Mechanical origin return instruction. (note: ZRN cannot support the function of soft limit and origin auxiliary signal)

Mechanical origin return [ZRN]			
16-bit instruction		32-bit instruction	ZRN
Execution condition	Rising/falling edge of the coil	Suitable model	XD, XL (except XD1, XL1)
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S	System parameter block address	32-bit, double words
D	Pulse output terminal	Bit

3. Suitable soft component

word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●		

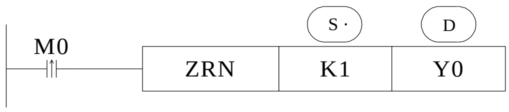
Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●					

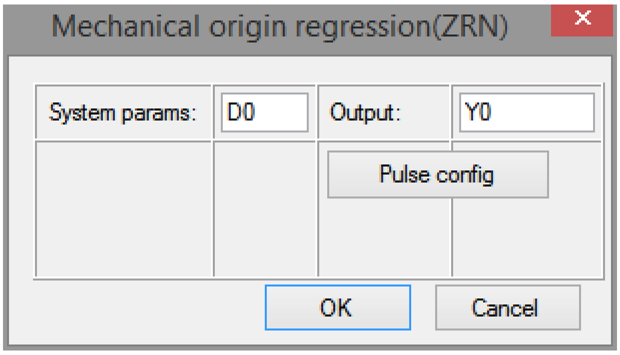
*Note: D means D, HD. TD means TD, HTD. CD means CD, HCD, HSCD, HSD. DM means DM, DHM.

DS means DS, DHS. M means M, HM, SM. S means S, HS. T means T, HT. C means C, HC.

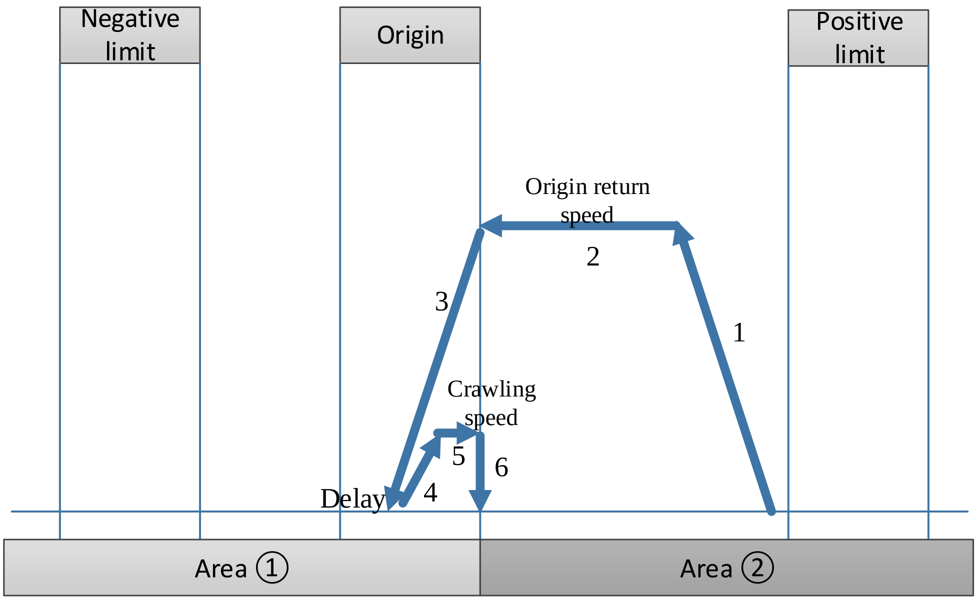
Function and action



- The system parameter block please refer to chapter 1-2-1-3.
- ZRN instruction panel configuration is shown as below:

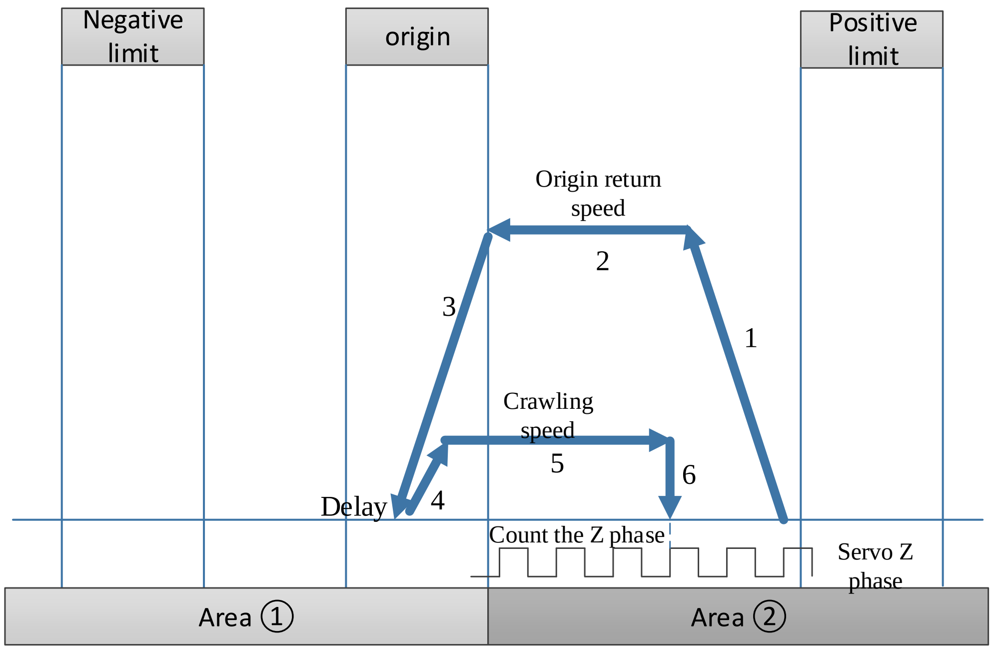


- Mechanical origin returning diagram:



Note:

If setting the servo Z phase, it starts to count the Z phase signal at the monment of leaving the origin signal with crawling speed (5), it stops mechanical origin return instruction after Z phase signal counting reached, please see below diagram:



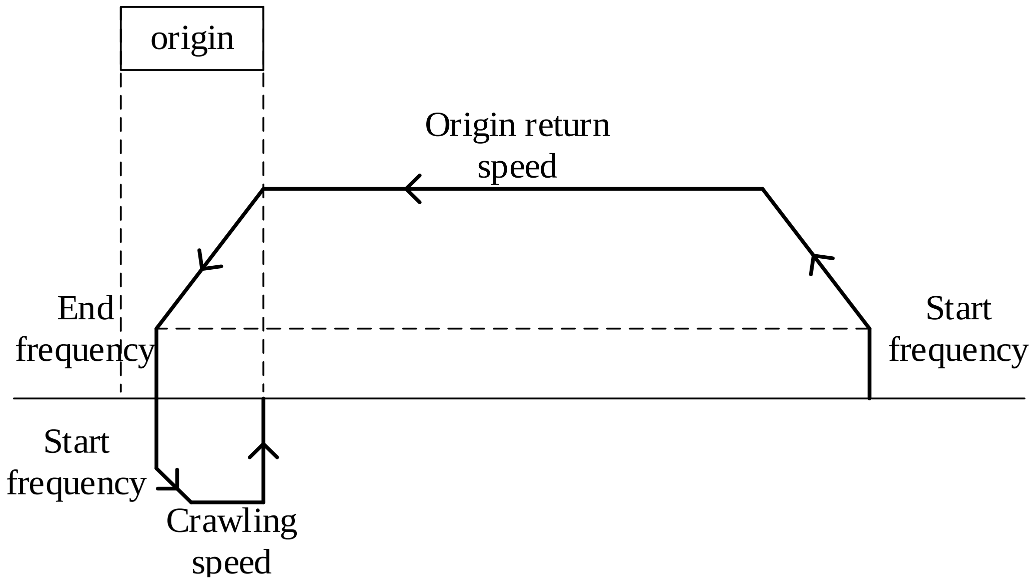
- Mechanical origin return movement

(1) when the origin return starts, it accelerates as the acceleration slope, after reaching the origin return speed, it will move towards origin return direction with this speed.

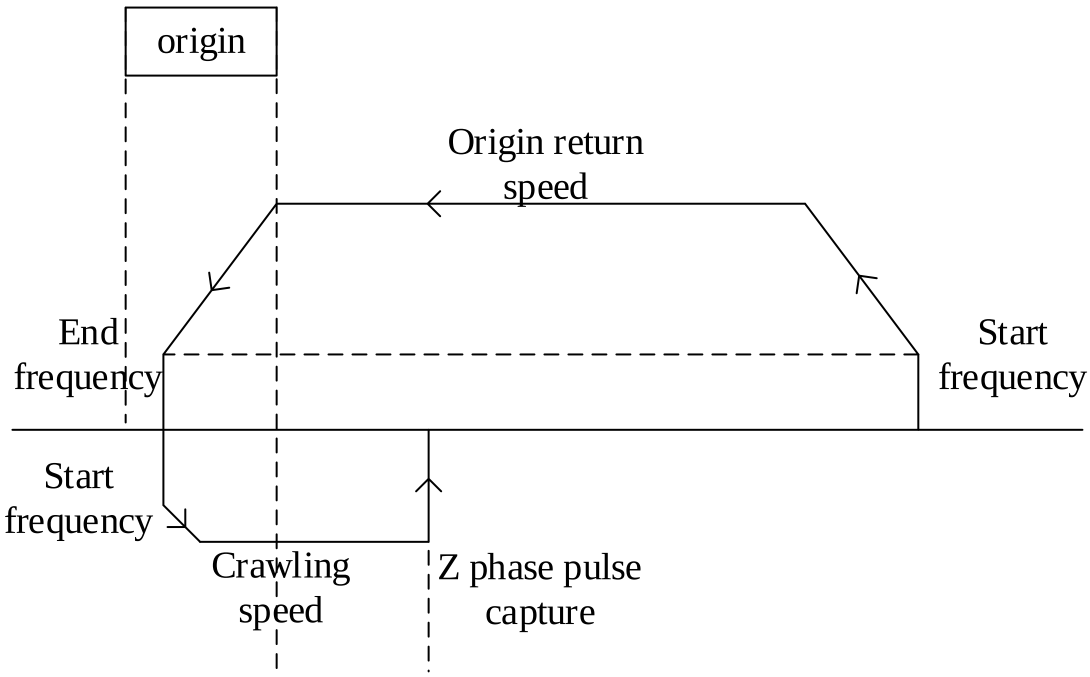
(2) when it meets the rising edge of origin signal, it will decelerate with deceleration slope until stop(frequency =0).

(3) delay(direction delay time in SFD), then accelerate with acceleration slope until reaching the crawling speed, it stops origin return action at the moment of leaving the origin signal falling edge (if setting the Z phase pulse, it starts counting the Z phase after leaving the origin signal falling edge, it will stop origin return action after the counting value reached).

(4) if setting the origin return clear signal CLR, it will output CLR signal and delay (the CLR signal delay time in SFD, CLR signal can be used to clear the servo motor error counter), finally, copy the mechanical origin position to present position and the origin return action finished.

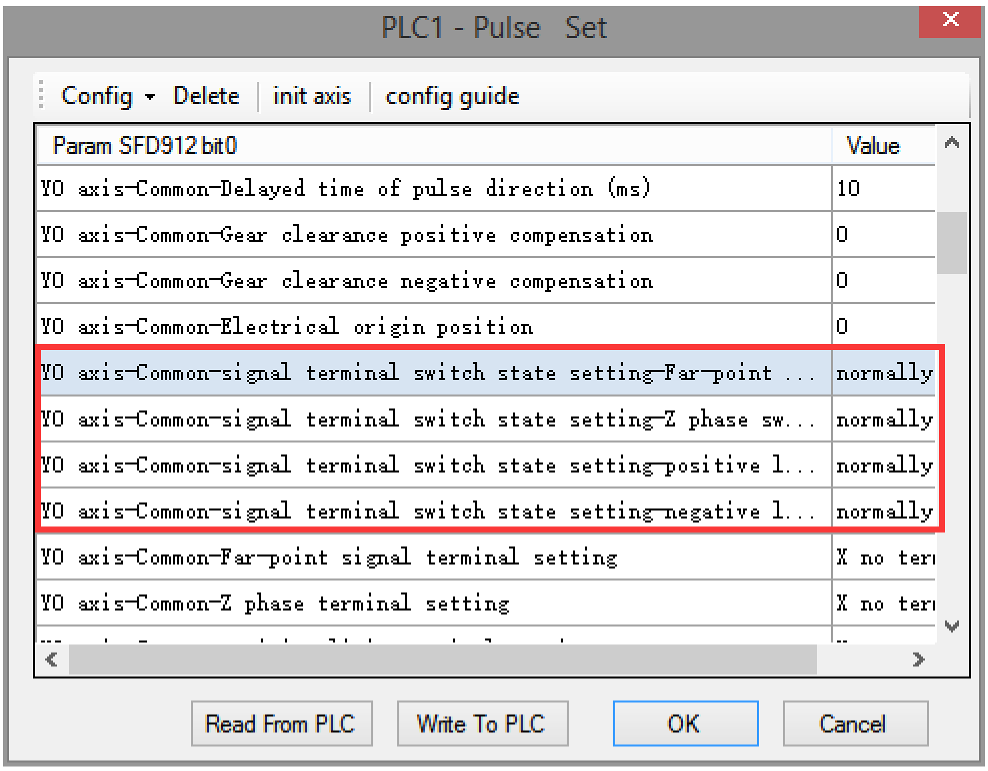


No Z phase capture



Z phase capture

Mechanical origin input terminal positive/negative logic (normally on/off) setting:



Mechanical orgin return setting notes:

The origin signal terminal can select all input points on the PLC; However, if the selected input point is the external interrupt terminal on the PLC, the process of returning to the mechanical origin will be processed according to the interrupt, so as to further improve the accuracy of returning to the mechanical origin (it will not be affected if Z phase is used to return to the origin). The selected input point is the external interrupt terminal not from the PLC, which will be affected by the scanning cycle of PLC in the process of mechanical origin (it will not be affected if Z phase is used to return to the origin). For detailed external interrupt terminals, please refer to appendix 4 of this manual.

Pulse output terminal configuration table:

PLC mode	Pulse channel	Pulse output terminal	Max output frequency	Output mode	Output mode
XD2-16T/RT XD2-24T/RT XD2-32T/RT XD2-48T/RT XD2-60T/RT	2	Y0, Y1	0~100KHz	Open collector mode	Pulse + direction
XD3-16T/RT XD3-24T/RT XD3-32T/RT XD3-48T/RT XD3-60T/RT	2	Y0, Y1	0~100KHz	Open collector mode	Pulse + direction
XD5-16T/RT XD5-24T/RT XD5-32T/RT XD5-48T/RT XD5-60T/RT	2	Y0、Y1	0~100KHz	Open collector mode	Pulse + direction
XD5-24T4 XD5-32T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector mode	Pulse + direction
XD5-48T6 XD5-60T6	6	Y0, Y1, Y2, Y3, Y4, Y5	0~100KHz	Open collector mode	Pulse + direction
XDM-24T4 XDM-32T4 XDM-60T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector mode	Pulse + direction
XDM-60T10	10	Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11	0~100KHz	Open collector mode	Pulse + direction
XD5E-30T4	4	Y0, Y1, Y2, Y3	0~100KHz	Open collector mode	Pulse + direction
XL3-16	2	Y0, Y1	0~100KHz	Open collector	Pulse + direction

				mode	

Note:

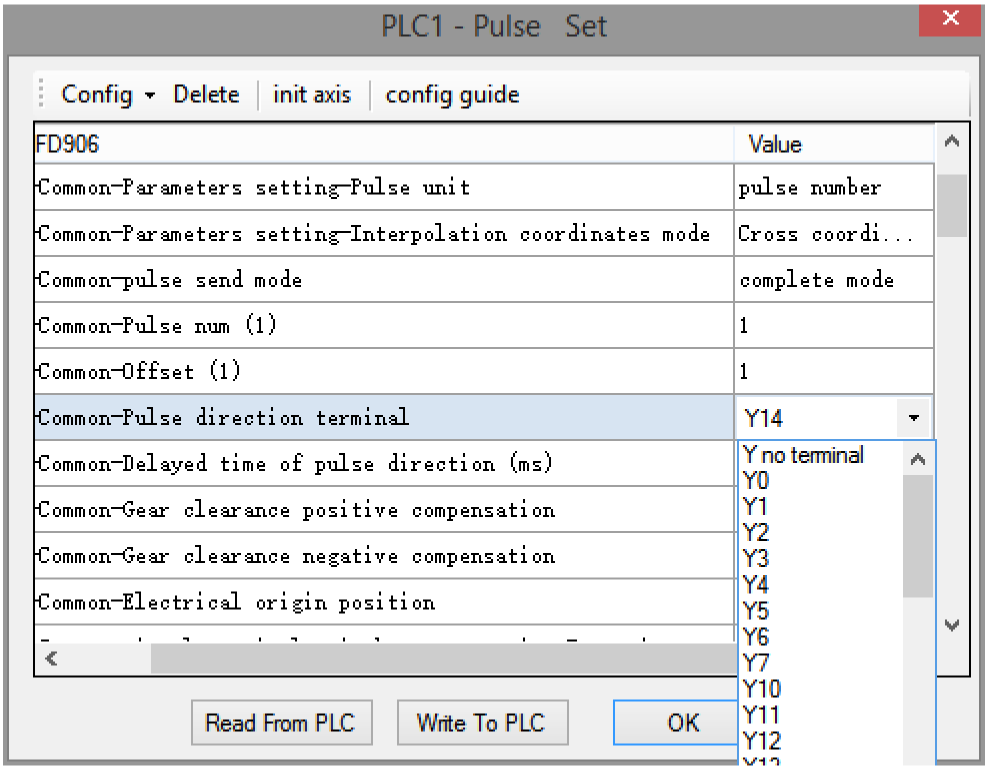
1: PLC can output 100 KHz to 200 KHz pulses, but we cannot sure that all servo is running, please connect 500 Ω resistance between output and 24V power supply.

2. when using the positioning command, the pulse direction terminal can be freely defined in all the output transistor terminals except the pulse output terminal;

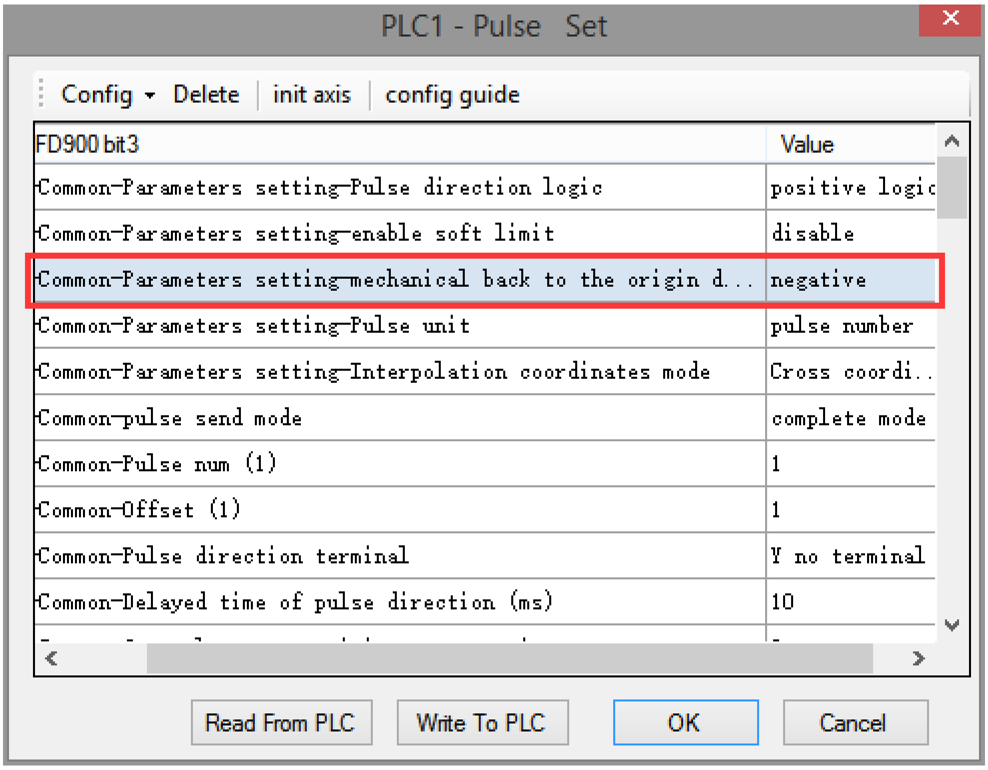
3. response time of pulse output transistor is 0.5us, response time of other output transistors is below 0.2ms.

4. when the pulse output terminal does not make the pulse output, it can also be used as the pulse direction terminal.

Mechanical origin returning pulse direction signal:

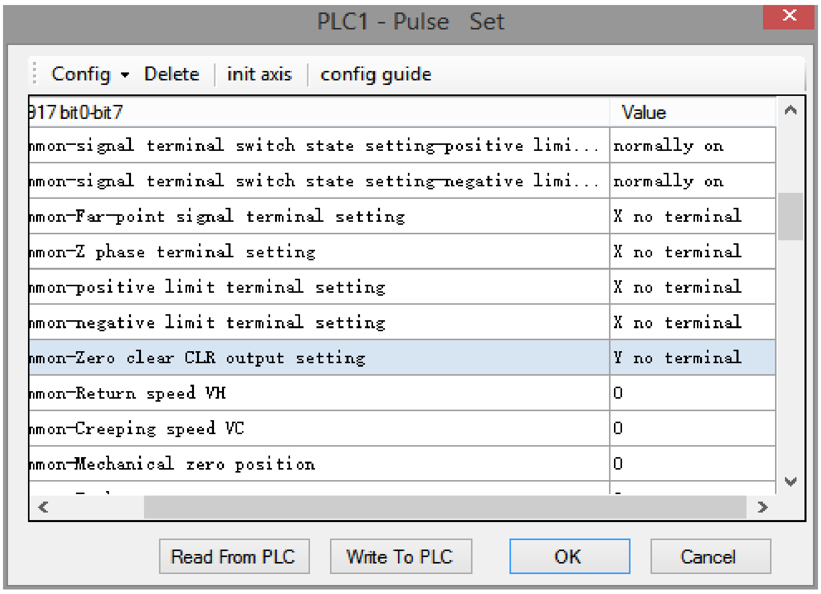


Origin direction setting of mechanical origin returning:



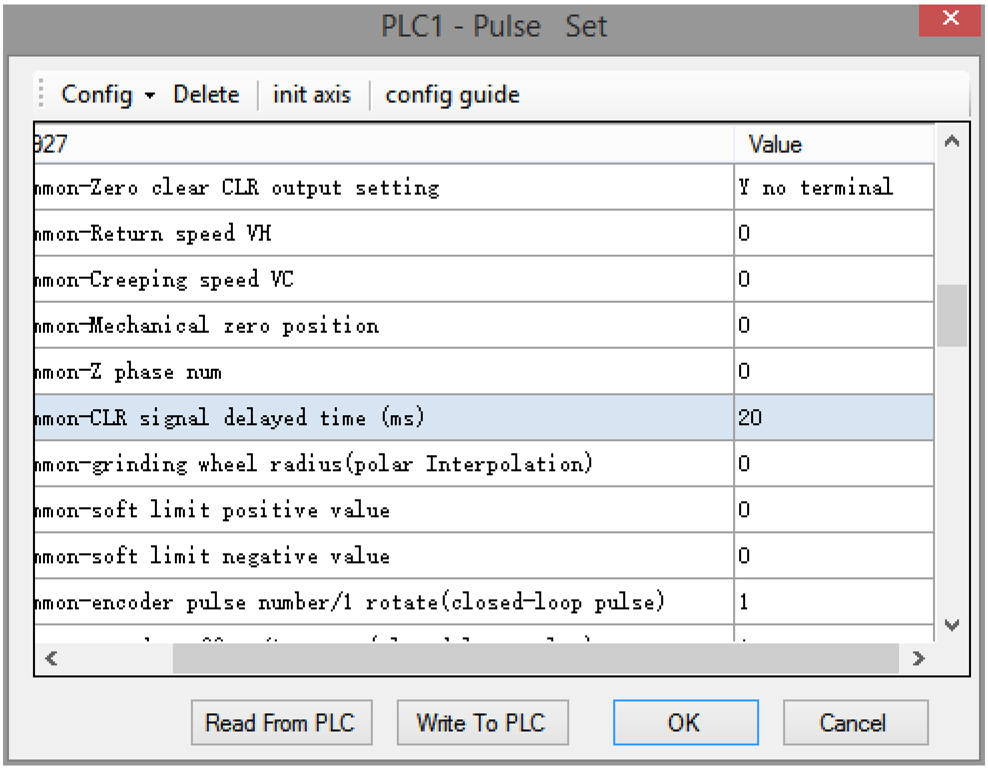
Clear output signal CLR

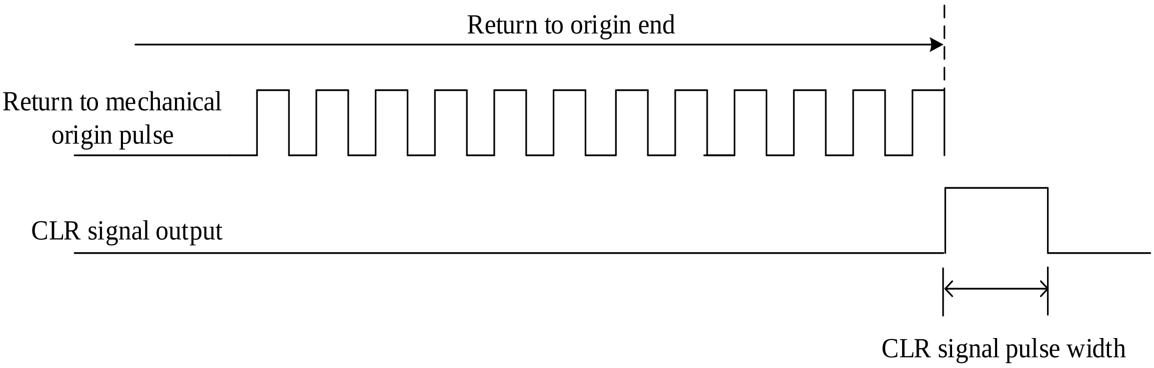
CLR signal setting, to output an output signal immediately after the end of returning to the mechanical origin, this signal can be sent to some other control equipment to achieve the purpose of rapid information transmission between each other. For example, after returning to the mechanical origin, the CLR signal is output to the servo driver immediately, so as to output clearance signal to clear the Error Counter of the servo motor. At last, copy the mechanical origin position value to the current position and the origin returning action is completed. The parameter configuration table is as follows:



CLR signal delay time:

the pulse width of CLR signal outputting after mechanical origin returning, the unit is ms, range is 0~32767 (default 20ms). The parameter configuration table is as follows:





CLR signal diagram

Note:

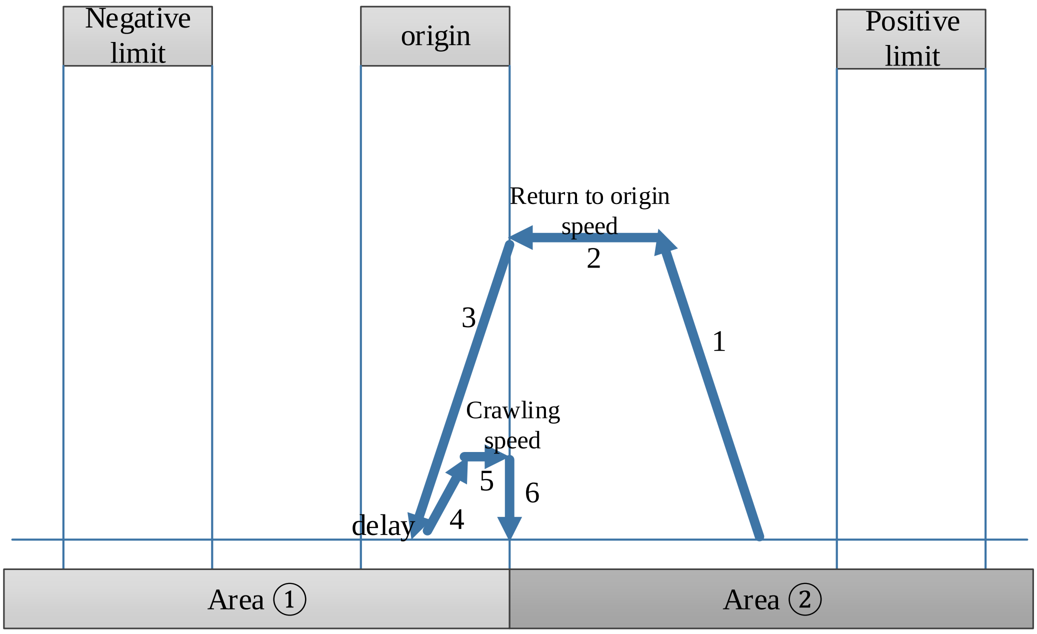
1. The CLR signal output terminal should use the output terminal of the PLC.
2. Do not set the delay time of CLR signal too small, or the servo driver may be unable to receive the CLR signal.

Motion analysis

1. The table is in area 2 when ZRN instruction started:

When the table is in area 2, it can be subdivided into three situations: the table is between the origin and the positive limit, the table is in the positive limit and the table is out of the positive limit.

(1) The workbench is between origin and positive limit, return to origin in reverse direction



Reverse return to origin

Actions:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and the acceleration is accelerated to the origin regression speed, and then the regression speed of the origin is pushed back toward the mechanical origin direction.

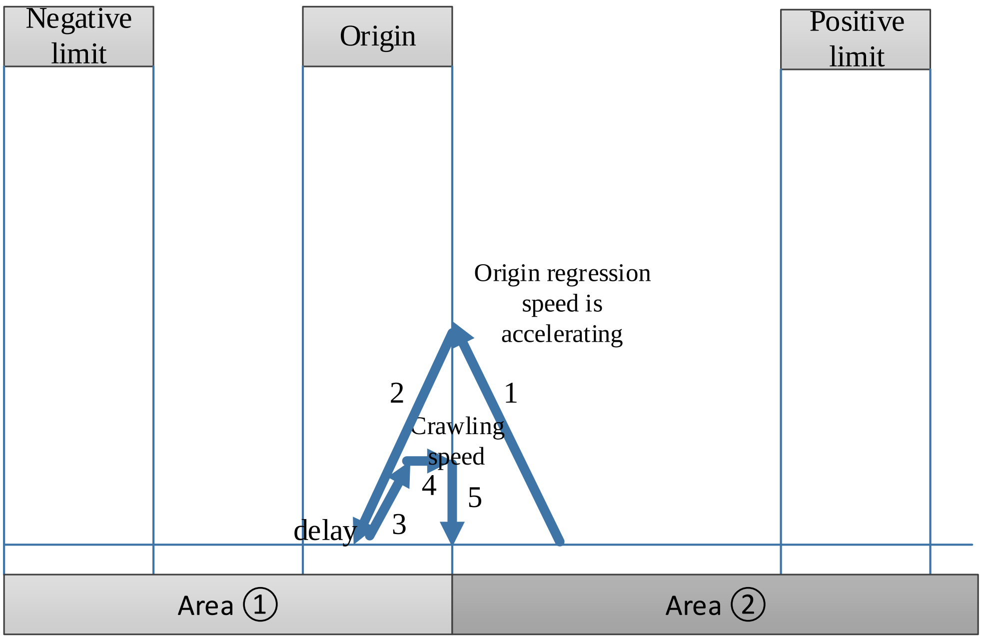
(2) When encountering the rising edge of the mechanical origin signal, slow down with the set deceleration slope until the deceleration to complete rest (frequency =0).

(3) delay (direction delay time in SFD), and then accelerate as the set acceleration slope, move forward until reaching the crawling speed, when leaving the mechanical origin falling edge signal instantaneous stop zero movement (if it sets the Z phase pulse, it starts to count Z phase signal after leaving the origin signal falling edge, then immediately stop motion when the counting reached).

(4) If "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

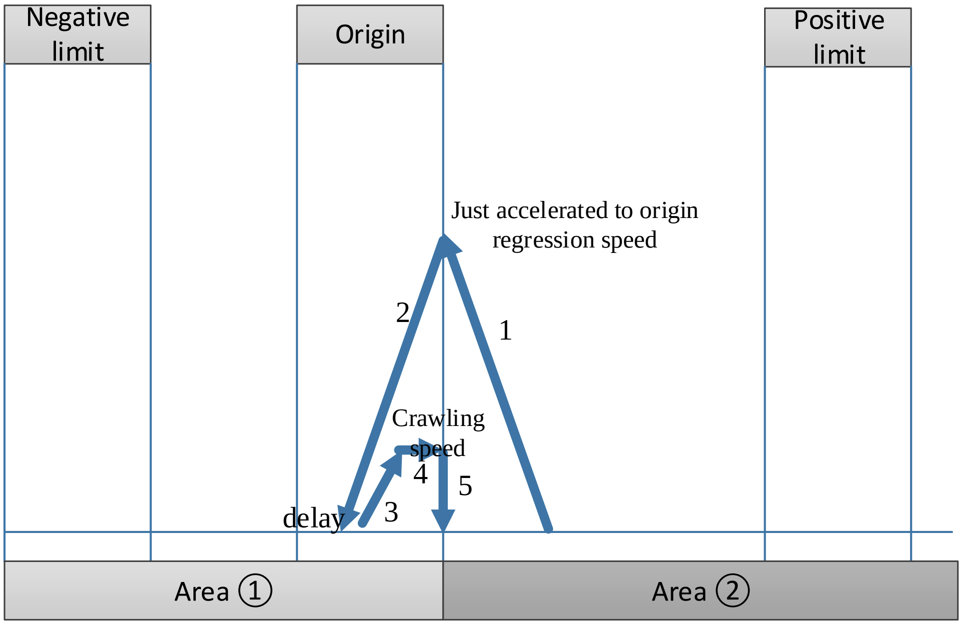
Special case 1:

When the acceleration of the just started ZRN instruction has reached the rising edge of the mechanical origin signal, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); delay (direction delay time in SFD) and then run in reverse direction at low speed as acceleration slope until reach origin regression speed, when leaving the origin falling edge signal instantaneous stop zero movement (if it sets the Z phase pulse, it starts to count Z phase signal after leaving the origin signal falling edge, then immediately stop motion when the counting reached), if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Special case 2:

When the acceleration of the just started ZRN instruction, it just accelerated to origin regression speed and reached the rising edge of the mechanical origin signal, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); delay (direction delay time in SFD) and then run in reverse direction at low speed as acceleration slope until reach origin regression speed, when leaving the origin falling edge signal instantaneous stop zero movement (if it sets the Z phase pulse, it starts to count Z phase signal after leaving the origin signal falling edge, then immediately stop motion when the counting reached), if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



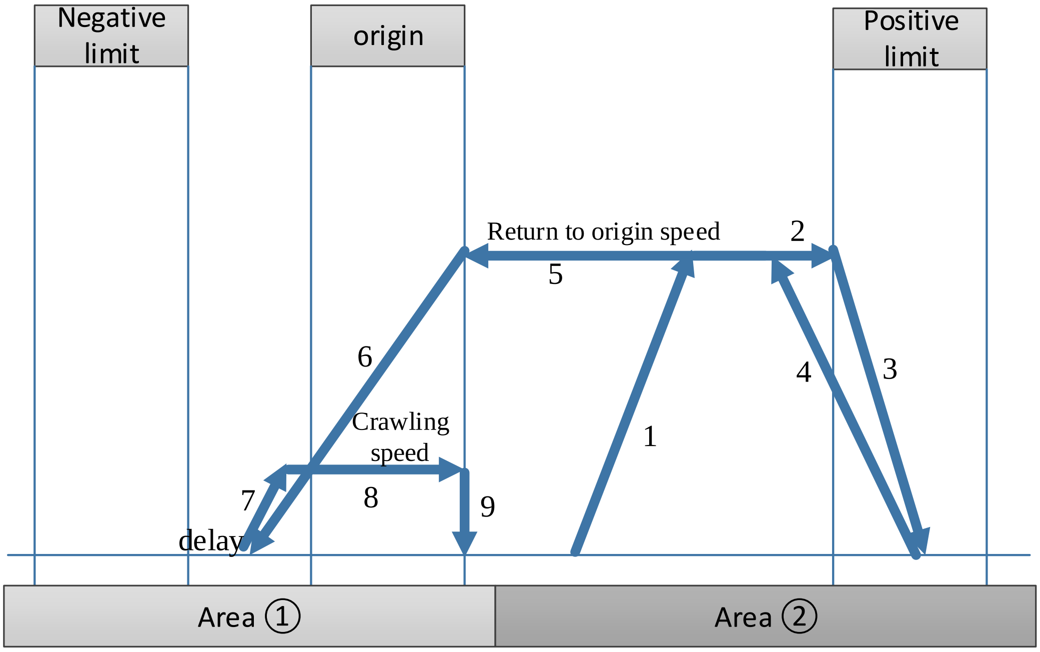
Note:

※1: In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the speed is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate in the same way.

※2: when it sets the servo Z phase pulse, Z phase pulse returning to origin capture function is effective, it will stop the mechanical origin regression in Z phase mode.

※3: If the stopping position falls beyond the negative limit position, it may lead to collision. Please try your best to avoid such situation. This can be done by reducing the set deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(2) workbench is between origin and positive limit, return to origin in forward direction



Return to origin in positive direction

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and the acceleration is accelerated to the origin regression speed, and then the regression speed of the origin moves toward the positive limit direction.

(2) When encountering the rising edge of the positive limit signal, slow down with the set deceleration slope until the deceleration to complete rest (frequency =0).

(3) Immediately reverse and start accelerating according to the specified acceleration slope until reaching origin regression speed, then the speed begins to recede towards the origin.

(4) when encountering the rising edge of origin signal, slow down with the set deceleration slope until the deceleration to complete rest (frequency =0).

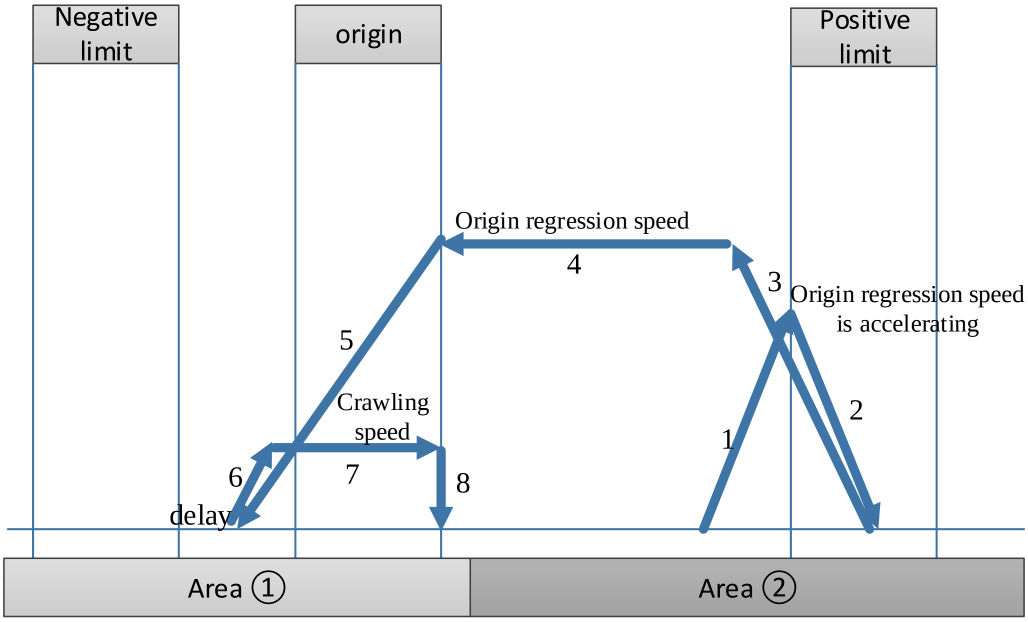
(5) delay (direction delay time in SFD), and then accelerate as the set acceleration slope, move forward until reaching the crawling speed, when leaving the mechanical origin falling edge signal instantaneous stop zero movement (if it sets the Z phase pulse, it starts to count Z phase signal after leaving the origin signal falling edge, then immediately stop motion when the counting reached).

(6) If "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

Special case 1:

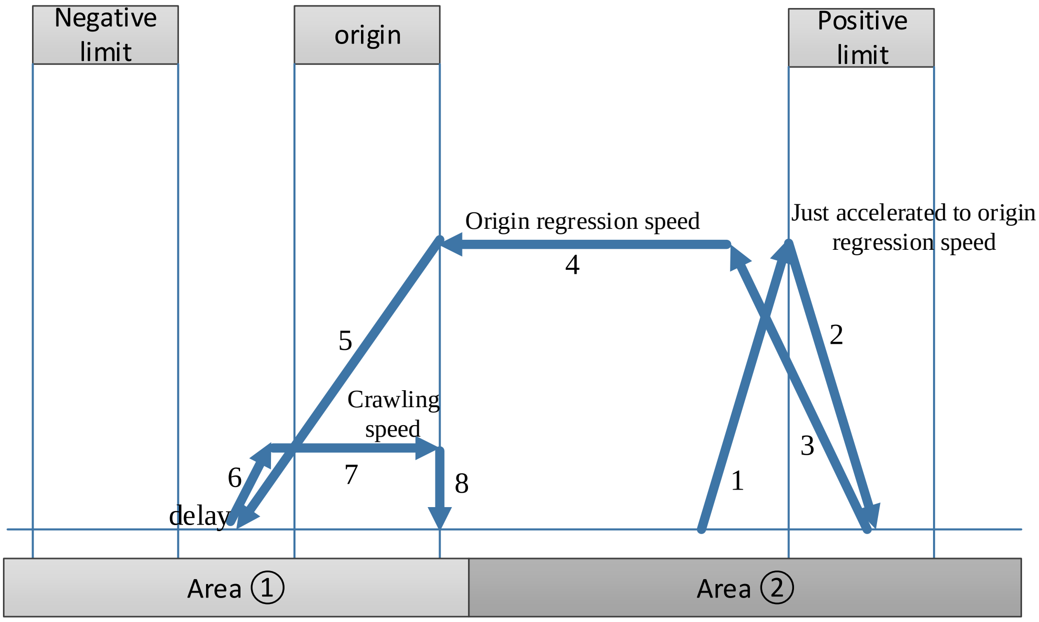
For the just started ZRN instruction, when accelerating in the positive limit direction and already reached the rising edge of the positive limit signal, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); then accelerate in reverse direction as acceleration slope until reach origin regression speed, then go back in origin direction, when meet the rising edge of origin signal, decelerate as deceleration slope until the

deceleration is completely still (frequency=0). Delay (direction delay time in SFD), low speed slow move in reverse direction with acceleration slope until reaching the origin regression speed,
When leaving the origin falling edge signal instantaneous stop pulse outputting (if it sets the Z phase pulse, it starts to count Z phase signal after leaving the origin signal falling edge, then immediately stop zero return motion when the counting reached), if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Special case 2:

For the just started ZRN instruction, when accelerating to origin regression speed in the positive limit direction and just reached the rising edge of the positive limit signal, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); then accelerate in reverse direction as acceleration slope until reach origin regression speed, then go back in origin direction, when meet the rising edge of origin signal, decelerate as deceleration slope until the deceleration is completely still (frequency=0). Delay (direction delay time in SFD), low speed slow move in reverse direction with acceleration slope until reaching the origin regression speed,
When leaving the origin falling edge signal instantaneous stop pulse outputting (if it sets the Z phase pulse, it starts to count Z phase signal after leaving the origin signal falling edge, then immediately stop zero return motion when the counting reached), if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touched the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

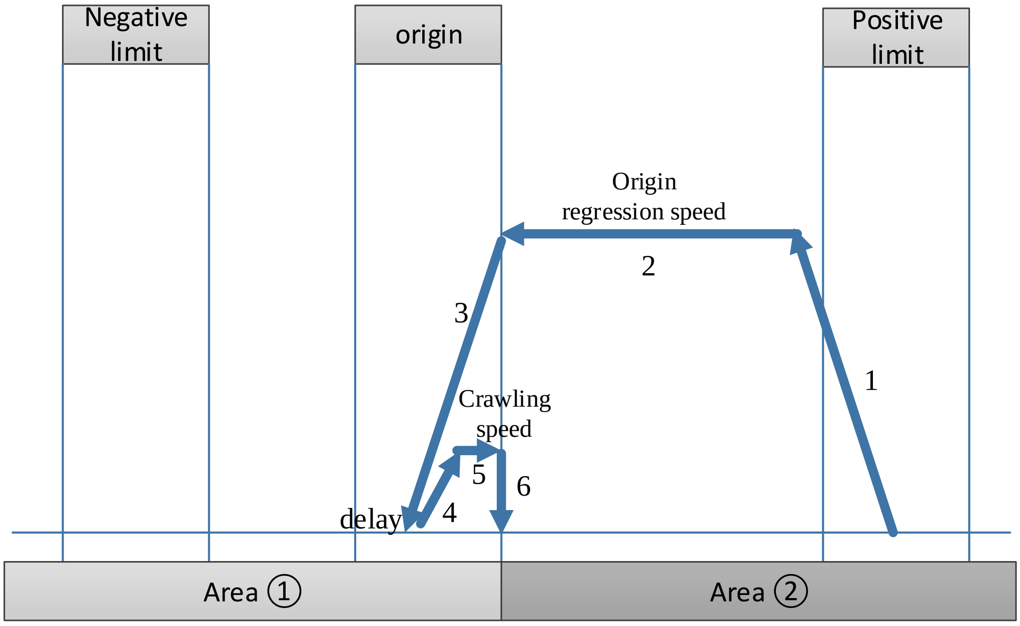
Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

※2: When the working table moves towards the positive limit with the speed of returning to the mechanical origin, it will start to decelerate according to the deceleration slope when it encounters the positive limit signal rising edge, and the deceleration stop position may fall on the positive limit or exceed the positive limit; Accidents that can occur when the positive limit is exceeded can be avoided by reducing the deceleration slope or widening the positive limit signal width. If the stopping position falls beyond the negative limit position, it may impact the machine. Please try your best to avoid such situation. This can be done by reducing the set deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(3) Execute origin returning when the workbench is in the positive limit

When the workbench is in the positive limit, return to the origin can only be performed by default in the reverse return to the origin mode, no matter whether the direction of return to the origin is set as forward return to the origin or reverse return to the origin, as shown in the figure below:



In the positive limit and execute origin returning

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and accelerated to the origin regression speed, and then the regression speed of the origin is withdrawn back to the direction of the origin.

(2) When encountering the rising edge of the origin signal, slow down with the deceleration slope until the deceleration is complete still (frequency =0).

(3) delay (the direction delay time in SFD), accelerate as the acceleration slope until reach the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(4) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touched the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

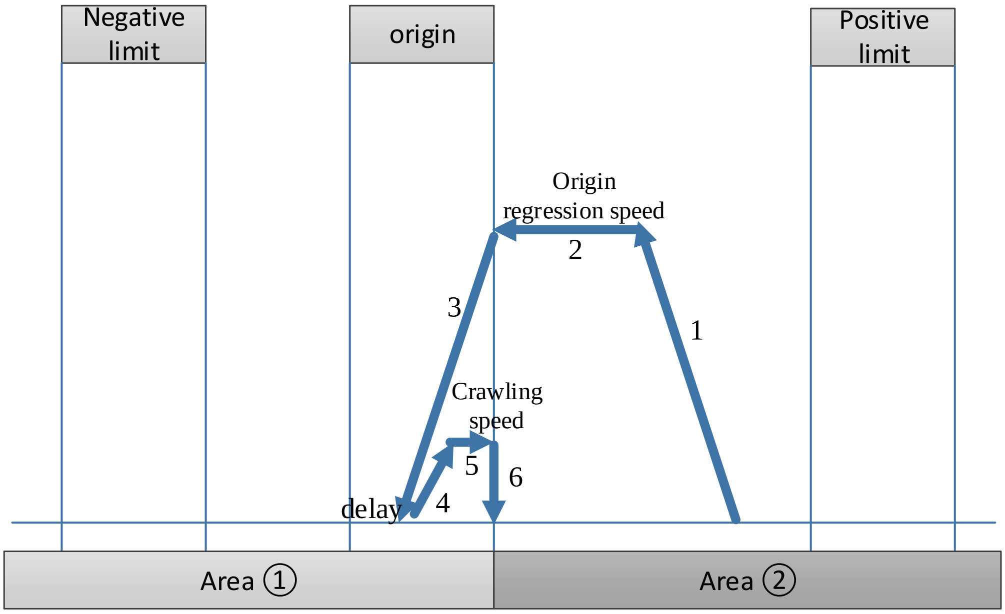
Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

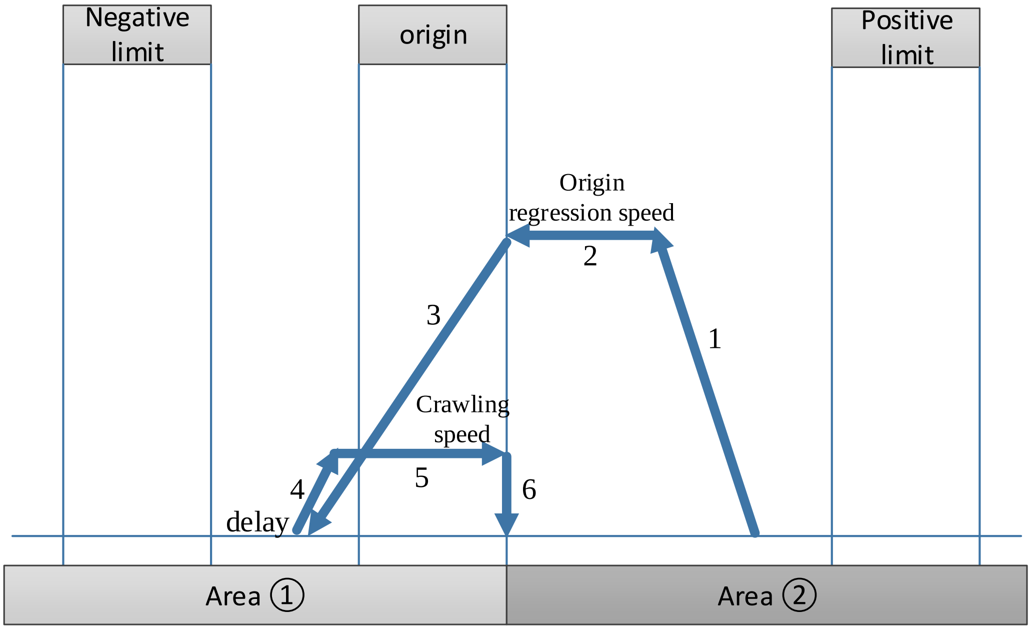
※2: If the stopping position falls beyond the negative limit position, it may impact the machine. Please try your best to avoid such situation. This can be done by reducing the set deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(4) execute the origin returning when workbench exceeds the positive limit
When the working table exceeds the positive limit, in order to prevent the occurrence of machine impact caused by positive return-to-origin, do not execute the return-to-origin. Please move the working table back to the negative(or positive) limit or between the positive limit and the negative limit manually, and then execute the mechanical return-to-origin instruction!
The limit switch width of the negative limit and positive limit can also be widened to avoid the occurrence of breaking off the positive limit and negative limit when the pulse deceleration stops.

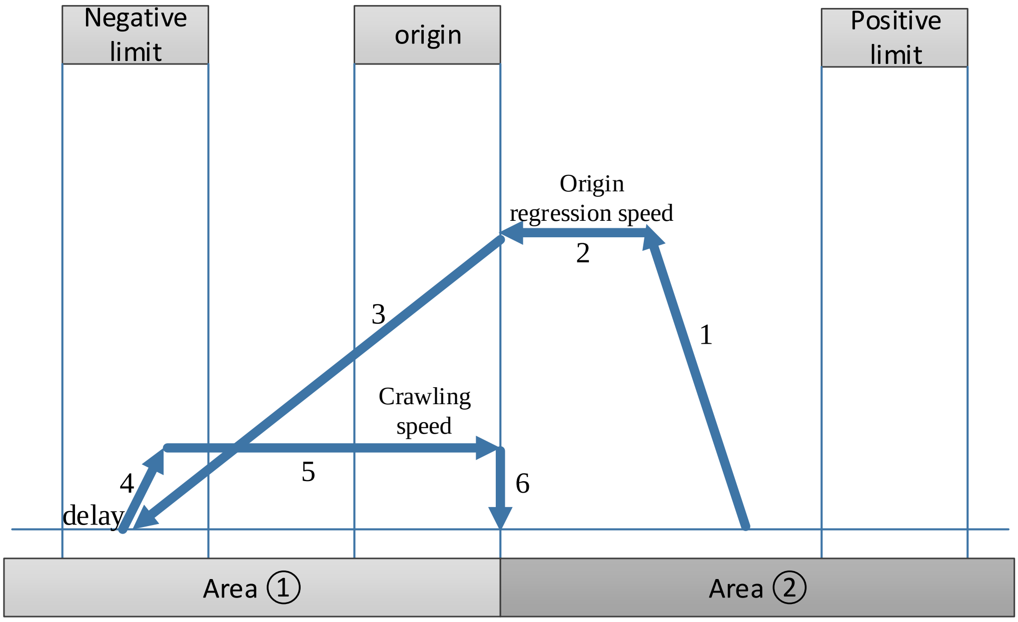
(5) When the table moves back toward the origin with the speed of mechanical return, it will start to slow down according to the set deceleration slope when it touches the rising edge of the mechanical origin. Due to the setting of different speed of mechanical return to the origin and deceleration slope, the final stop position of the table is relatively long, which shall be executed according to the following situations:



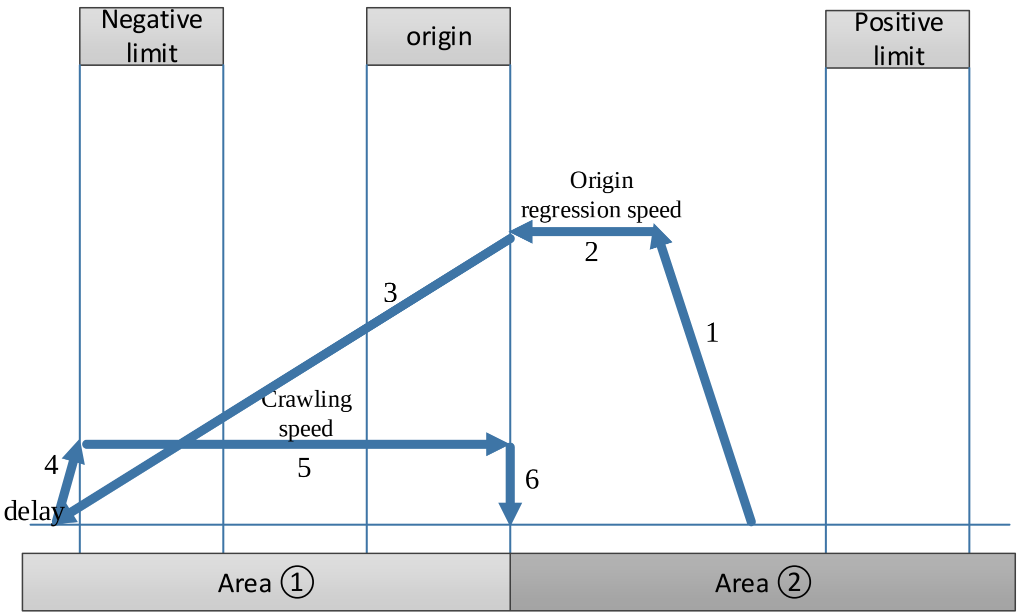
Stop position is on the mechanical origin



Stop position is between mechanical origin and negative limit



Stop position is on the negative limit



Stop position exceeded negative limit

Note:

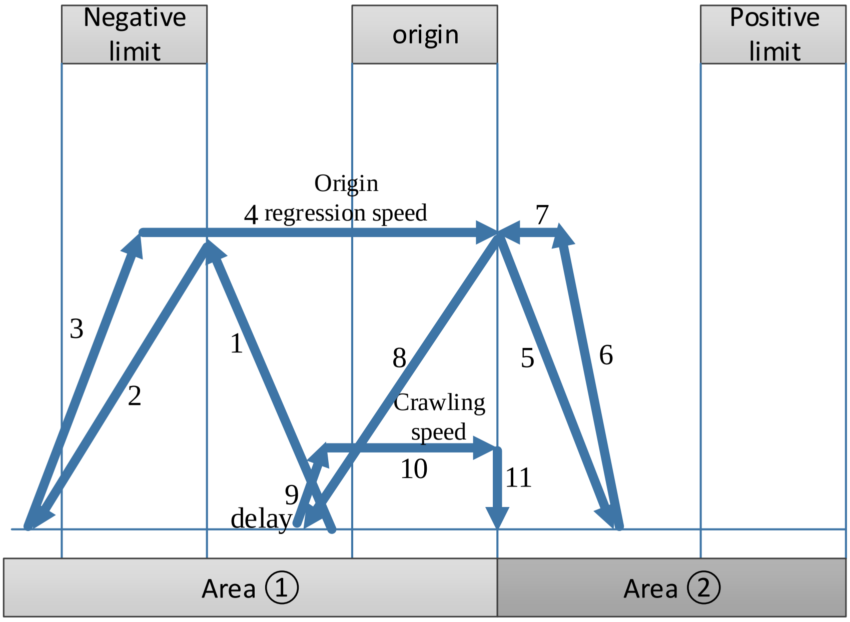
※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

※2: If the stopping position falls beyond the negative limit position, it may impact the machine. Please try your best to avoid such situation. This can be done by reducing the set deceleration slope or lengthening the length between the negative limit and the mechanical limit.

2. when the mechanical origin returning instruction ZRN starts, the working table is in area ①:

When the work table is located in the region, it can be divided into four situations: the work table is between the origin and the negative limit, the work table is at the mechanical origin, the work table is at the negative limit and the work table is beyond the negative limit position.

(1) execute origin regression when the work table is between the origin and negative limit



Execute origin regression in reverse direction

Action:

(1) When the origin regression action starts, the acceleration is carried out first by the set acceleration slope, and then go back in the negative limit direction with the origin regression speed after accelerating to the origin regression speed.

(2) when the work table encounters the rising edge of negative limit with the origin regression speed, it decelerates as the set deceleration slope until stop.

(3) accelerate as the set acceleration slope until reach the origin regression speed, move forward in mechanical origin direction.

(4) When the working table breaks away from the falling edge of the mechanical origin at the speed of mechanical return, it immediately begins to slow down according to the set deceleration slope, until the speed is 0.

(5) The working table immediately accelerates to the speed of returning to the mechanical origin according to the set acceleration slope, and moves back toward the mechanical origin.

(6) When encountering the rising edge of the origin signal, slow down with the deceleration slope until complete still (frequency =0).

(7) delay (the direction delay time in SFD), accelerate as the acceleration slope until reach the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(8) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

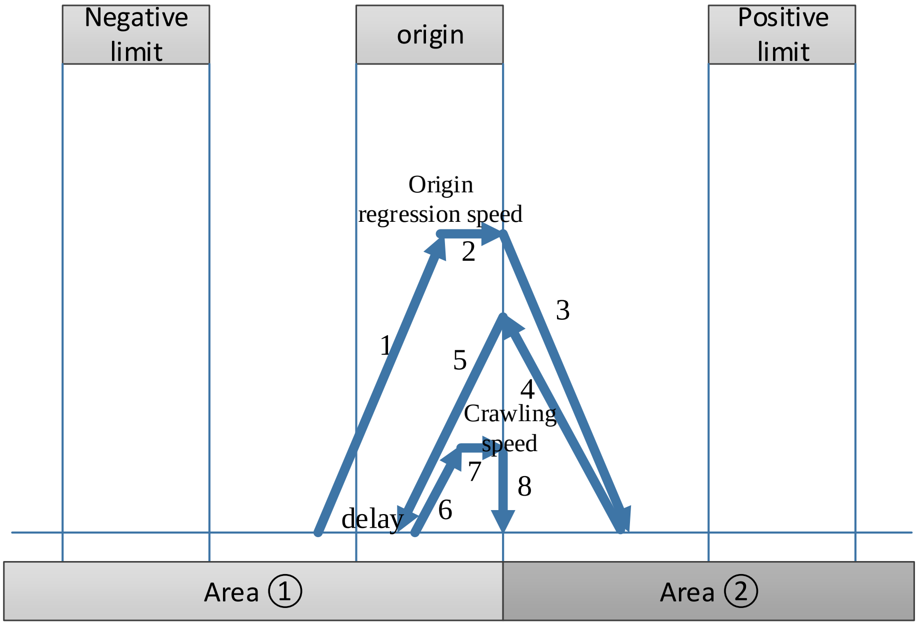
Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.
※2: If the stopping position falls beyond the negative limit position, it may impact the machine. Please try your best to avoid such situation. This can be done by reducing the set deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(2) execute the origin regression when the work table is between origin and negative limit



Return to origin in positive direction

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and then accelerated to the origin regression speed and moved forward in mechanical origin direction.
(2) When the working table breaks away from the falling edge of the mechanical origin at the speed of mechanical return, it immediately begins to slow down according to the set deceleration slope, until the speed is 0.
(3) accelerate as the set acceleration slope until reach the mechanical origin regression speed, go back in mechanical origin direction.
(4) when the work table encounters the rising edge of origin signal, it decelerates as the set deceleration slope until stop (frequency is 0). Delay (the direction delay time in SFD), accelerate as the acceleration slope until reach the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action

at once when the count value reached)
(5) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

Conclusion:

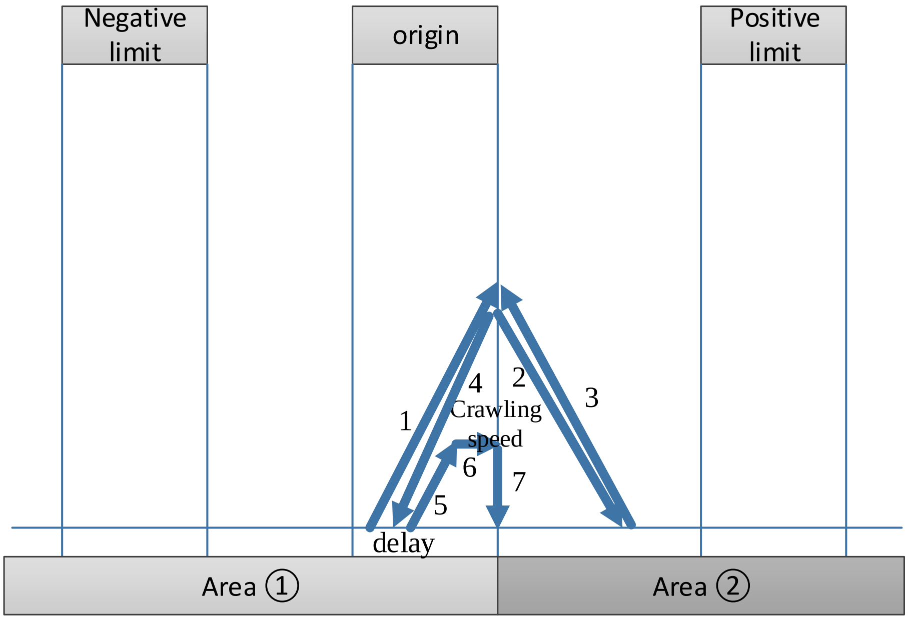
In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.
※2: When the origin returning action is started, the speed shall be accelerated by the set acceleration slope first. No matter whether the speed is accelerated to the speed of mechanical return to the origin, the work table will start to decelerate according to the set deceleration slope as soon as it touches the decline edge of mechanical origin signal.

(3) execute the origin returning when the work table is at the mechanical origin
When execute the reverse origin returning and the work table is at the mechanical origin, it will switch to positive origin returning inside, the details please refer to condtion (4).

(4) execute the positive origin regression when the work table is at the mechanical origin



Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and then accelerated to the origin regression speed and moved forward in falling edge of mechanical origin direction.
(2) Whether the table has been accelerated to the speed of the mechanical return to the origin according to the set acceleration slope, it will immediately begin to decelerate according to the set deceleration slope at the descent edge of the mechanical origin until the speed is 0.
(3) The working table immediately starts to accelerate to the speed of returning to the mechanical origin according to the set acceleration slope, and moves back toward the mechanical origin.
(4) Whether the table has been accelerated to the speed of the mechanical return to the origin according to the set acceleration slope, it will immediately begin to decelerate according to the set deceleration slope at the rising edge of the mechanical origin until the speed is 0. Delay (the direction delay time in SFD), accelerate as the acceleration slope until reach the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)
(5) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

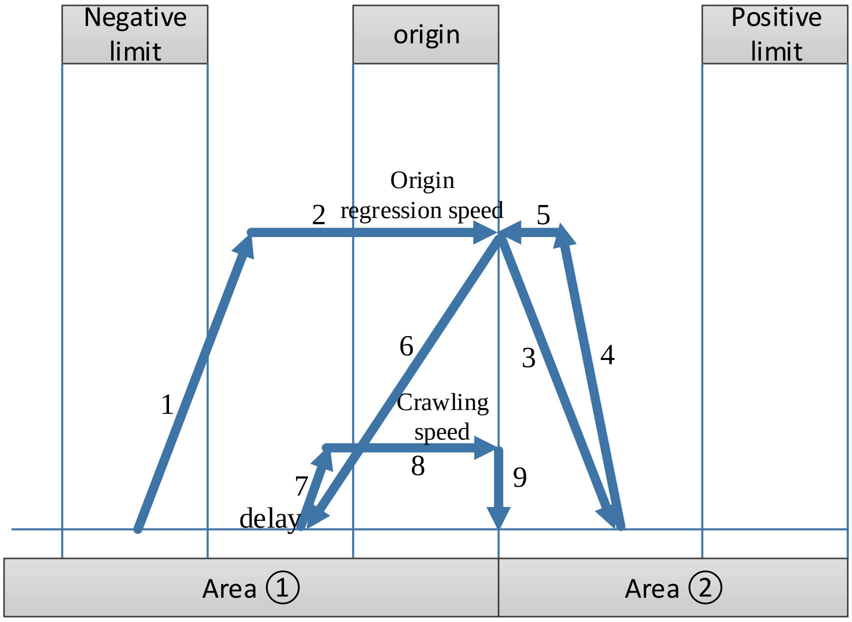
Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.
※2: When the origin returning action is started, the speed shall be accelerated by the set acceleration slope first. No matter whether the speed is accelerated to the speed of mechanical return to the origin, the work table will start to decelerate according to the set deceleration slope as soon as it touches the decline edge of mechanical origin signal.

(5) execute the origin returning when the working table is at the negative limit
When the working table is at the negative limit, whatever the origin returning direction is set to positive or negative, it must execute as defaulted positive direction, shown as below:



Execute origin regression at the negative limit

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and then accelerated to the origin regression speed and moved forward in origin direction.

(2) When encountering the descent edge of the origin signal, slow down by the deceleration slope until complete rest (frequency =0).

(3) The table starts to accelerate immediately according to the set acceleration slope. Whether it has accelerated to the speed of mechanical return to the origin or not, as long as the table touches the rising edge of mechanical origin signal, it will immediately start to decelerate according to the set deceleration slope.

(4) when the work table decelerated to stop, it started to delay (the direction delay time in SFD), then accelerated as the acceleration slope until reaching the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(5) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

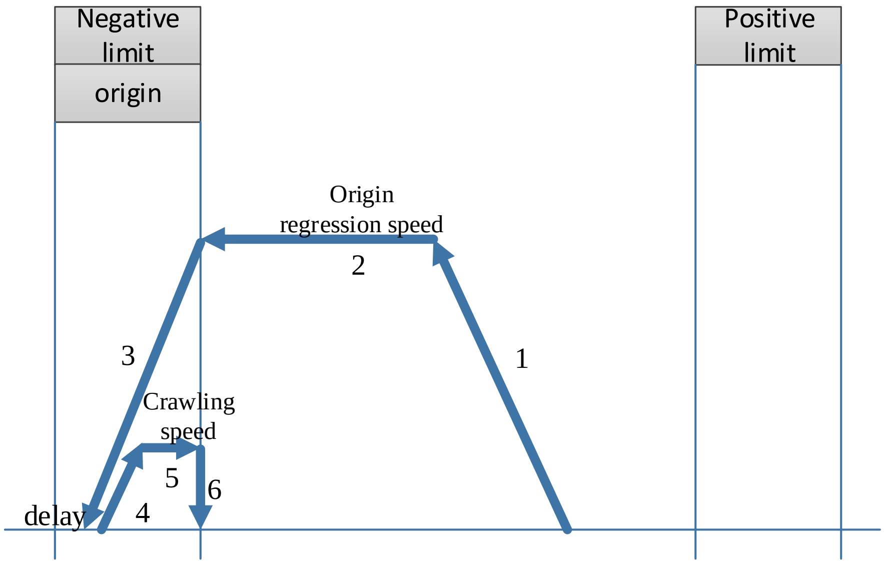
(6) execute origin returning when the work table exceeded negative limit
When the working table exceeds the negative limit, in order to prevent the occurrence of machine impact caused by reverse-returning to the origin, please do not go back to the origin. Please move the working table back to the negative or positive limit or between them by manual and then carry out the execution of the mechanical returning to the origin instruction!
The limit switch width of the negative limit and positive limit can also be widened to avoid the occurrence of breaking off the positive limit and negative limit when the pulse deceleration stops.

3. When in consideration of equipment cost or mechanical structure, negative limit switches and mechanical origin switches may need to be used with a proximity switch or travel switch.

First, we set the mechanical origin and negative limit switch in system parameter block as the same input point. When executing the ZRN mechanical return instruction, this input point is used as the mechanical origin. This input point is used as a negative limit when using pulse output commands such as PLSR, PLSF, DRVI, and DRVA.

In view of the position of the work table returning to the mechanical origin, the following will be explained according to the following situations: the work table is between negative limit and positive limit, the work table is in negative limit, the work table is in positive limit, the work table exceeds positive limit position and the work table exceeds negative limit position.

(1) execute reverse origin returning when the work table is between negative limit and positive limit



Return to origin in reverse direction

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and accelerated to the origin regression speed, and then went back toward the

mechanical origin direction.

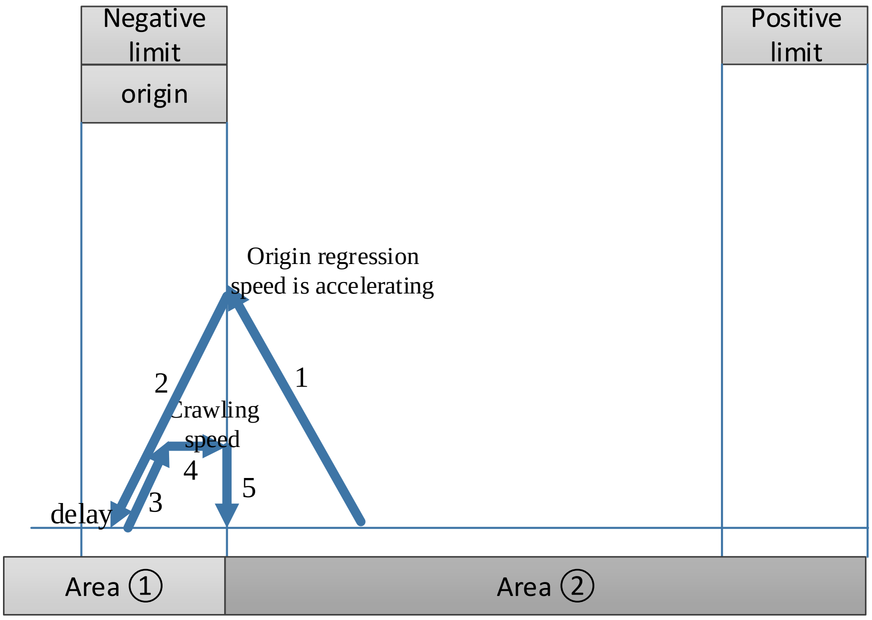
(2) When encountering the rising edge of the origin signal, slow down by the deceleration slope until complete rest (frequency =0).

(3) delay (the direction delay time in SFD), then accelerated as the acceleration slope until reaching the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(4) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

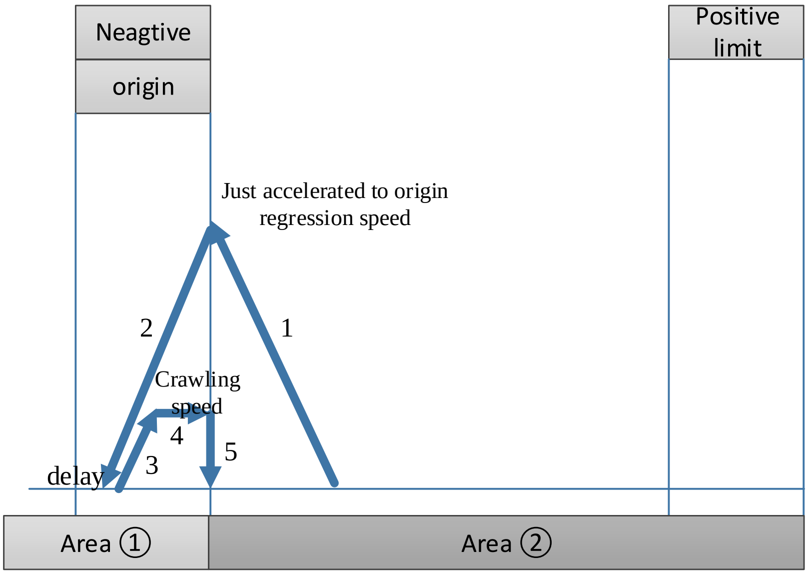
Special case 1:

When the acceleration of the just started ZRN instruction has reached the rising edge of the mechanical origin signal, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); delay (the direction delay time in SFD), then reverse move at slow speed as acceleration slope until reach origin regression speed, when at the moment of leaving the origin signal falling edge, if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Special case 2:

In the acceleration process of the just started ZRN instruction, when it just accelerated to origin regression speed, it reached the rising edge of the mechanical origin signal, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); delay (the direction delay time in SFD), then reverse move at slow speed as acceleration slope until reach origin regression speed, stop returning action at the moment of leaving the origin signal falling edge (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached), if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Conclusion:

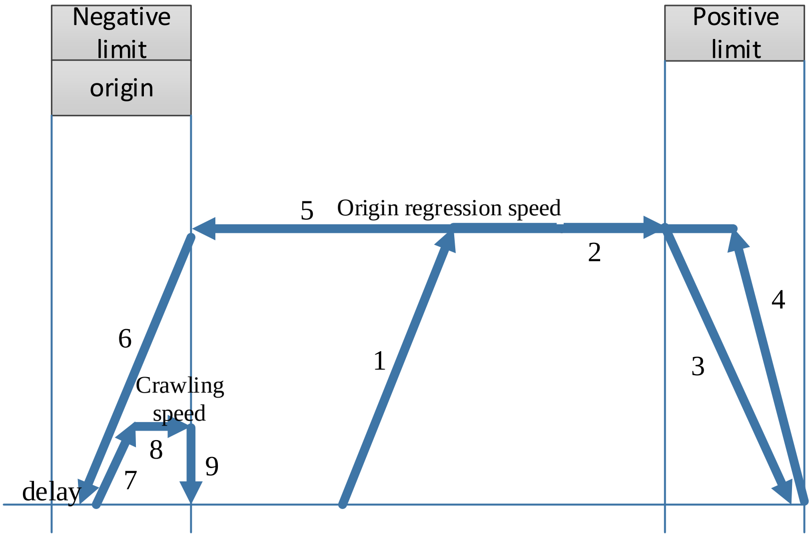
In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

※2: If the stopping position falls beyond the negative limit position, it may lead to machine impact. Please try your best to avoid such situation. This can be done by reducing the stated deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(2) execute origin returning in forward direction when the work table is between negative limit and positive limit



Return to origin in positive direction

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and accelerated to the origin regression speed, and then went forward toward the positive direction of positive limit.

(2) When encountering the rising edge of the origin signal, slow down by the deceleration slope until complete rest (frequency =0).

(3) Immediately reverse and start accelerating according to the specified acceleration slope until reaching the origin regression speed and begins to recede towards the origin.

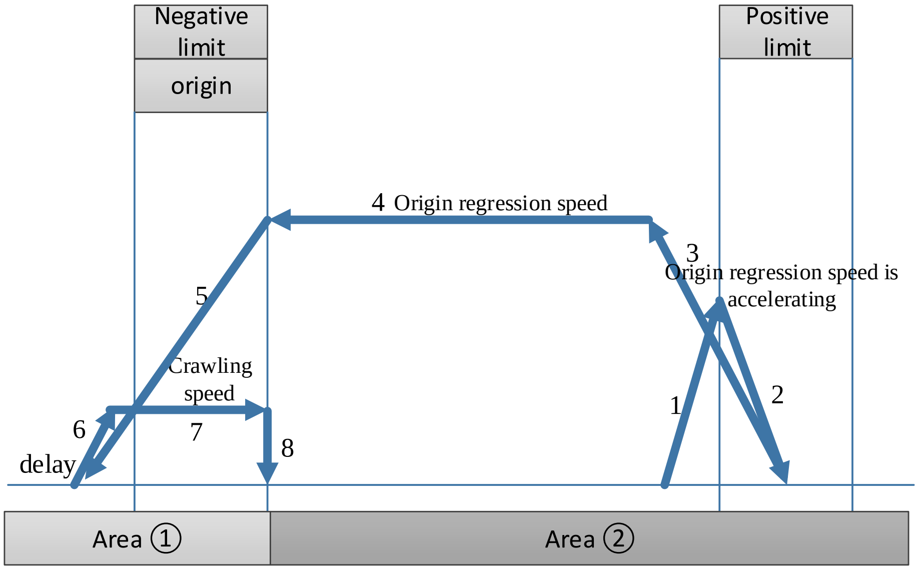
(4) When encountering the rising edge of the origin signal, slow down by the deceleration slope until complete rest (frequency =0).

(5) delay (the direction delay time in SFD), then accelerated as the acceleration slope until reaching the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(6) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

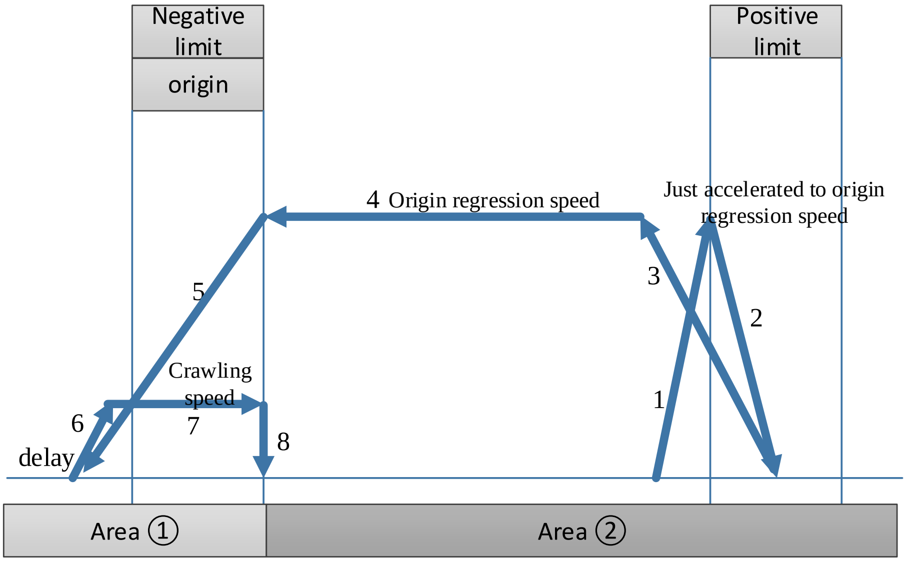
Special case 1:

For the just started ZRN instruction, when it has already reached the rising edge of the positive limit signal in the process of accelerating towards positive limit, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); immediately reverse and start accelerating according to the set acceleration slope until reaching the origin regression speed, then start go back, when encountering the rising edge of the origin signal, slow down by the deceleration slope until complete stop (frequency =0); delay(direction delay time in SFD), then reverse move at slow speed as acceleration slope until reach origin regression speed, at the moment of leaving the origin signal falling edge, stop pulse outputting at once(if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached). If "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Special case 2:

For the just started ZRN instruction, when it just reached the rising edge of the positive limit signal in the process of accelerating towards positive limit and just accelerated to origin returning speed, the deceleration slope is used as the deceleration action until the deceleration is completely still (frequency =0); immediately reverse and start accelerating according to the set acceleration slope until reaching the origin regression speed, then start go back, when encountering the rising edge of the origin signal, slow down by the deceleration slope until complete stop (frequency =0); delay(direction delay time in SFD), then reverse move at slow speed as acceleration slope until reach origin regression speed, at the moment of leaving the origin signal falling edge, stop pulse outputting at once(if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached). If "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.



Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note:

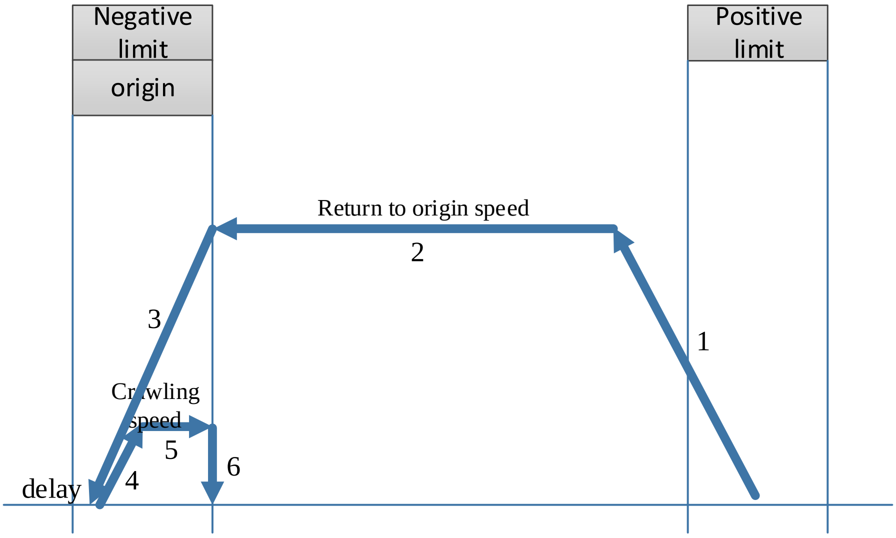
※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

※2: When the working table moves towards the positive limit with the speed of returning to the mechanical origin, it will start to decelerate according to the deceleration slope when it encounters the positive limit signal rising edge, and the deceleration stop position may fall on the positive limit or exceed the positive limit; Accidents that can occur when the positive limit is exceeded, which can be avoided by reducing the deceleration slope or widening the positive limit signal width.

※3: If the stopping position falls beyond the negative limit position, it may lead to machine impact. Please try your best to avoid such situation. This can be done by reducing the stated deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(3) execute the origin returning when the work table is in the positive limit

When the work station is in the positive limit, return to the origin can only be performed by default in the reverse return to the origin mode, no matter whether the direction of return to the origin is set as forward return to the origin or reverse return to the origin, as shown in the figure below:



Return to origin in the positive limit

Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, and accelerated to the origin regression speed, and then the regression speed of the origin is withdrawn back towards the direction of the origin.

(2) When encountering the rising edge of the origin signal, slow down by the deceleration slope until complete rest (frequency =0).

(3) delay (the direction delay time in SFD), then accelerated as the acceleration slope until reaching the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(4) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

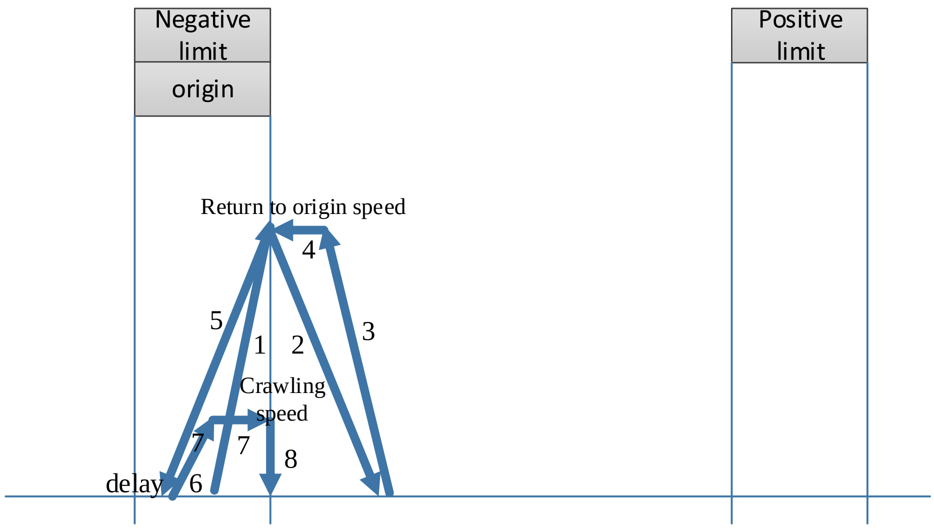
Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

※2: If the stopping position falls beyond the negative limit position, it may lead to machine impact. Please try your best to avoid such situation. This can be done by reducing the stated deceleration slope or lengthening the length between the negative limit and the mechanical limit.

(4) execute origin returning when the work table is at the mechanical origin

When the worktable is at the mechanical origin, the worktable will return to the origin in positive direction no matter the setting direction is positive or negative, as shown in the figure below:



Action:

(1) When the origin regression action starts, the acceleration is carried out first with the set acceleration slope, after accelerated to the origin regression speed, move forward towards mechanical origin falling edge direction with origin returning speed.

(2) Whether or not the work table has been accelerated to the speed of the mechanical return to the origin according to the set acceleration slope, it will immediately begin to decelerate according to the set deceleration slope when leaving the descent edge of the mechanical origin until the speed acceleration is 0.

(3) The working table immediately starts to accelerate to the speed of returning to the mechanical origin according to the set acceleration slope, and moves back toward the mechanical origin.

(4) whatever the working table has been accelerated to the speed of mechanical return to the origin according to the set acceleration slope, when encountering the rising edge of the origin signal, the deceleration slope is used as the deceleration action until complete rest (frequency =0). Delay (the direction delay time in SFD), then accelerated as the acceleration slope until reaching the crawling speed, then move forward, stop zero returning action at the moment of leaving the falling edge of origin signal (if it set Z phase pulse, it starts counting the Z phase after leaving the falling edge of the origin signal, stop zero returning action at once when the count value reached)

(4) if "zeroing clear CLR signal" is set, it will output the clear signal immediately and delay (CLR signal delay time in SFD can be used to clear the Error Counter of the servo motor), At last, copy the mechanical origin position value to the current position and the zeroing action will be completed.

Conclusion:

In the above cases, as long as the rising edge of the origin signal is touched (the right edge of the origin), whether the acceleration has reached the speed of returning to the mechanical origin, is in the process of acceleration, or just accelerated to the speed of returning to the origin, the deceleration starts immediately according to the deceleration slope, until the deceleration is 0. Similarly, when the working table described below touches the rising edge of the negative limit (the right edge of the negative limit) and the rising edge of the positive limit (the left edge of the positive limit), please operate them in the same way.

Note:

※1: When the servo Z phase pulse is set, the Z phase pulse back to the origin capture function is effective, and the mechanical return to the origin will be stopped according to the Z phase mode.

※2: When the return operation of the origin is started, it will be accelerated by the set acceleration slope first. No matter the speed is accelerated to the speed of mechanical return to the origin, the work table will start to decelerate according to the set deceleration slope as soon as it touches the decline edge of mechanical origin signal.

※3: When the table starts to accelerate towards the mechanical origin signal, whether it has accelerated to the speed of mechanical return to the origin or not, as long as the table touches the rising edge of the mechanical origin signal, it will immediately start to decelerate according to the set deceleration slope.

(5) execute the origin returning when the work table exceeds the positive limit

When the working table exceeds the positive limit, in order to prevent the occurrence of machine impact caused by positive return-to-origin, do not execute the return-to-origin. Please move the working table back to the negative(positive) limit or between the positive limit and the positive limit manually, and then execute the mechanical return-to-origin instruction!

The limit switch width of the negative limit and positive limit can also be widened to avoid the occurrence of breaking off the positive limit and negative limit when the pulse deceleration stops.

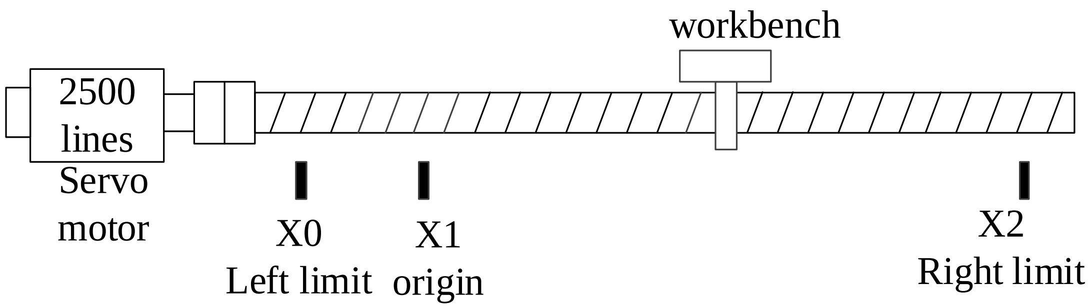
(6) execute the origin returning when the work table exceeds the negative limit

When the working table exceeds the negative limit, in order to prevent the occurrence of machine impact caused by positive return-to-origin, do not execute the return-to-origin. Please move the working table back to the negative(positive) limit or between the positive limit and the positive limit manually, and then execute the mechanical return-to-origin instruction!

The limit switch width of the negative limit and positive limit can also be widened to avoid the occurrence of breaking off the positive limit and negative limit when the pulse deceleration stops.

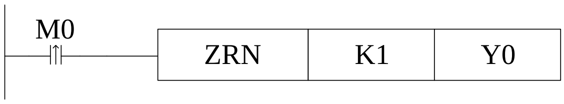
Example 1

As shown in the diagram below, one servo driver (electronic gear ratio is 1:1 by default) controls one servo motor (encoder 2500 lines), which is connected to the ball screw, whose pitch is 10mm. the ball screw drives workbench which can move right and left. Now the workbench needs to return to the origin, left limit switch connects the PLC input X0 (normally open), the right limit switch connects the PLC input X2 (normally open), the origin position switch connects the PLC input X1 (normally open), the origin regression speed VH is 10000hz, direction delay time in SFD is 100 ms, crawling speed VC is 100hz, not count the Z phase signal, pulse output port is Y0, direction terminal is Y2, mechanical origin position is set to 0, accelerate slope is 1000hz per 100 ms, The deceleration slope is 1000Hz per 150ms.

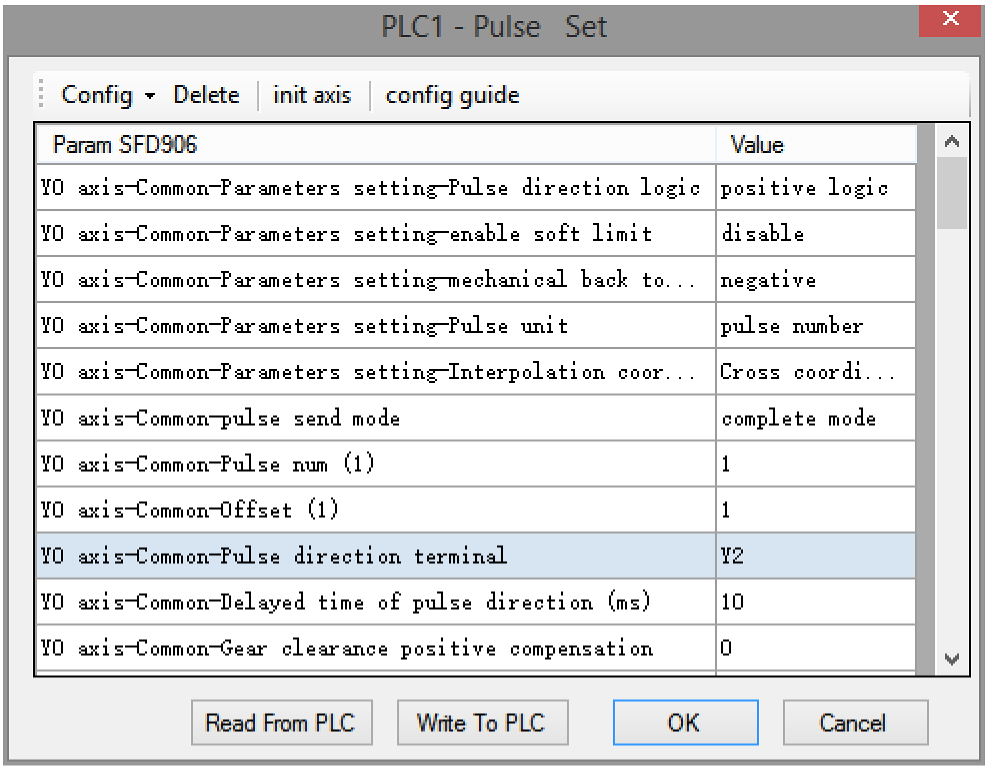


Structure diagram

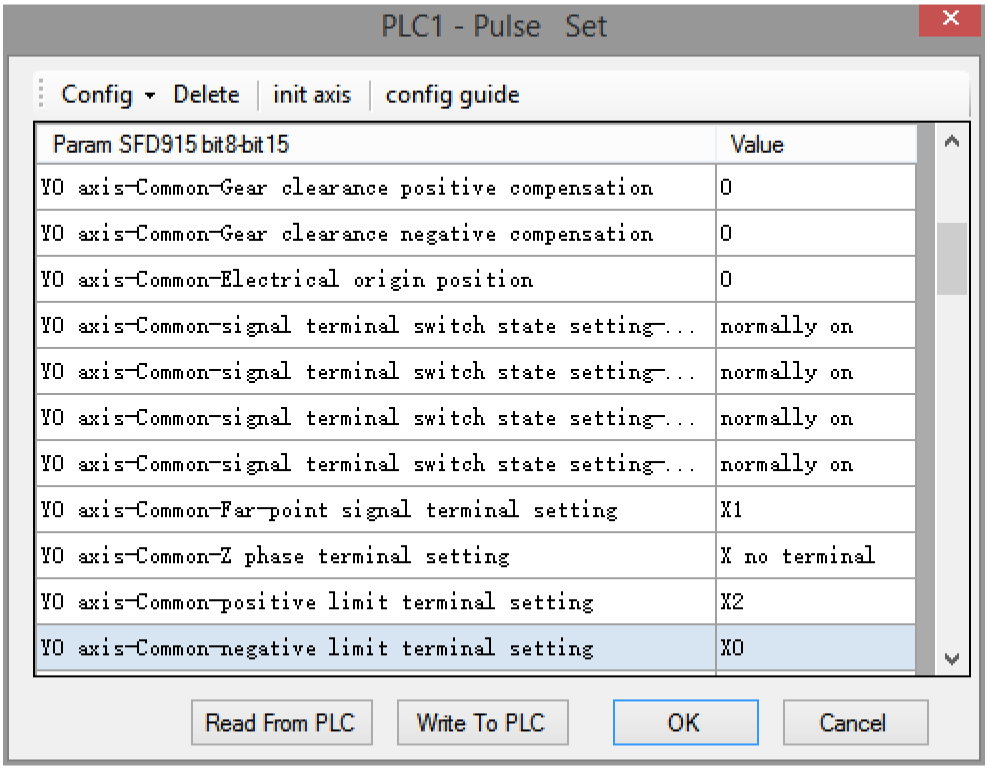
- **The instruction to return to the mechanical origin**

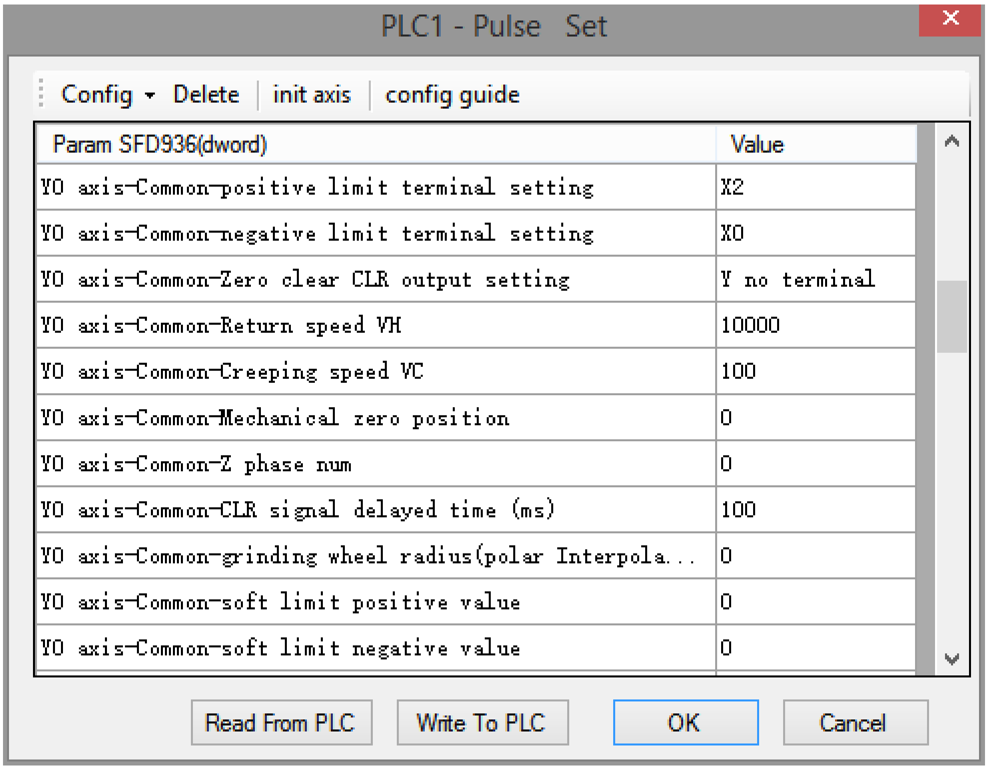


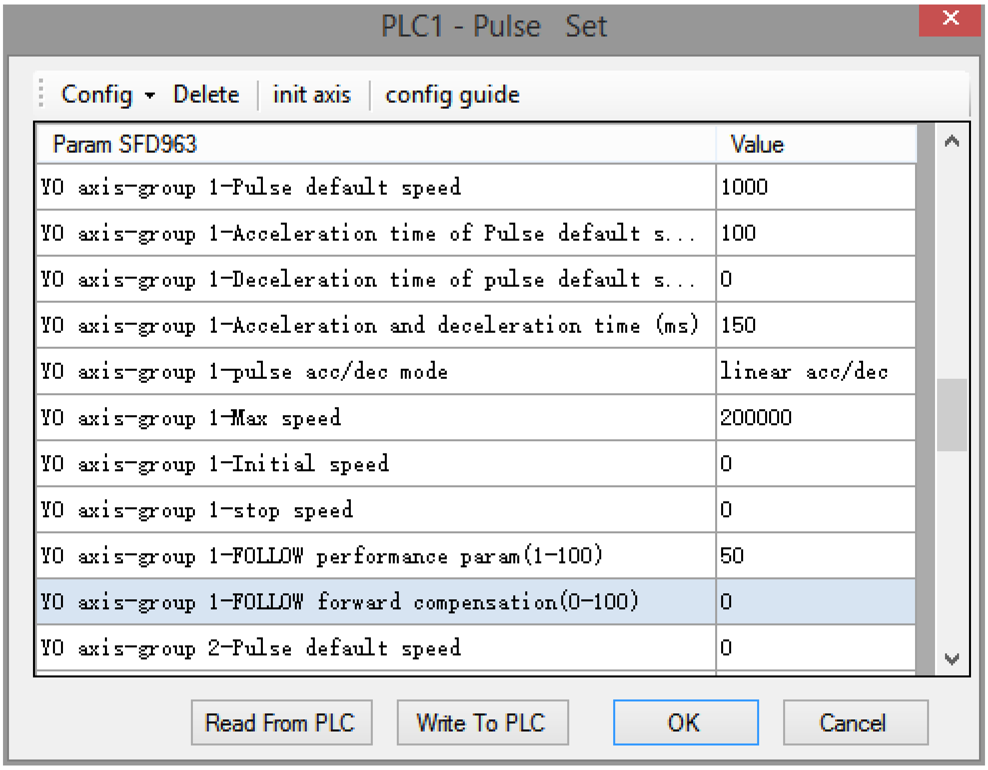
- **System parameter configurations**



Param SFD906	Value
Y0 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y0 axis-Common-Parameters setting-enable soft limit	disable
Y0 axis-Common-Parameters setting-mechanical back to...	negative
Y0 axis-Common-Parameters setting-Pulse unit	pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y0 axis-Common-pulse send mode	complete mode
Y0 axis-Common-Pulse num (1)	1
Y0 axis-Common-Offset (1)	1
Y0 axis-Common-Pulse direction terminal	Y2
Y0 axis-Common-Delayed time of pulse direction (ms)	10
Y0 axis-Common-Gear clearance positive compensation	0


PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD915 bit8-bit15
Value
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X1
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X2
Y0 axis-Common-negative limit terminal setting
X0
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD936(dword)
Value
Y0 axis-Common-positive limit terminal setting
X2
Y0 axis-Common-negative limit terminal setting
X0
Y0 axis-Common-Zero clear CLR output setting
Y no terminal
Y0 axis-Common-Return speed VH
10000
Y0 axis-Common-Creeping speed VC
100
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
100
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Read From PLC
Write To PLC
OK
Cancel



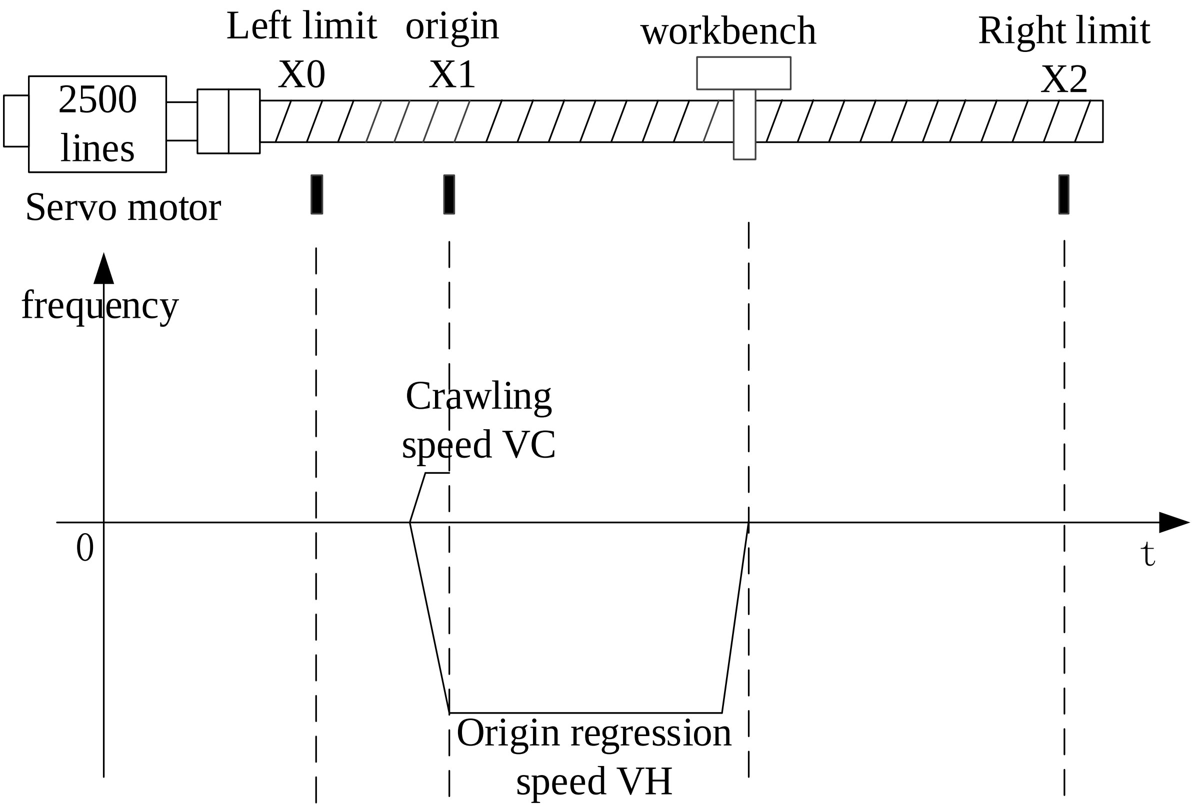
PLC1 - Pulse Set

Config | Delete | init axis | config guide

Param SFD963	Value
Y0 axis-group 1-Pulse default speed	1000
Y0 axis-group 1-Acceleration time of Pulse default s...	100
Y0 axis-group 1-Deceleration time of pulse default s...	0
Y0 axis-group 1-Acceleration and deceleration time (ms)	150
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0
Y0 axis-group 2-Pulse default speed	0

Read From PLC | Write To PLC | OK | Cancel

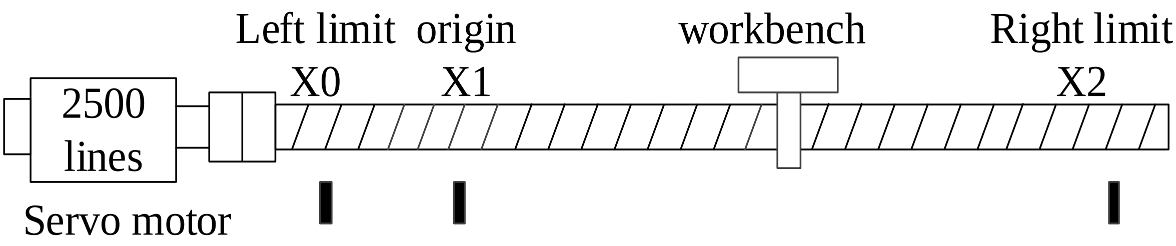
➢ **Mechanical origin regression motion diagram**



- in the moment of leaving the falling edge of origin signal X1 with crawling speed, the mechanical origin regression end immediately.
- if origin regression speed, acceleration/deceleration time, and left limit origin position settings are unreasonable, in the deceleration process of touching the origin signal it has already touched left limit, although there are solutions for such special cases inside the software, we try our best to avoid such special cases in the design of the solution. Special circumstances are not explained here.
- Y2 pulse direction terminal always keeps OFF when the workbench is moving from right to left, Y2 is ON when reverse moving with crawling speed until stop.

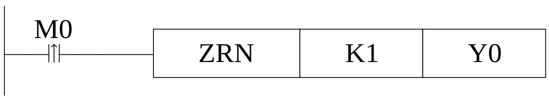
Example 2

As shown in the diagram below, one servo driver (electronic gear ratio is 1:1 by default) controls one servo motor (encoder 2500 lines), which is connected to the ball screw, whose pitch is 10mm. the ball screw drives workbench which can move right and left. Now the workbench needs to return to the origin, left limit switch connects the PLC input X0 (normally open), the right limit switch connects the PLC input X2 (normally open), the origin position switch connects the PLC input X1 (normally open), the origin regression speed VH is 10000hz, direction delay time in SFD is 100 ms, crawling speed VC is 100hz, count the Z phase signal when reverse leaving the origin signal(connects to PLC input X4), Z phase number is set to 6, pulse output port is Y0, direction terminal is Y2, mechanical origin position is set to 0, accelerate slope is 1000hz per 100 ms, The deceleration slope is 1000Hz per 150ms.

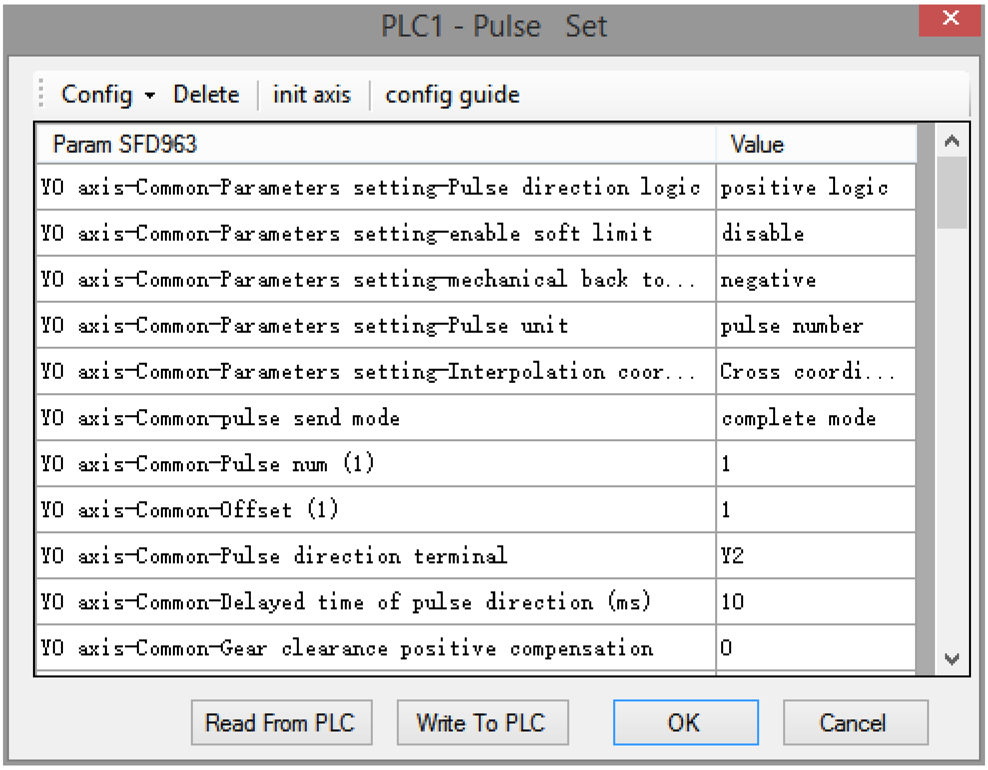


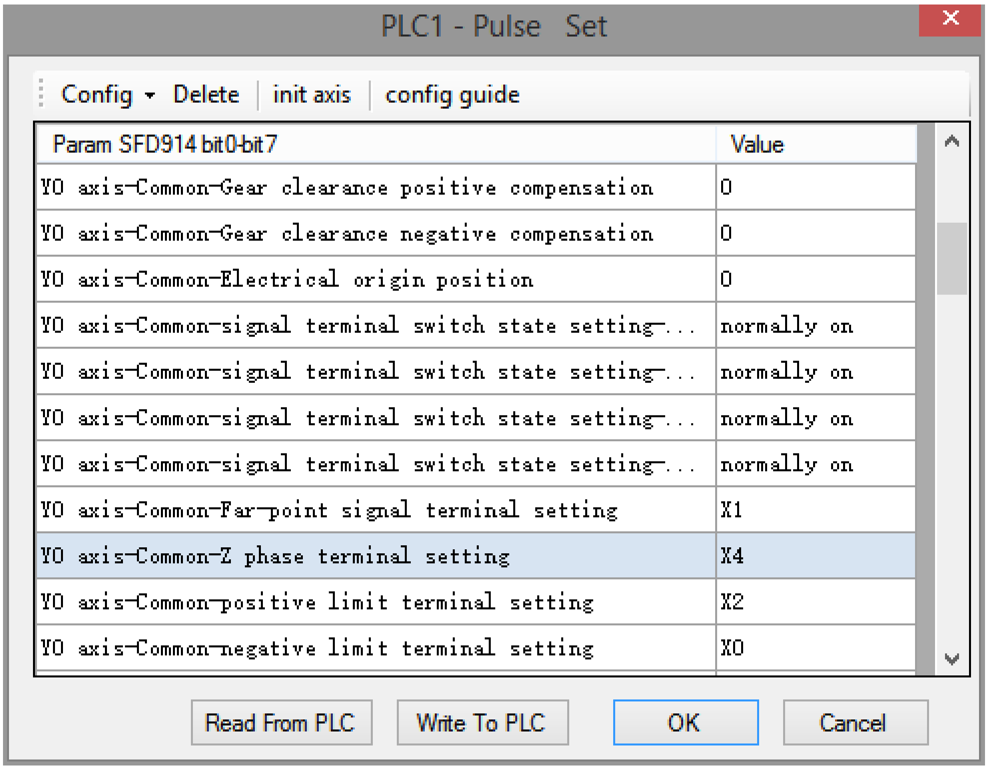
Structure diagram

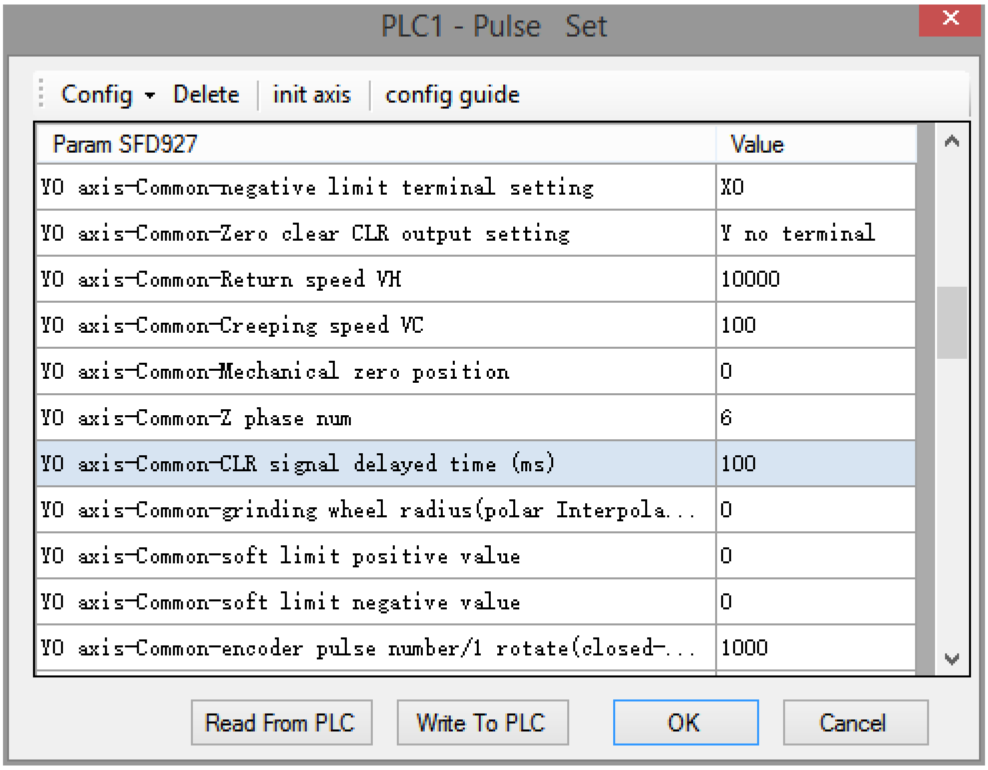
- **The instruction of origin regression**



- **System parameter configurations**


PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD963
Value
Y0 axis-Common-Parameters setting-Pulse direction logic
positive logic
Y0 axis-Common-Parameters setting-enable soft limit
disable
Y0 axis-Common-Parameters setting-mechanical back to...
negative
Y0 axis-Common-Parameters setting-Pulse unit
pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...
Cross coordi...
Y0 axis-Common-pulse send mode
complete mode
Y0 axis-Common-Pulse num (1)
1
Y0 axis-Common-Offset (1)
1
Y0 axis-Common-Pulse direction terminal
Y2
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD914 bit0-bit7
Value
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X1
Y0 axis-Common-Z phase terminal setting
X4
Y0 axis-Common-positive limit terminal setting
X2
Y0 axis-Common-negative limit terminal setting
X0
Read From PLC
Write To PLC
OK
Cancel

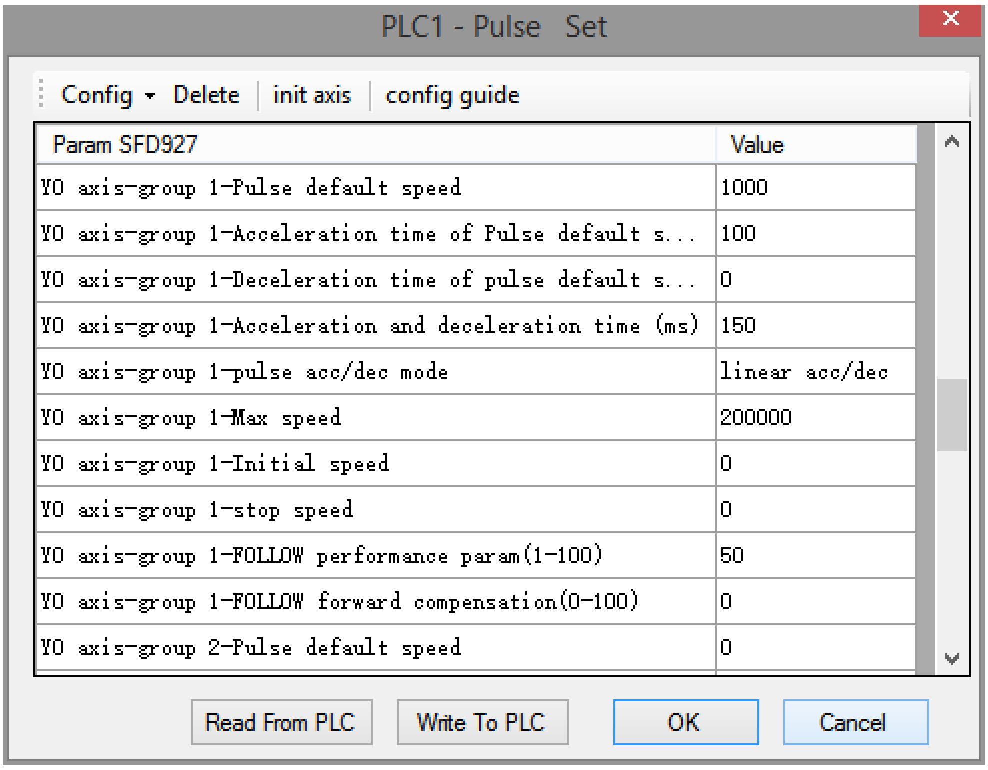


PLC1 - Pulse Set

Config Delete init axis config guide

Param SFD927	Value
Y0 axis-Common-negative limit terminal setting	X0
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	10000
Y0 axis-Common-Creeping speed VC	100
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	6
Y0 axis-Common-CLR signal delayed time (ms)	100
Y0 axis-Common-grinding wheel radius(polar Interpola...	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...	1000

Read From PLC Write To PLC OK Cancel



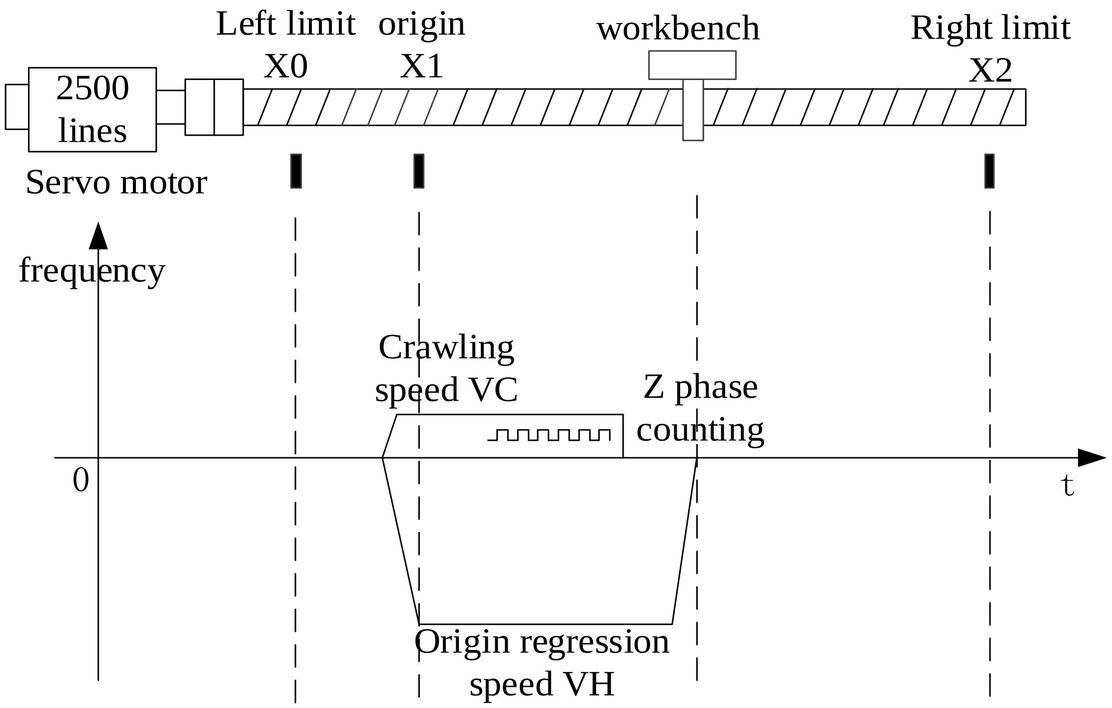
PLC1 - Pulse Set

Config Delete init axis config guide

Param SFD927	Value
Y0 axis-group 1-Pulse default speed	1000
Y0 axis-group 1-Acceleration time of Pulse default s...	100
Y0 axis-group 1-Deceleration time of pulse default s...	0
Y0 axis-group 1-Acceleration and deceleration time (ms)	150
Y0 axis-group 1-pulse acc/dec mode	linear acc/dec
Y0 axis-group 1-Max speed	200000
Y0 axis-group 1-Initial speed	0
Y0 axis-group 1-stop speed	0
Y0 axis-group 1-FOLLOW performance param(1-100)	50
Y0 axis-group 1-FOLLOW forward compensation(0-100)	0
Y0 axis-group 2-Pulse default speed	0

Read From PLC Write To PLC OK Cancel

- **Mechanical origin regression motion diagram**



- When leaving origin signal X1 with crawling speed, count Z phase at once, pulse stop at once when the Z phase counting value reached, the mechanical origin regression end immediately.
- if origin regression speed, acceleration/deceleration time, and left limit origin position settings are unreasonable, in the deceleration process of touching the origin signal it has already touched left limit, although there are solutions for such special cases inside the software, we try our best to avoid such special cases in the design of the solution. Special circumstances are not explained here.
- Y2 pulse direction terminal always keeps OFF when the workbench is moving from right to left, Y2 is ON when reverse moving with crawling speed until stop.

1-2-7. Pulse stop [STOP]

1. deceleration stop pulse outputting

Pulse stop [STOP]			
16-bit instruction	STOP	32-bit instruction	-
Execution condition	Rising edge /falling edge of the coil	Suitable model	XD, XL (except XD1, XL1)
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S	The terminal to stop the pulse outputting	bit
D	Pulse stop mode (0: stop slowly, 1: scram)	16-bit, word

3. Suitable soft component

word

operand	System								constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●	●	●	●	●	●	●	●			

bit

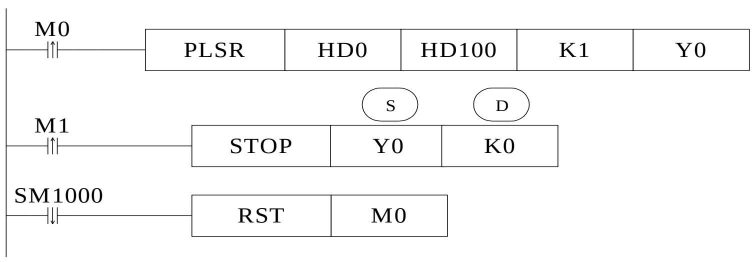
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S		●					

*Note: D means D, HD; TD means TD, HTD; CD means CD, HCD, HSCD, HSD. DM means DM, DHM; DS means DS, DHS.

M means M, HM, SM; S means S, HS; T means T, HT; C means C, HC.

Function and action

Instruction format

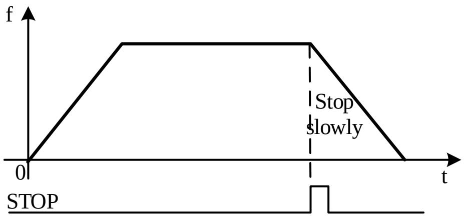


- Pulse stop mode: K0 (stop slowly), K1(scram)
- When M0 is from OFF to ON, PLSR instruction outputs pulse from Y0, and stop pulse outputting when the pulse output numbers reached setting value
- At the rising edge of M1, STOP instruction will stop the pulse outputting of Y0 immediately,

as the D parameter is K0, the pulse will stop slowly.

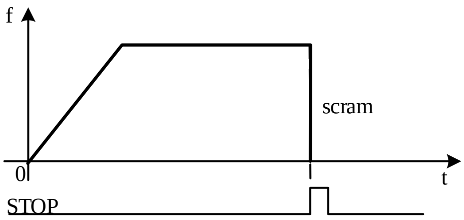
- Stop pulse includes PLSR, PLSF, DRVI, DRVA, ZRN.

- Stop slowly (K0)



According to the descending slope, the current pulse frequency of the pulse falls to the pulse stop frequency or the number of pulses in the pulse section is all sent out and stop the pulse output.

- Scram (K1)



Stop the pulse outputting immediately.

1-2-8. Pulse continue [GOON]

1. Instruction overview

Continue the pulse output.

Pulse continue [GOON]			
16-bit instruction	GOON	32-bit instruction	-
Execution condition	Rising/falling edge of the coil	Suitable model	XD, XL (except XD1, XL1)
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S	The terminal to continue outputting the pulse	bit

3. Suitable soft component

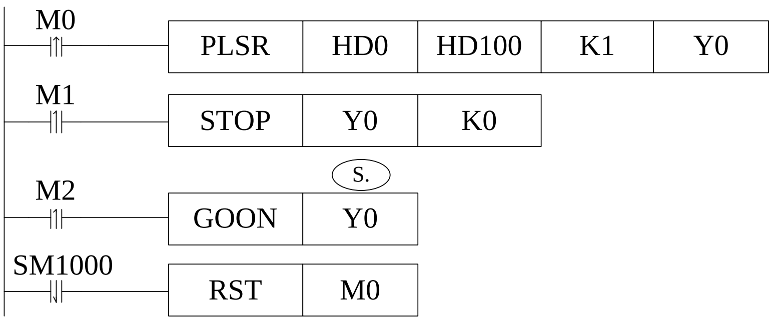
Bit	Operand	System						
		X	Y	M*	S*	T*	C*	Dn.m
	S		●					

*Note: D means D, HD; TD means TD, HTD; CD means CD, HCD, HSCD, HSD. DM means DM, DHM; DS means DS, DHS.

M means M, HM, SM; S means S, HS; T means T, HT; C means C, HC.

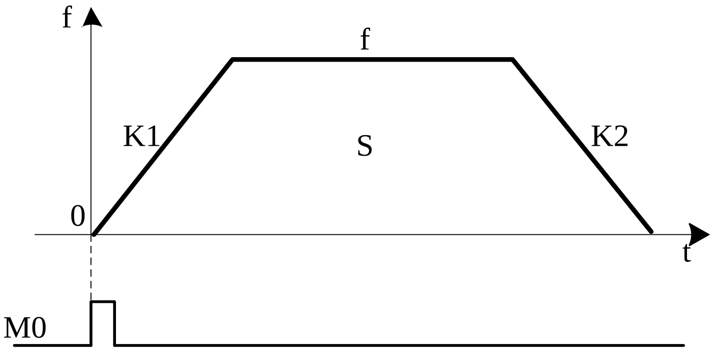
Function and action

Instruction format

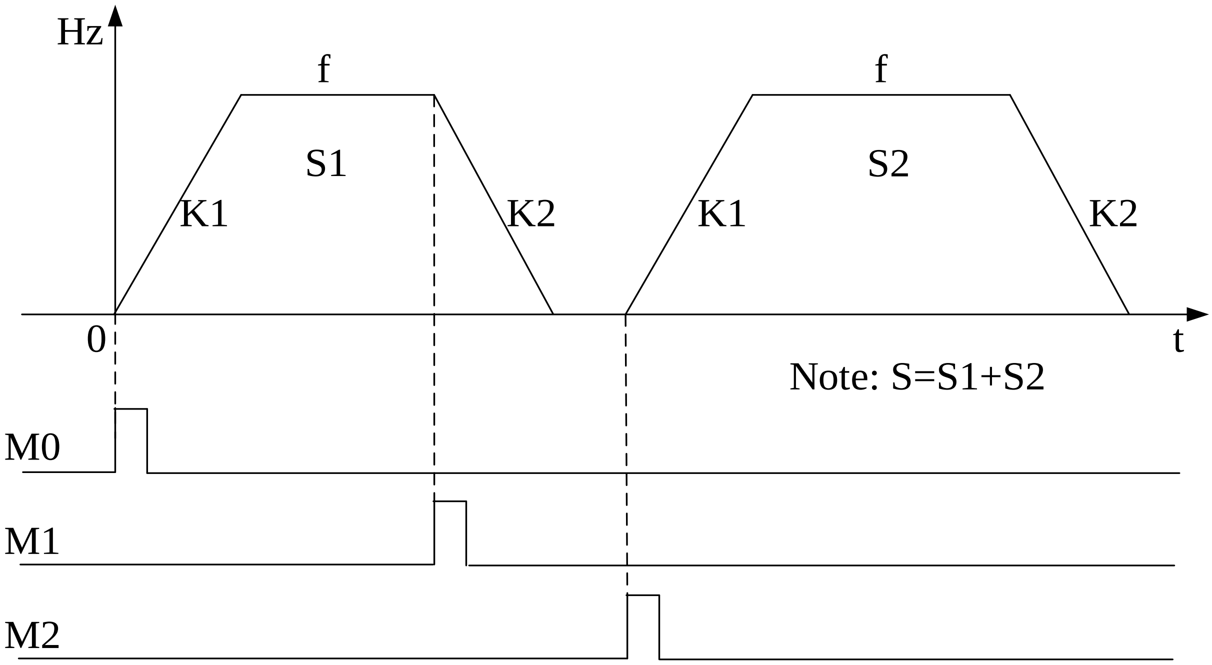


- When M0 from OFF to ON, PLSR instruction outputs pulse from Y0; When the number of output pulses reaches the set value, stop the output pulse.
- In the process of sending pulse, M1 from OFF to ON rising edge, STOP instruction immediately stop Y0 pulse outputting, as the parameter is K0, so the pulse will stop slowly;
- when M2 from OFF to ON rising edge, GOON Y0 instruction is executed, remaining pulses will send out according to the original deceleration slope.
- Please set ON M2 after pulse stop, otherwise GOON will not send pulse.
- Pulse continue instruction is applicable to the PLSR, DRVI, DRVA instructions.

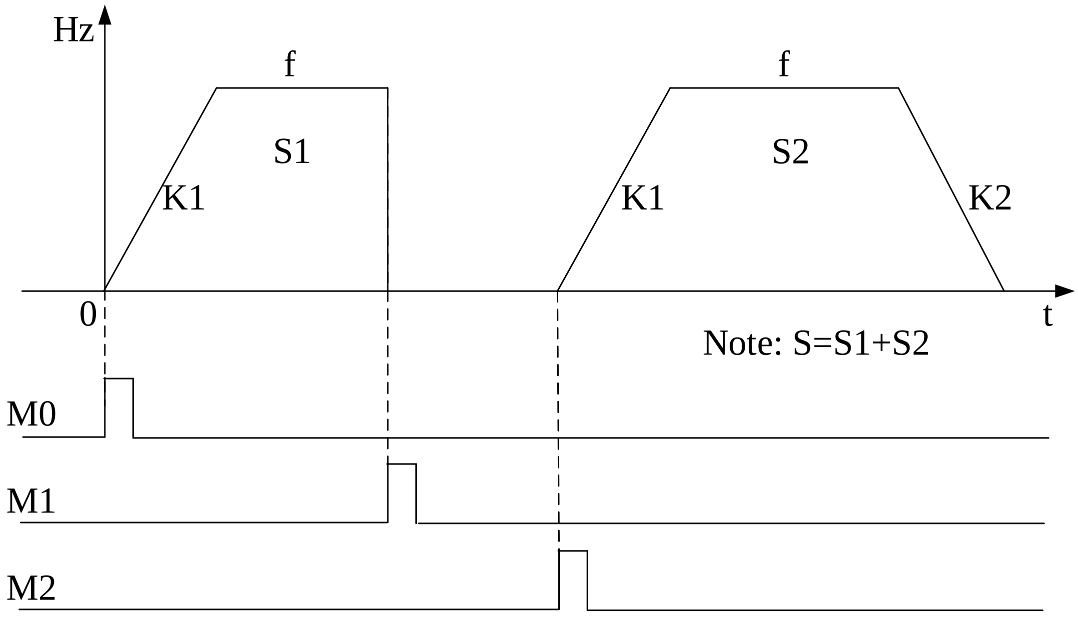
- The schematic diagram is as follows:



Complete Pulse diagram



Pulse continue wave diagram（STOP Y0 K0）



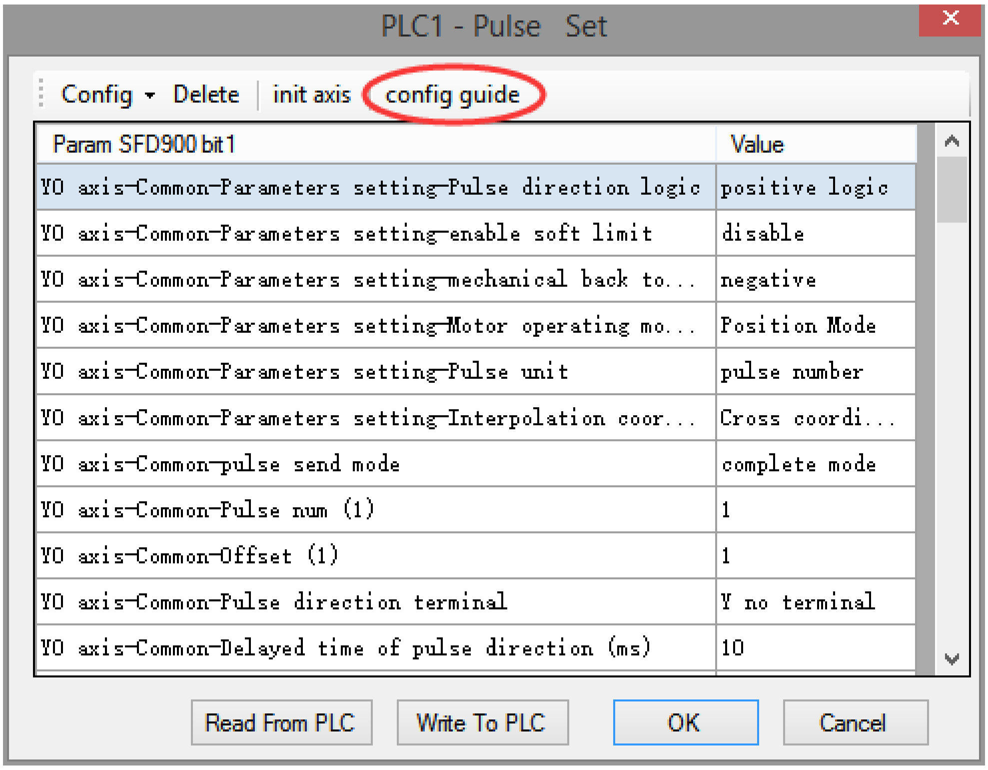
Pulse continue wave diagram（STOP Y0 K1）

1-3. Pulse parameter configuration wizard

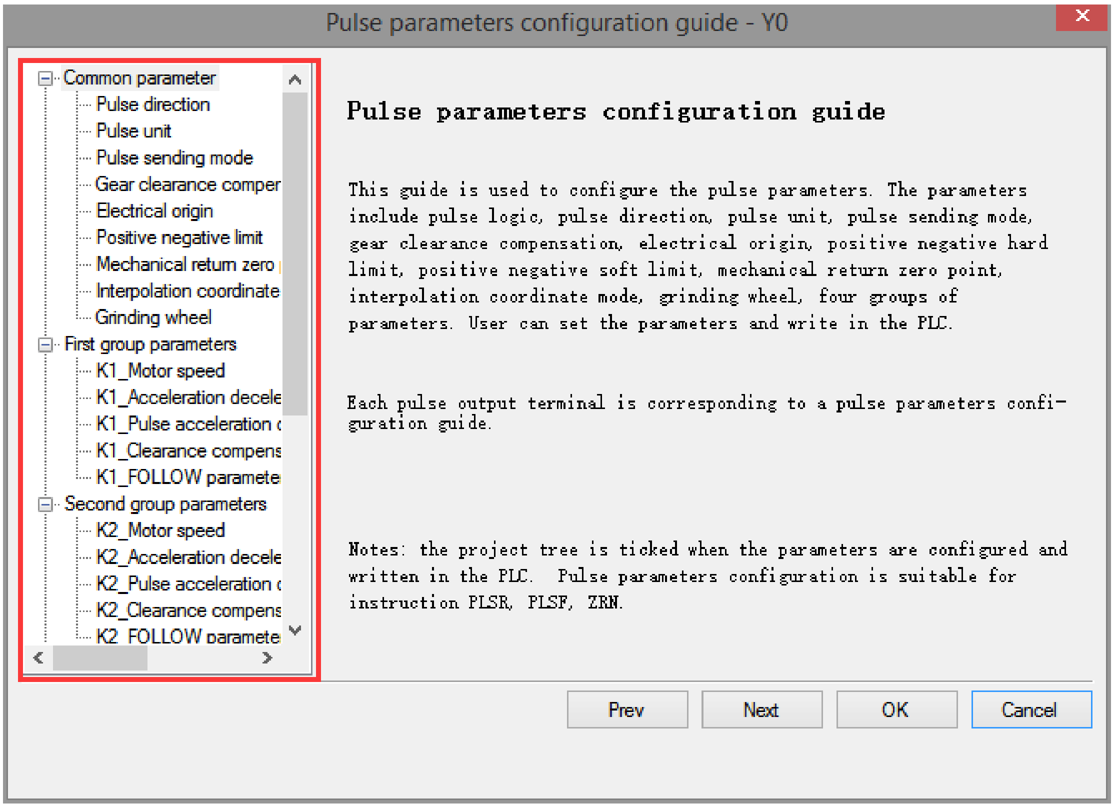
Pulse parameter configuration wizard function was added in V3.3.2 and higher version software. Because there are many system parameters of the pulse axis (including common parameters and the first to fourth sets of parameters), it may be difficult for novices. To solve this problem, a pulse parameter configuration wizard is added to the latest PC software, which configures the pulse parameters of each pulse axis directly through the pulse parameter configuration wizard, which is simple and convenient.

1-3-1. Pulse Parameter Configuration Wizard Opening Mode

On the top of the pulse parameter configuration interface, there is a "Config guide" option. Click on the "Configuration Wizard" to open the pulse parameter configuration wizard. As shown in the figure:

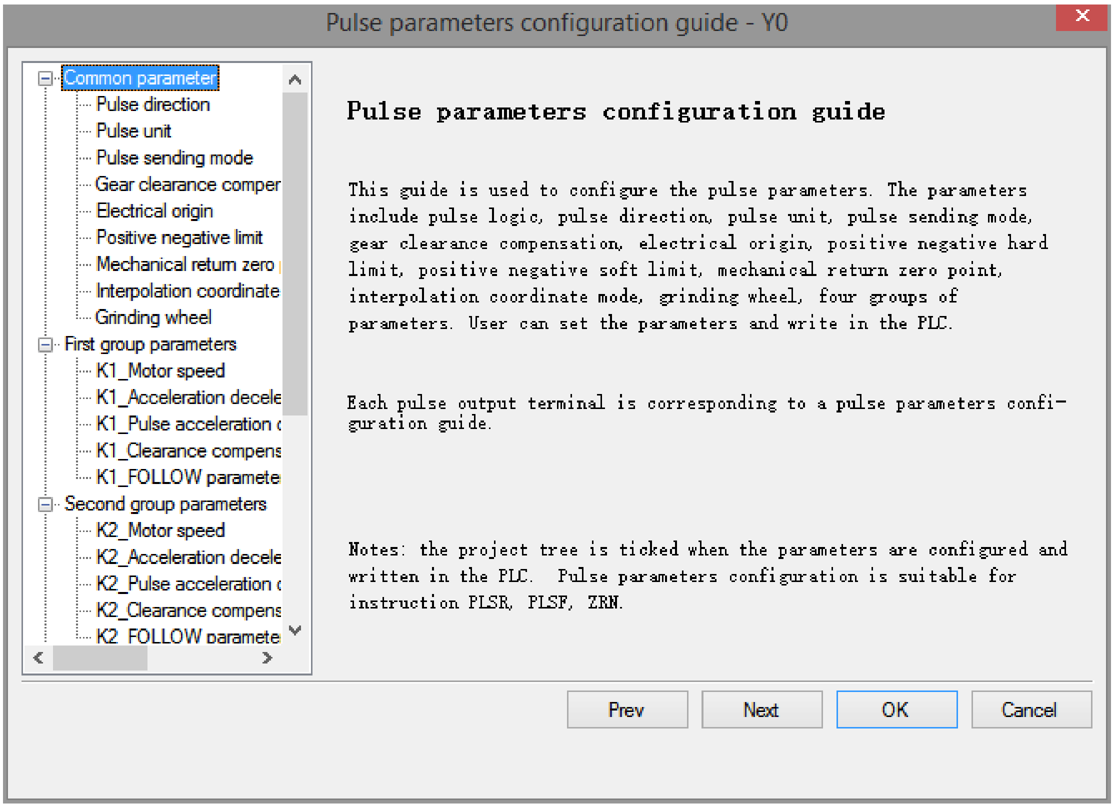


Engineering Tree is on the left of the following window. You can select the option you want to open in the Engineering Tree, and click directly to open it quickly. As shown in the figure:



1-3-2. Instructions for the Use of the Pulse Parameter Config guide

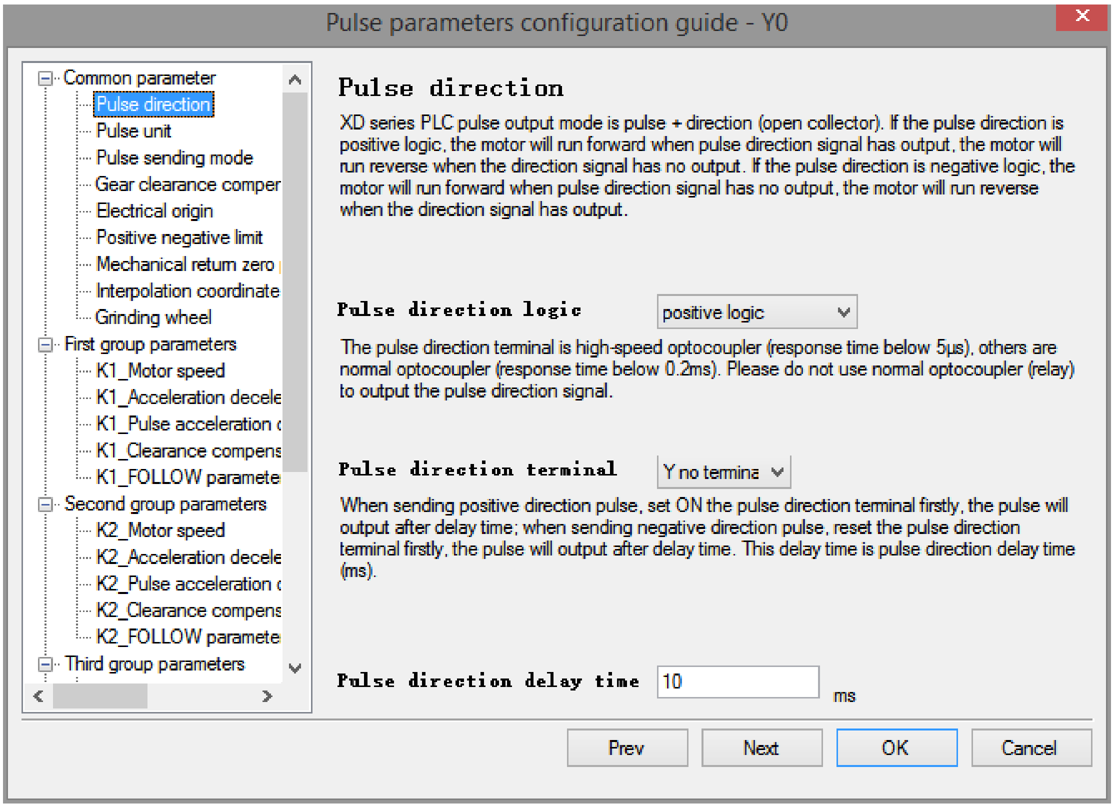
The pulse parameter config guide describes:



This interface is mainly used to briefly explain the pulse parameter configuration wizard.

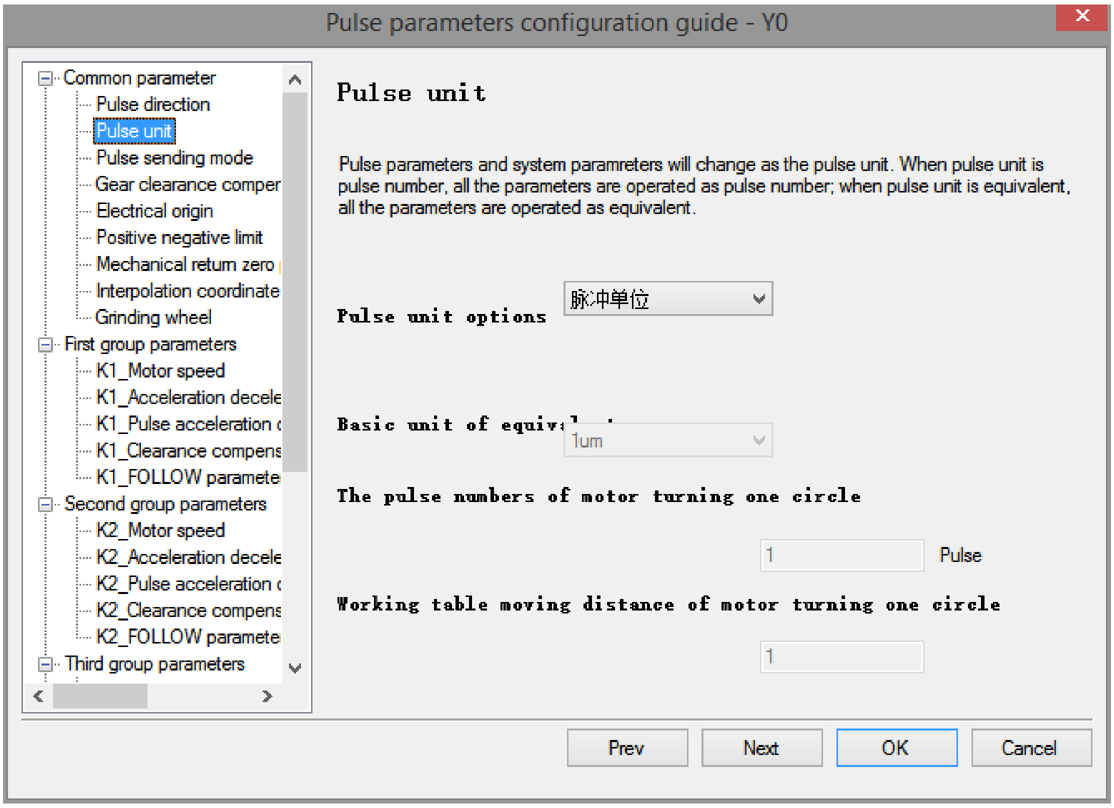
★ Common parameter—pulse direction

It is used to set the pulse direction logic, the pulse direction terminal and the delay time of the pulse direction.

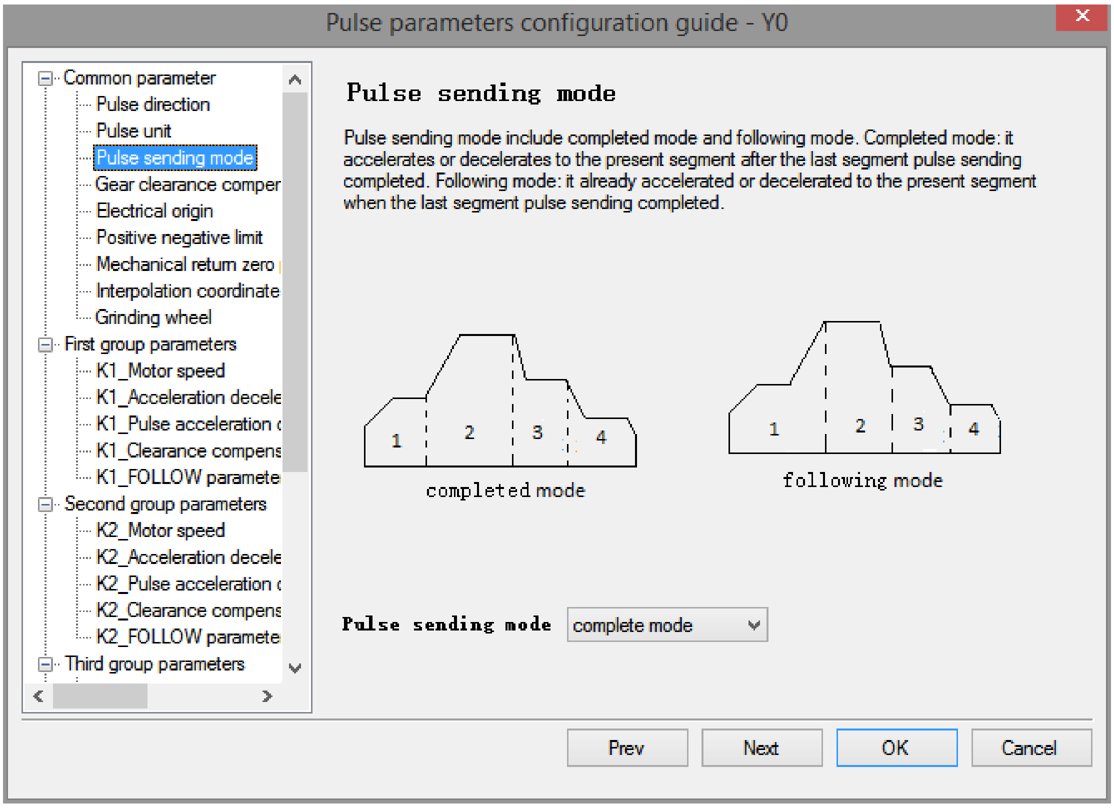


★ **common parameters—pulse unit**

It is used to set the unit of pulse, the basic unit of equivalent, the number of pulses and the amount of movement.

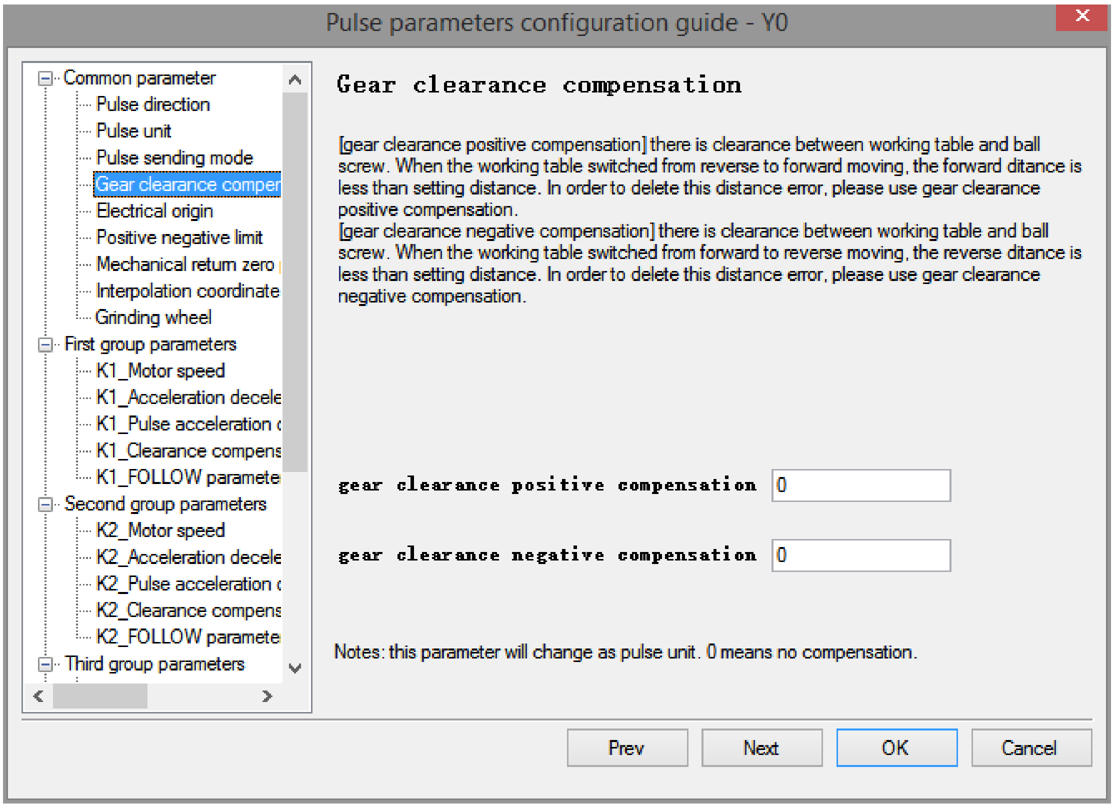


★ Common parameters—pulse sending mode

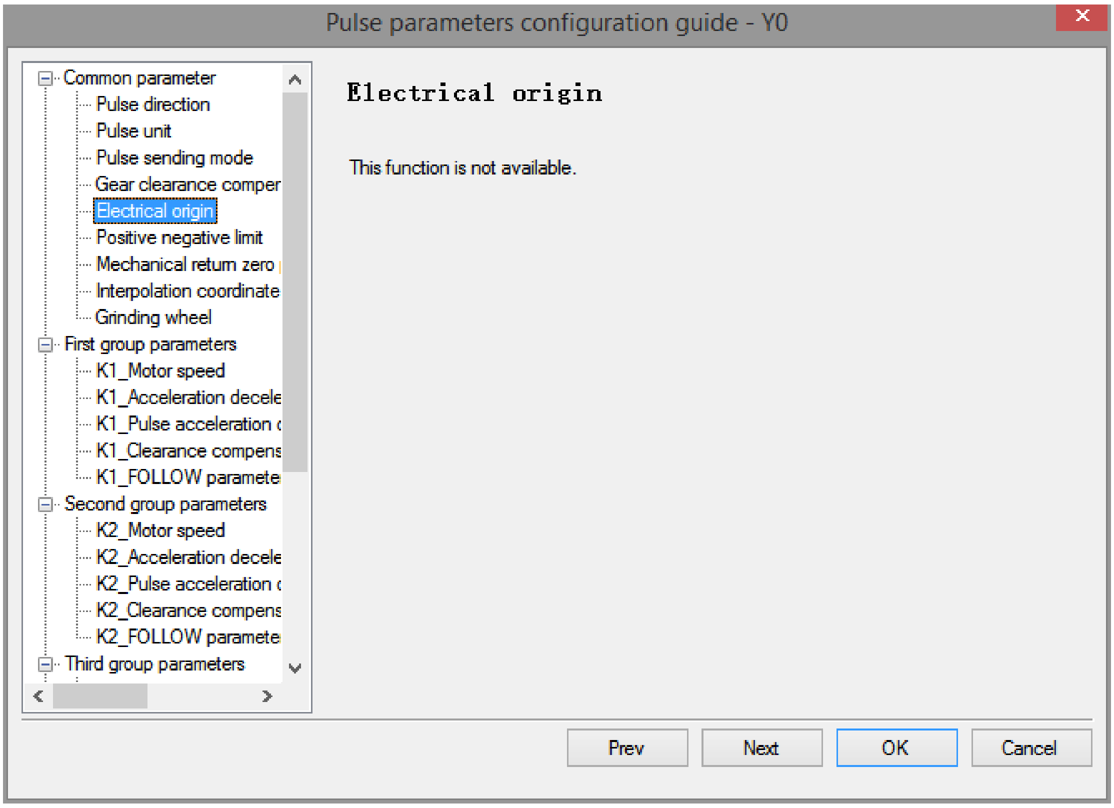


★ Common parameters—gear clearance compensation

It is used for setting forward compensation of gear clearance and reverse compensation of gear clearance.

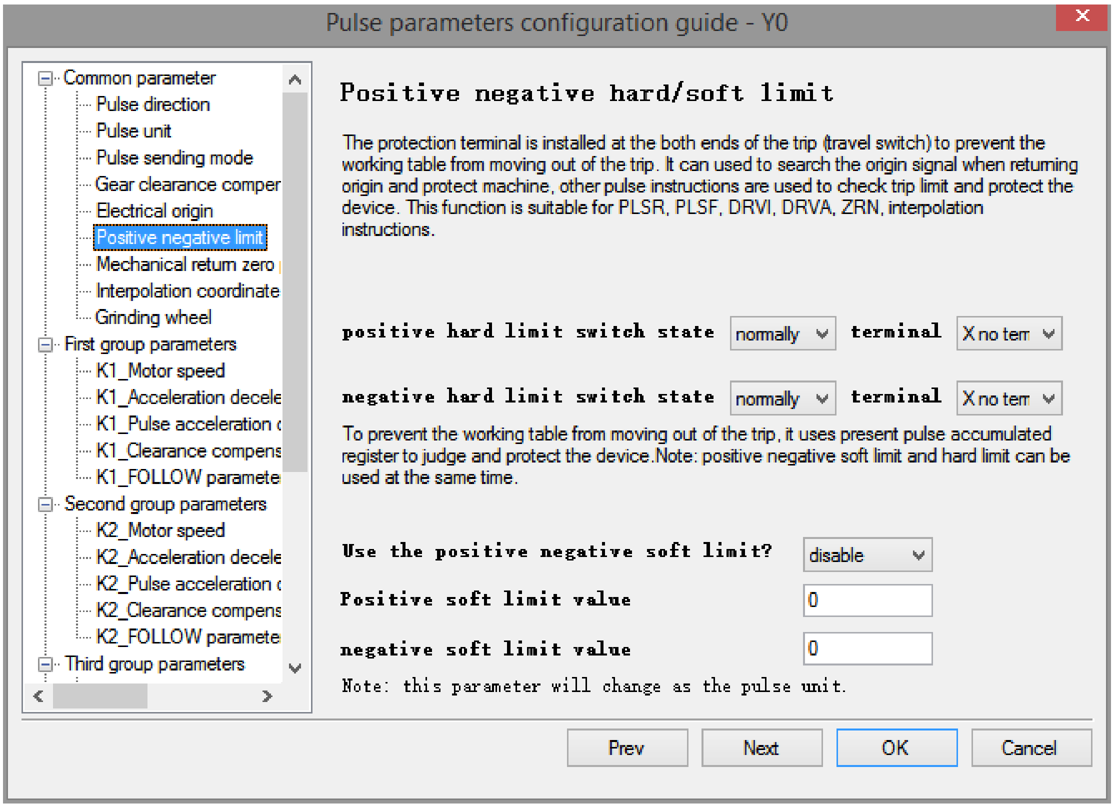


★ Common parameters —electric origin



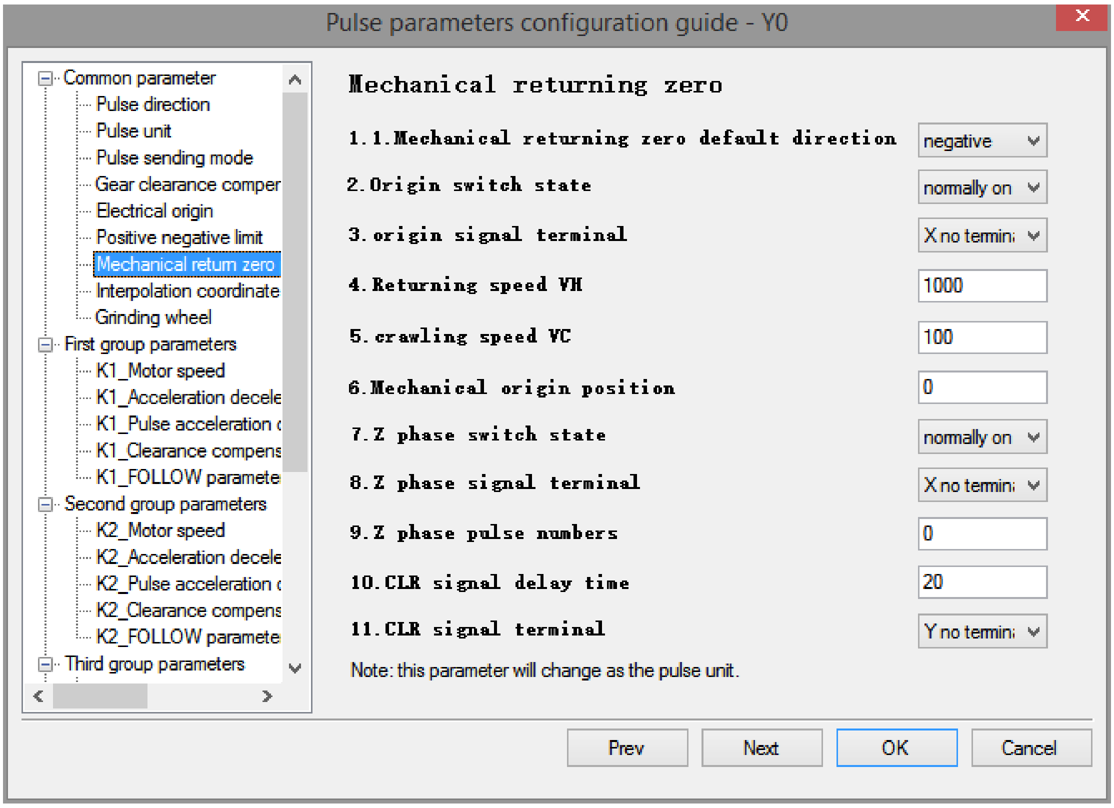
★ Common parameters—positive neagtive hard/soft limit

Used for setting positive and negative hard limit and positive and negative soft limit.



★ Common parameters—Mechanical Zero Return Setting

Used to set the default direction of mechanical zero return, origin switch, Z phase switch, regression speed, CLR signal, mechanical origin position.

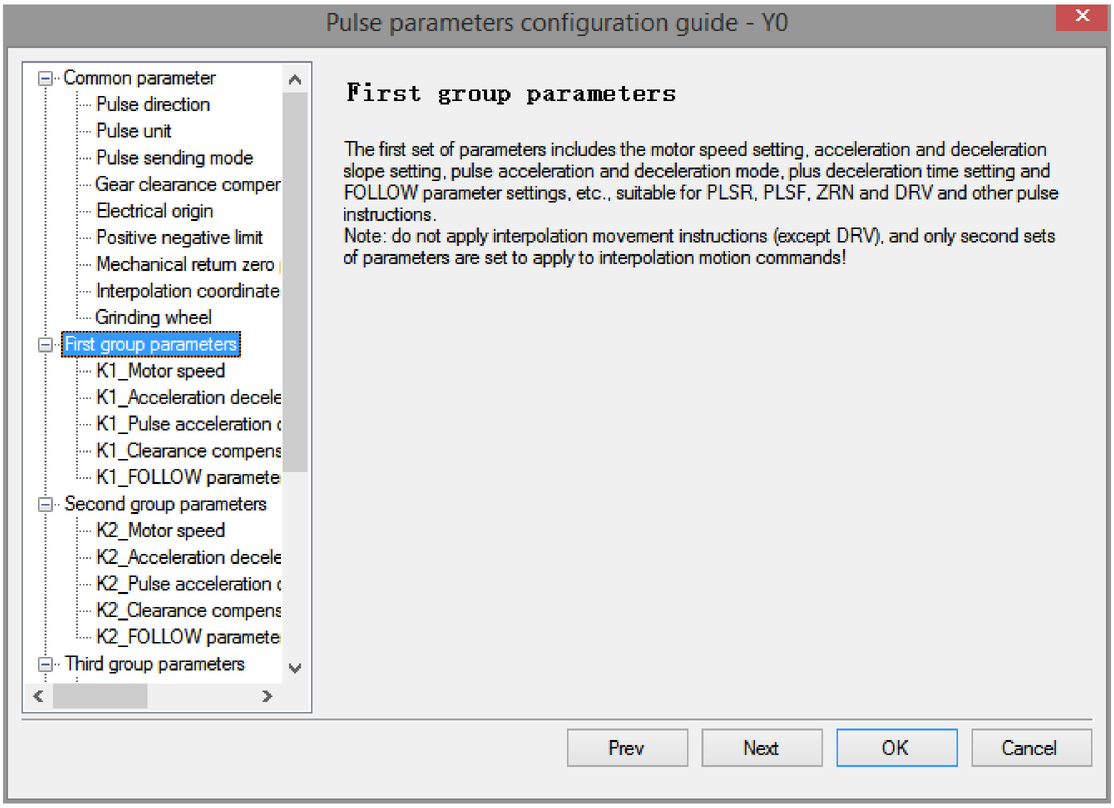


★ Common parameters —Interpolation coordinate mode

★ Common parameters —grinding wheel radius

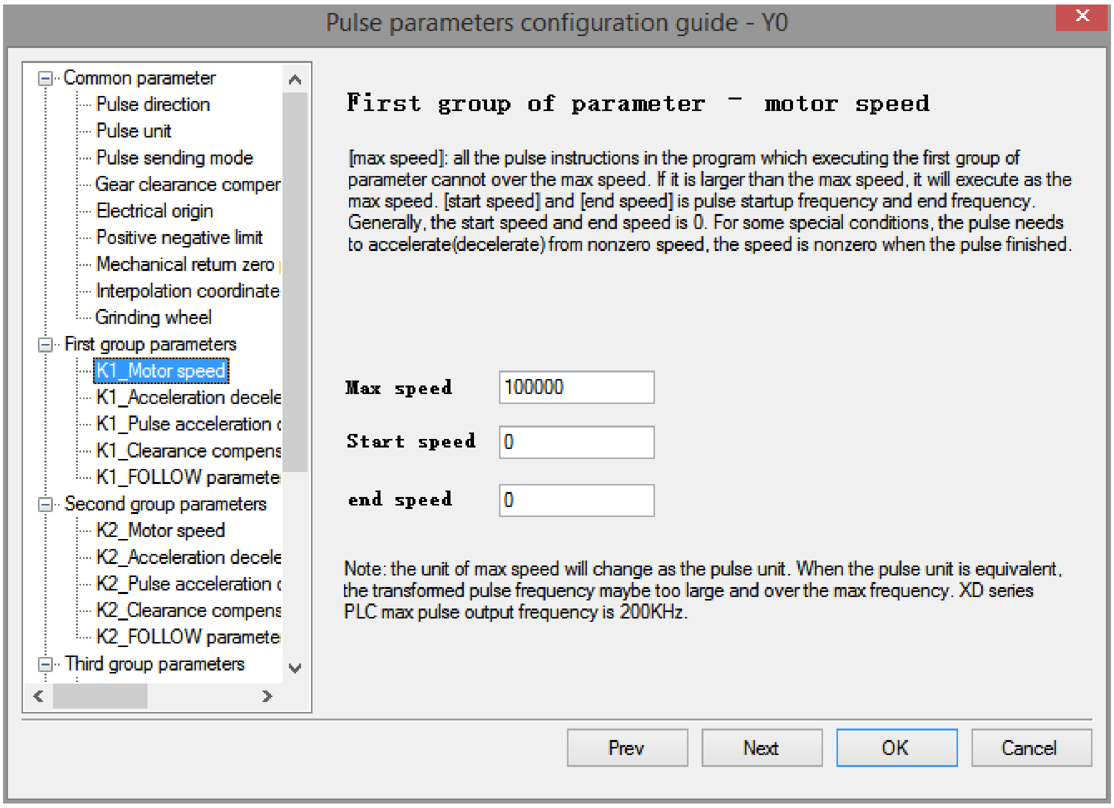
The functions are not avaliable.

★ First group parameter setting



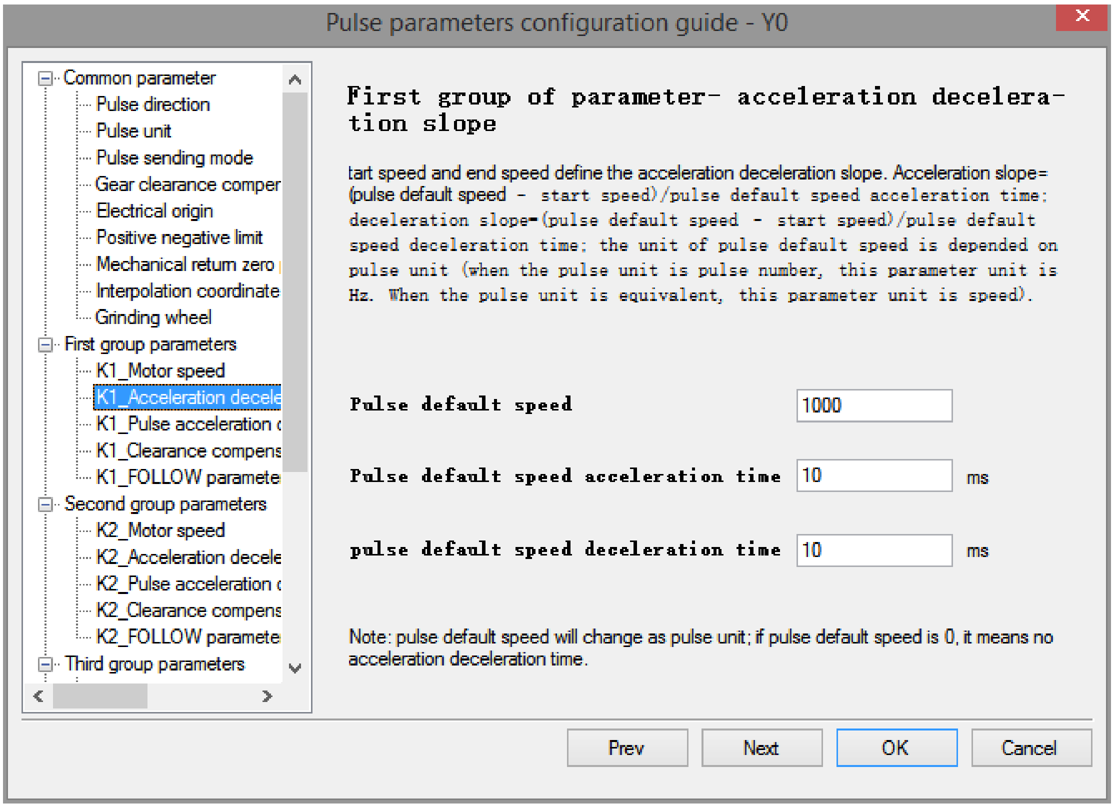
★ First group parameters—motor speed

Used to set the maximum speed, starting speed, termination speed.



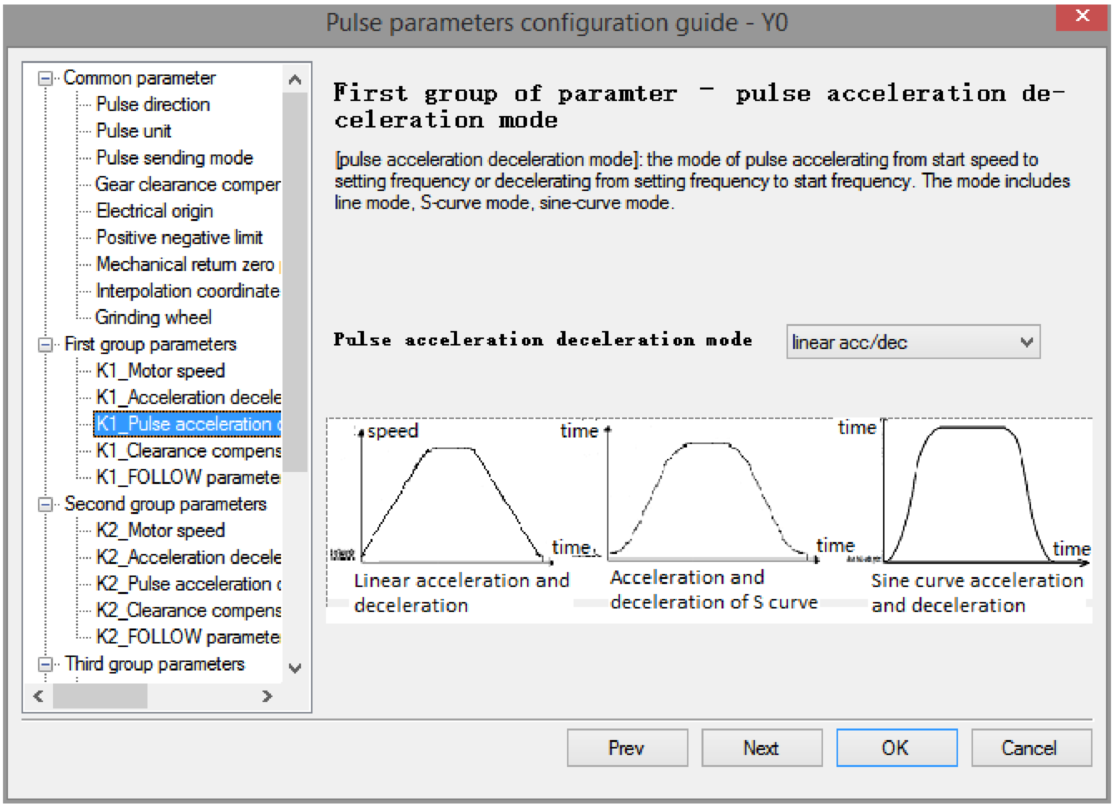
★ First group parameters —Acceleration and deceleration slope

Used to set default speed, default speed acceleration time, default speed deceleration time.



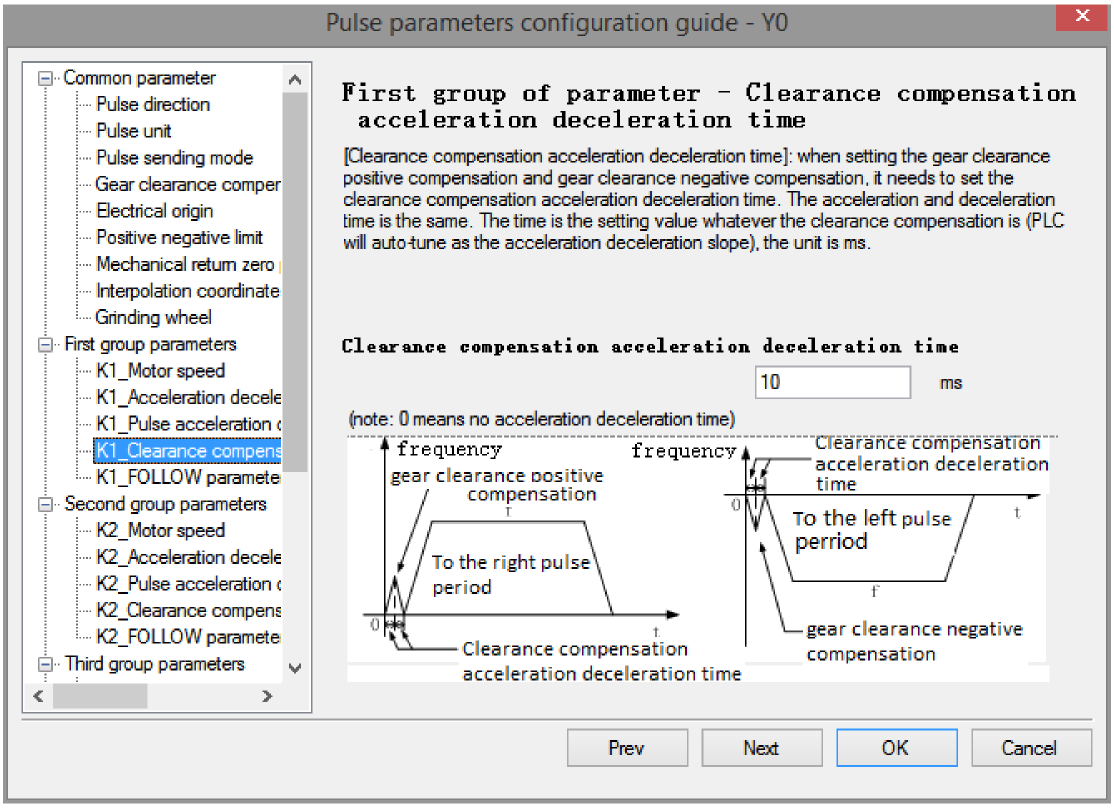
★ First group parameters —Pulse acceleration and deceleration mode

It is used to set three pulse acceleration and deceleration modes.



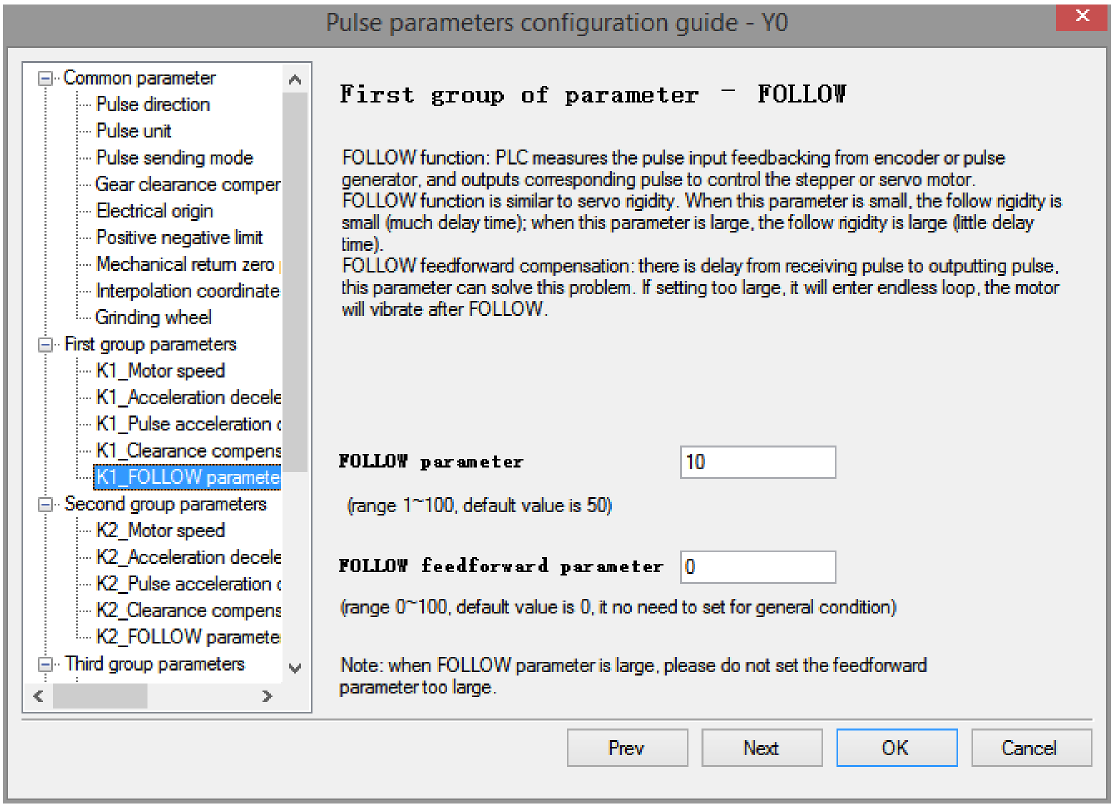
★ **First group parameters —Clearance compensation acceleration and deceleration time**

It is used to set the clearance compensation acceleration and deceleration time.



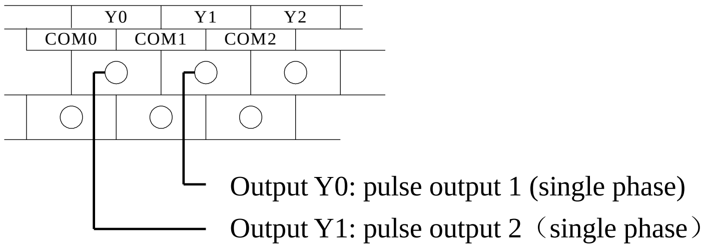
★ **First group parameters —FOLLOW parameter**

It is used to set the FOLLOW parameter and feedforward parameter.

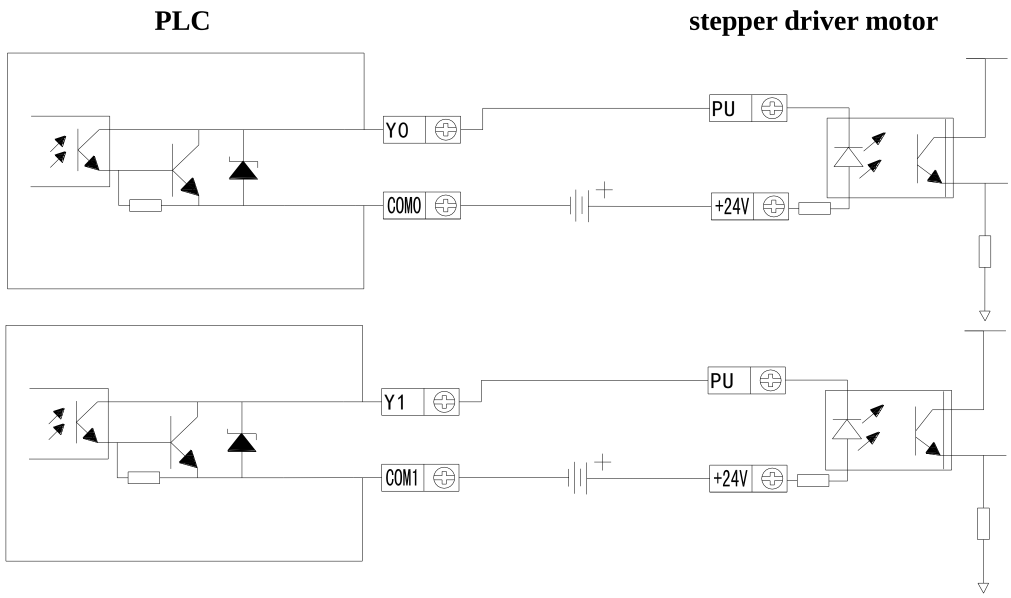


The second to fourth group of parameters are the same as the first group of parameters, please refer to the first group of parameters! After configuring the parameters, the program is downloaded to the PLC again, and then the power is cut off and restarted to take effect.

1-4. Output wiring and notes

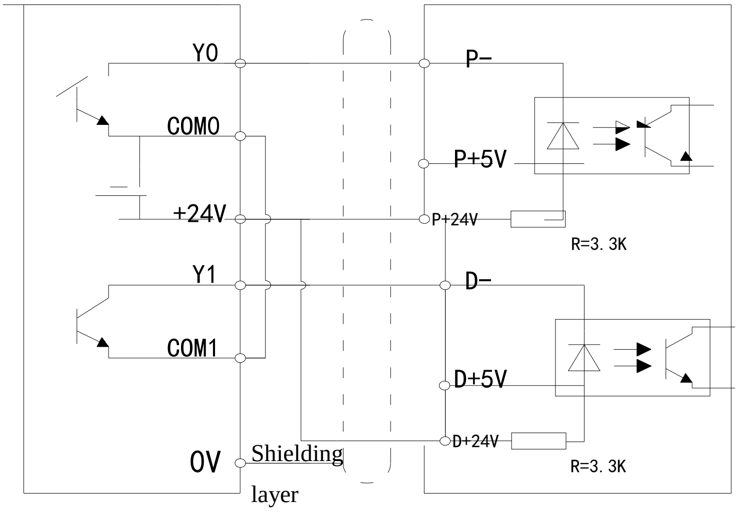


Below is a wiring diagram of the connection between the T-type output terminal and the stepper motor driver.



Note: If the pulse and direction terminals of stepper motor are driven by DC5V, please connect 2.2K resistance behind the pulse output terminal and direction output terminal.

Below is a wiring diagram of the connection between the T-type output terminal and the XINJE servo motor driver.

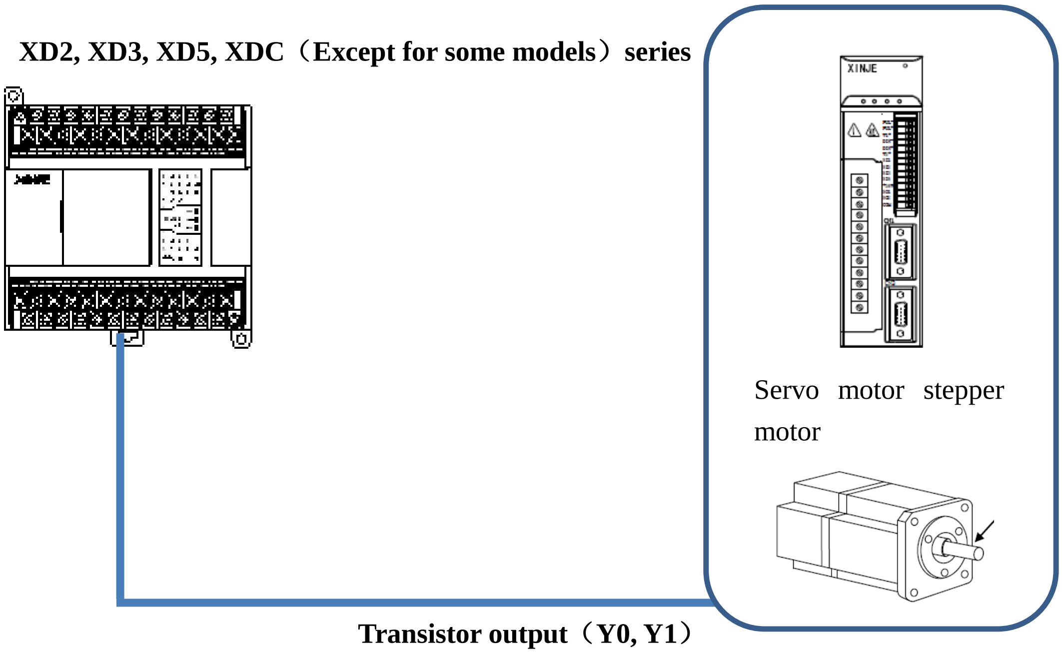


Note: Please suspend P+5V and D+5V.

Detailed hardware wiring diagram refers to XD/XL Series Programmable Controller hardware User Manual.

1-4-1. Composition of Connecting Equipment

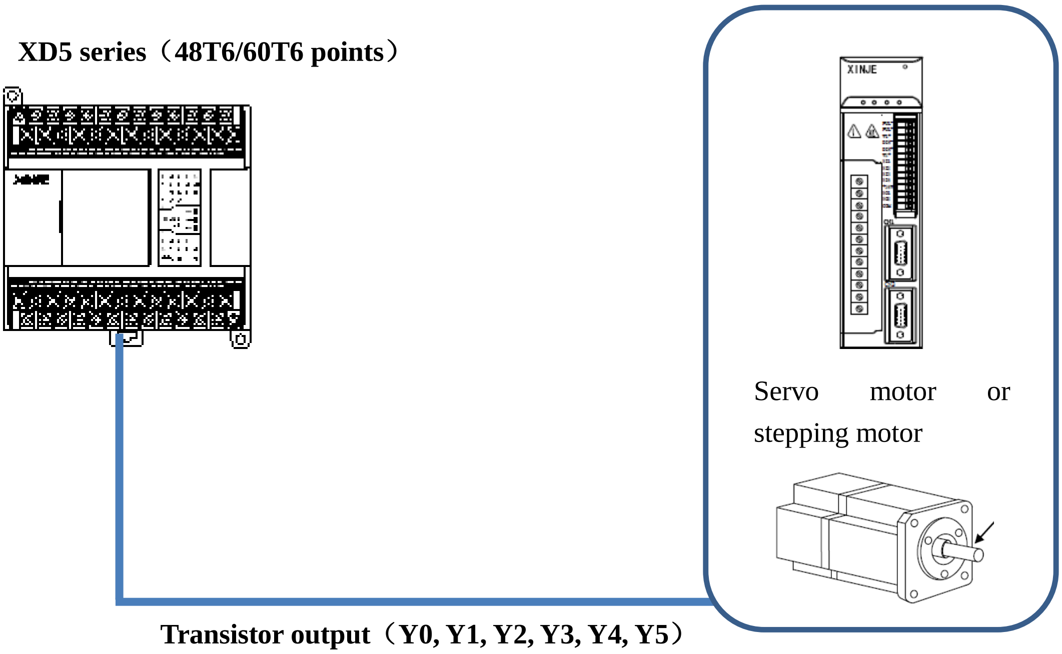
- **XD2, XD3, XD5, XDC series PLC**



※:Two-axis servo motor or stepping motor can be controlled.

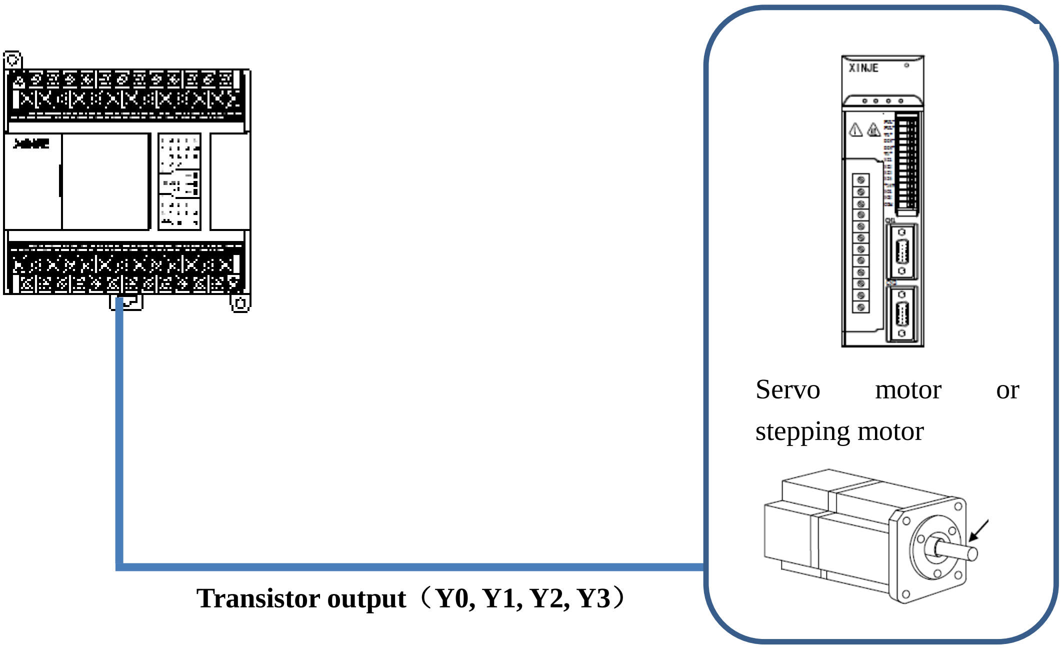
- **XD5, XDM, XD5E, XDME series PLC**

XD5 series（48T6/60T6 points）



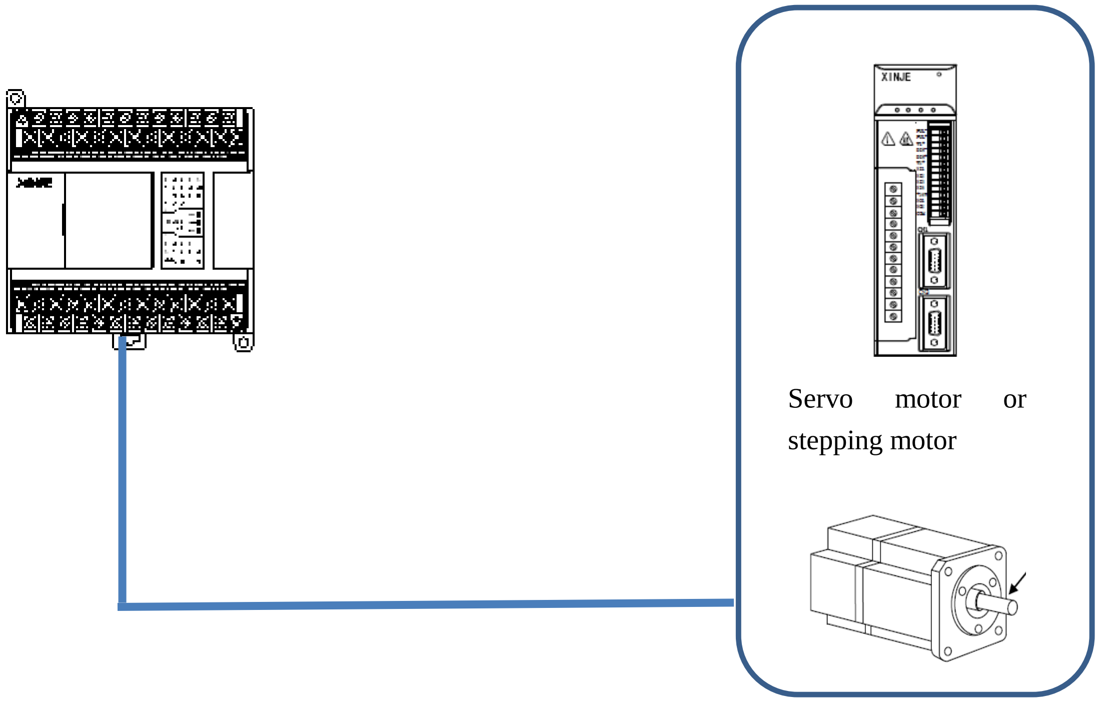
※: Six-axis servo motor or stepping motor can be controlled.

XD5/XDM series（24T4/32T4）, XDM series（60T4/60T4L）, XD5E-30T4



※: Four-axis servo motor or stepping motor can be controlled.

XDM series（60T10）, XD5E series (60T10), XDME series (60T10)

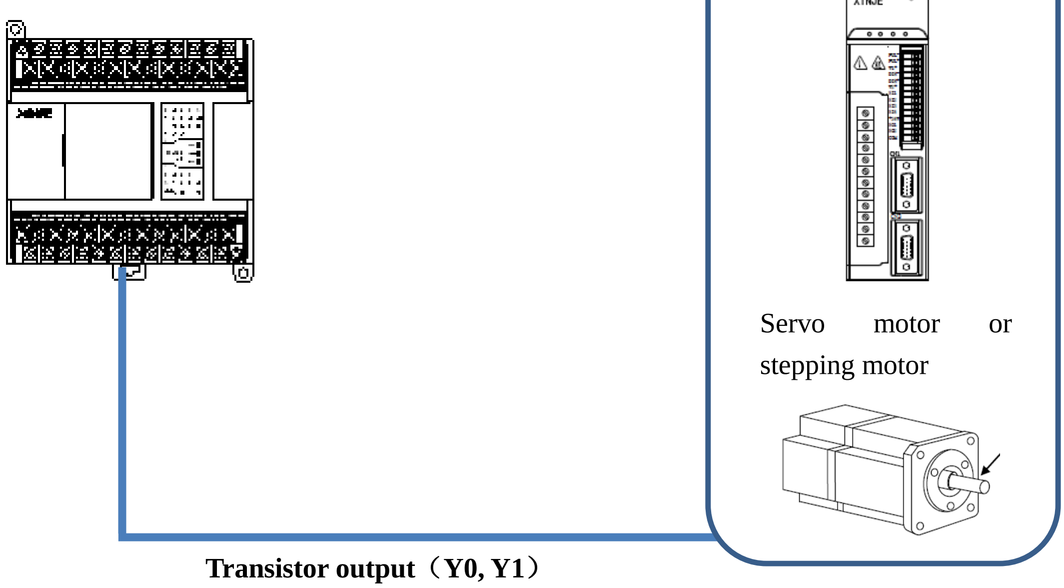


Transistor output（Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11）

※: Ten-axis servo motor or stepping motor can be controlled.

- **XDC series PLC**

XDC series

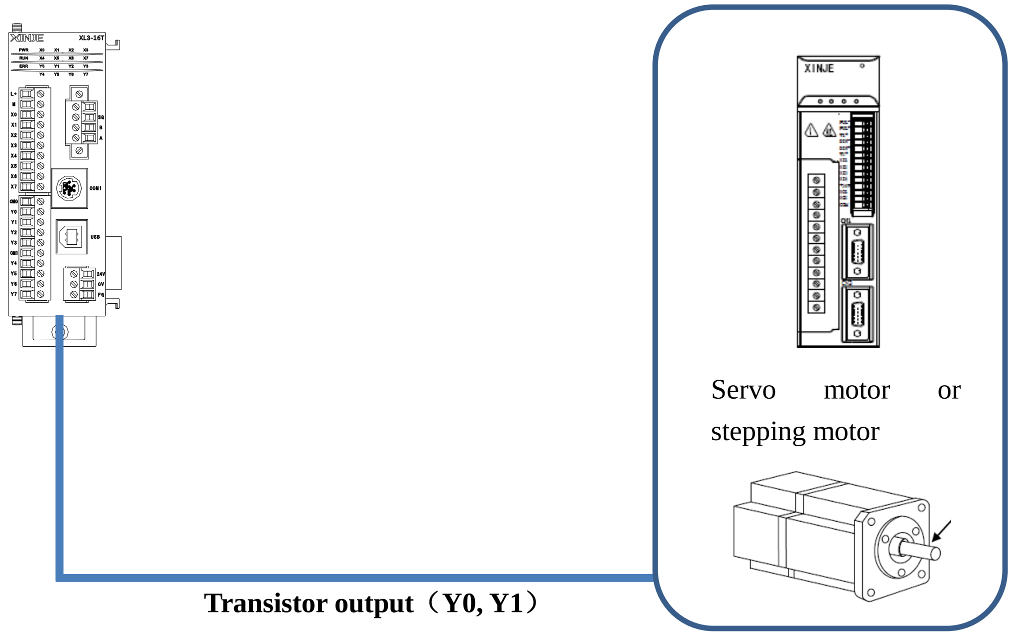


Transistor output（Y0, Y1）

※: Two-axis servo motor or stepping motor can be controlled.

- **XL3 series PLC**

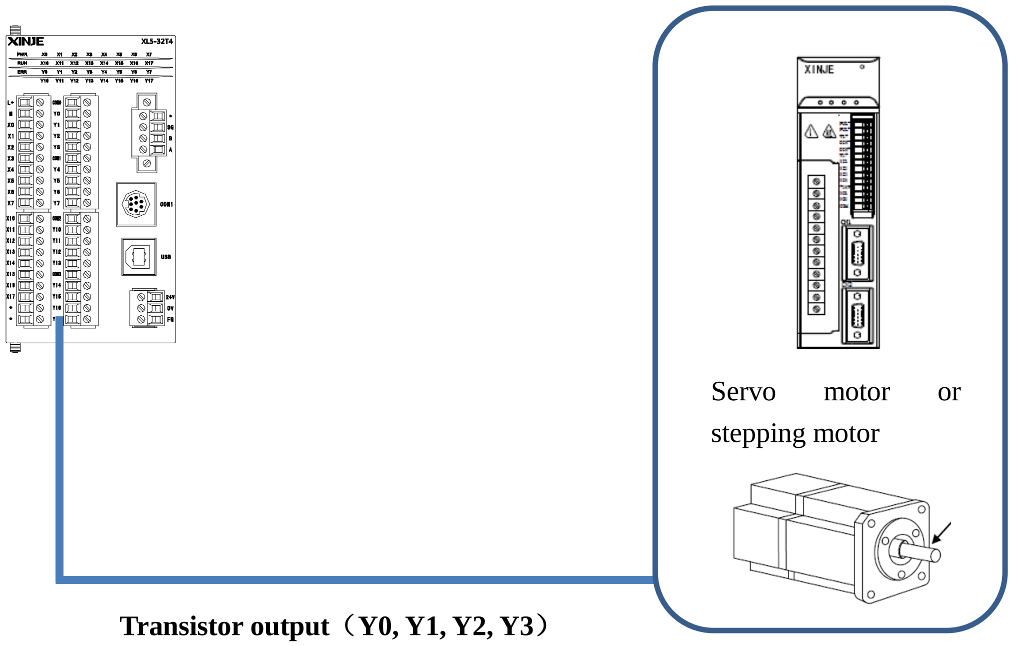
XL3 series



※: Two-axis servo motor or stepping motor can be controlled.

- **XL5, XL5E series PLC**

XL5, XL5E series



※: 4-axis servo motor or stepping motor can be controlled.

1-4-2. Pulse output performance specification

Pulse output performance specification:

Parameter name	XD2/XD3/XDC series XD5-16/24/32/48/60 XL3-16	XD5-48T6/60T6	XDM-24T4/32T4 XDM-24T4/32T4/60T4 XD5E-30T4 XDM-60T4L XL5-32T4 XL5E-32T4	XDM-60T10 XD5E-60T10 XDME-60T10
Number of control axes	Independent 2 axis	Independent 6 axis	Independent 4 axis	Independent 10 axis
Interpolation function	nonsupport	nonsupport	support	support
Output mode	Open circuit mode of collector			
Output form	Pulse + direction			
Max frequency	100KHz			
Acceleration and deceleration treatment	Linear acceleration and deceleration + S curve acceleration and deceleration + sine curve acceleration and deceleration			
Control unit	Pulse, 1mm, 0.1mm, 0.01mm, 1um			
Positioning range	-2147483648~2147483647（pulse）			
Programming language	Ladder chart			
Manual pulse connection	nonsupport	nonsupport	Support(only XDM support)	support

Note:

(1) All XD/XL series PLC's pulse output must be transistor output type, otherwise it can't send pulse!

(2) PLC can output high-speed pulses ranging from 100KHz to 200KHz, but it can not guarantee the normal operation of all servos. Please connect 500Ω resistance between the output and 24V power supply.

1-4-3. Positioning control layout and wiring notes

>>>>> Design notes<<<<<

Danger!

Please set up a safety circuit outside the programmable controller, so that when there are abnormal external power supply and programmable controller failure, the whole system can also be ensured to operate in a safe state. Misoperation and misoutput may lead to accidents.

1. Make sure to set up emergency stop circuit, protection circuit, interlocking circuit to prevent reverse and positive actions simultaneously, positioning upper and lower limits and other interlocking circuits to prevent mechanical breakage outside the programmable controller.

2. When the programmable controller CPU detects abnormalities through self-diagnostic functions such as watchdog timer, all outputs become OFF. In addition, when abnormalities occur in the input and output control parts which cannot be detected by the programmable controller CPU, the output control sometimes fails.

At this point, please design the external circuit and structure to ensure that the machine is running in a safe state.

3. Because of the faults of relays, transistors, thyristors and so on in the output unit, sometimes the output is always ON or OFF.

In order to ensure the safe operation of machinery, please design the external circuit and structure for the output signal which may lead to major accidents.

Attention!

1. The control line should not be tied up with the main circuit or power line, or close to the connection.

   In principle, please leave more than 100 mm or away from the main circuit. Otherwise, the noise will cause misoperation.

2. When using, please ensure that the built-in programming interface, power connector, input and output connector are not subject to external forces.

   Otherwise, it will lead to disconnection and malfunction.

>>>>> Wiring notes <<<<<

Danger!

1. When installing, wiring and other operations, be sure to disconnect all external power supply before operation.

Otherwise, there is a risk of electric shock and product damage.

2. After installation, wiring and other operations, when running on power, be sure to install the attached wiring terminal cover on the product.

Otherwise, there is a risk of electric shock.

Attention！
1. AC power supply wiring should be connected to the special terminals recorded in the basic unit manual. If AC power supply is connected to DC output input terminal and DC power supply terminal, the programmable controller will be burned down. 2. DC power supply wiring should be connected to the special terminals recorded in the basic unit manual. If AC power supply is connected to DC output input terminal and DC power supply terminal, the programmable controller will be burned down. 3. Please do not wiring the empty terminals outside. It may damage the product. 4. Grounding terminals of basic units of XD/XL series should be D grounded with wires over 2 mm^2 (grounding resistance below 100Ω). However, do not grounding with strong current (refer to XD/XL Series Programmable Controller hardware User Manual). 5. When processing bolt holes and wiring operations, do not drop chips and wire chips into the ventilation holes of the programmable controller. Otherwise, it may lead to fire, malfunction and misoperation. 6. When using, make sure that the input and output connectors are not subject to external forces. Otherwise, it will lead to disconnection and malfunction. 7. The input and output cables should be firmly mounted on the specified connectors. Poor contact can lead to erroneous movements. 8. When wiring the basic units of XD/XL series and terminal of XD/XL series extension equipment, please follow the following precautions. Otherwise, it may lead to electric shock, fault, short circuit, wire breakage, misoperation and damage to the product. - Please process the end of the wire according to the size recorded in the manual. Tightening torque, please follow the torque recorded in the manual.

>>>>> Cautions in Starting and Maintenance <<<<<

Danger！
1. Do not touch the terminal when electrifying. Otherwise, there is the danger of electric shock, and it may cause misoperation. 2. When cleaning and tightening terminals, be sure to operate after disconnecting all external power supply. If operated in the state of electrification, there is a danger of electric shock. 3. In order to change procedures, perform mandatory output, RUN, STOP and other operations

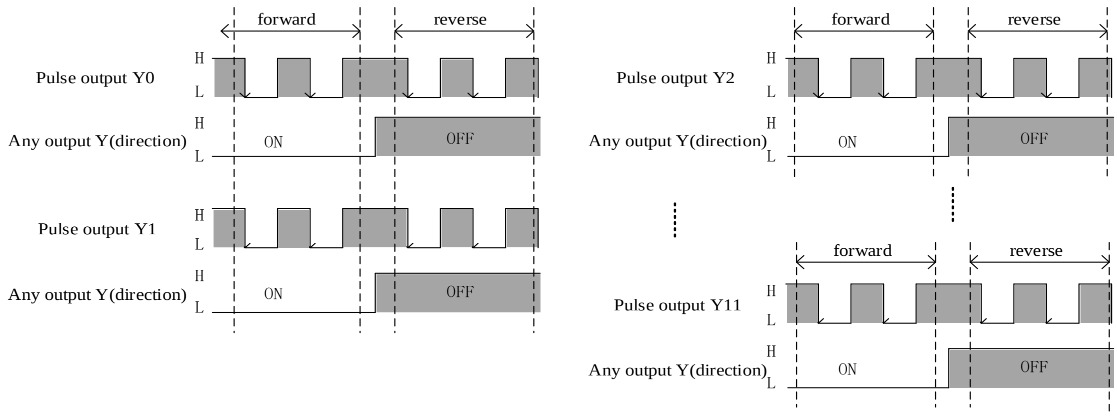
during operation, you must read the manual well before you can operate it with full confirmation of safety. Operational errors may lead to mechanical damage and accidents.

Attention!
1. Do not disassemble or alter products without authorization. Otherwise, it may cause malfunction, misoperation and fire. 2. When disassembling and assembling connecting cables such as extended cables, please operate after disconnecting the power supply. Otherwise, it may cause malfunction and misoperation. 3. Be sure to cut off the power supply when disassembling and assembling the following equipment. Otherwise, it may cause malfunction and misoperation. --Peripheral devices, extended function boards, special adapters, --Input and Output Extension Module, Network Module, etc.

1-4-4. Setting of Servo Amplifier (Driving Unit) Side

Pulse Output Form of Programmable Controller Side

The pulse output types of XD/XL series PLC are all collector open circuit signals (pulse + direction), as shown in the following figure:



Note: ON and OFF represent the output state of the programmable controller; H and L represent the waveform of HIGH and LOW.

- **Setting of Instruction Pulse Input Form for Servo Amplifier (Driving Unit)**

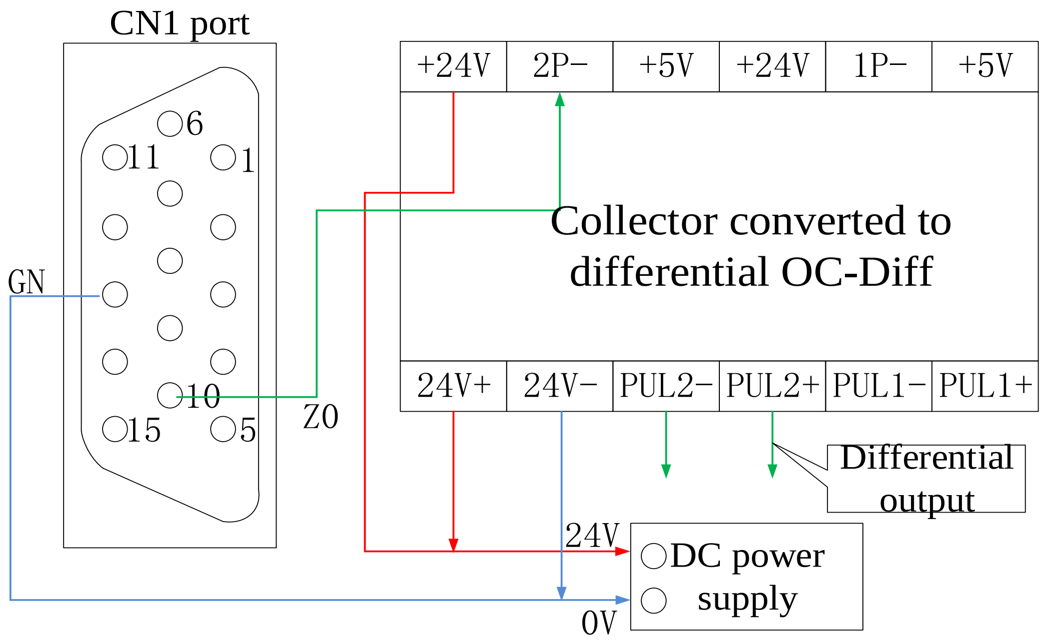
As shown in the table below, please make the input form of the pulse in the parameters of servo amplifier (driving unit) coincide with the output form of the programmable controller.

servo amplifier (driving unit)	Pulse output form of basic unit	Collector convert to differential DC-Diff

	Transistor output (Leakage output)	Differential drive
	Pulse + direction	Forward and reverse pulses
Instruction pulse input form	Pulse + sign	Forward and reverse pulses
Instruction pulse logic	Negative logic	Negative logic

Note: The main pulse output form of XD/XL series PLC is collector open-circuit signal output (pulse + direction). The collector open-circuit signal output (pulse + direction) can be converted into differential signal output through collector-to-differential expansion board DC-Diff.

Wiring diagram of the open collector signal (pulse + direction) converted into differential signal by DC-Diff (taking DS2-21P5-A as an example):



DS series servo driver parameter settings:

Series	Parameter	Settings	
		Pulse+direction (negative logic)	Differential signal (negative logic)
DS2-AS	—	√	—
DS2-AS2	—	√	—
DS2-AS6	P2-00	2	1
DS2-BS	—	√	—
DS2-BS6	P2-00	2	1
DS2-BSW	—	√	—
DS2-BSW6	P2-00	2	1
DS3-PQA	P2-00	2	1
DS3E-PFA	P2-00	2	1
DS3 series	P0-10	2	1
DS3E series	P0-10	2	1

- **Electronic Gear Ratio of Servo Amplifier (Driving Unit) (Taking DS2 Series as an Example)**

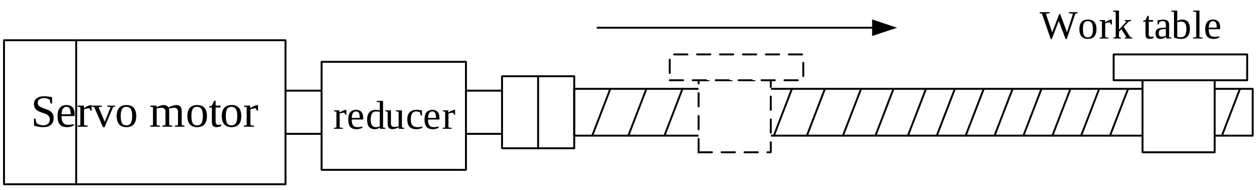
By using the electronic gear of the servo motor, the movement of each pulse can be set.
For the setting of electronic gears, please refer to the manual of servo driver, set values that are consistent with the use.

Example 1

The movement of each pulse is set to 10 μ m (when using mechanical screw).

Mechanical specifications

Servo driver	DS2 series
Rated Speed of Servo Motor	3000r/min
Ball screw lead pitch（Pb）	10mm
Reduction ratio of reducer（n）	1：5
Resolution of servo motor（Pt）	10000PLS/REV



f0：Instruction pulse frequency
CMX：Electronic gear/numerator
CDV：Electronic gear/denominator
NR：Servo motor speed r/min
X：Movement per pulse mm

The formula for calculating the ratio of electronic gears is as follows:

$$\frac{CMX}{CDV} = X \times \frac{Pt}{n \times Pb} = 10 \times 10^{-3} \times \frac{10000}{1/5 \times 10} = \frac{50}{1}$$

As can be seen from the figure above, the ratio of electronic gear of servo driver should be set to 50:1.
At this time, the rotation speed of the servo motor at the maximum output pulse frequency (200,000 Hz) of the basic unit is calculated as follows:

$$NR = \frac{CMX}{CDV} \times \frac{60}{Pt} \times f0$$

$$= \frac{50}{1} \times \frac{60}{10000} \times 200000$$

$$= 6000r/min > 3000r/min \text{（Rated speed）}$$

Note: Please set the maximum speed on the side of the programmable controller so that the rotation speed of the servo motor can be controlled below the rated speed.

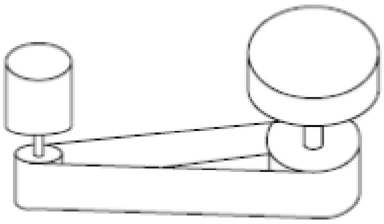
Example 2

The movement of each pulse is set to 0.01 degree (turntable).

Mechanical specifications

Servo driver	DS2 series
Servo motor rated speed	3000r/min
Turn table angle	360°/REV
Reduction ratio（n）	1：5
Servo motor resolution（Pt）	10000PLS/REV

Servo motor
Pt=10000[PLS/REV]



Synchronous belt：1:5

F0： Instruction pulse frequency[Hz]（Collector open circuit）
CMX：Electronic gear（Instruction Pulse Multiplier numerator）
CDV：Electronic gear（Instruction Pulse Multiplier denominator）
NR： Servo motor speed [r/min]
X ： Movement per pulse[°]

The formula for calculating the ratio of electronic gears is as follows:

$$\frac{CMX}{CDV} = X \times \frac{Pt}{n \times 360} = 1 \times 10^{-2} \times \frac{10000}{1/5 \times 360} = \frac{25}{18}$$

As can be seen from the figure above, the ratio of electronic gear of servo driver should be set to 25:1.

At this time, the rotation speed of the servo motor at the maximum output pulse frequency (200,000 Hz) of the basic unit is calculated as follows:

$$NR = \frac{CMX}{CDV} \times \frac{60}{Pt} \times f0$$

$$= \frac{25}{18} \times \frac{60}{10000} \times 100000$$

$$= 833.33r/min < 3000r/min \text{ (Rated speed)}$$

Because the rotating speed of the servo motor is below the rated speed, the maximum speed of the programmable controller side does not need to be limited.

- **Ready signal of servo driver（take DS2 as an example）**

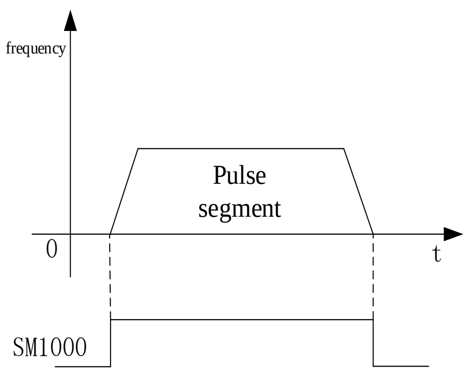
DS2 series servo enabling signal effectively represents the electrification of the servo motor. When the servo enabling signal is invalid, the motor does not operate.

Series name	Parameter	Setting value
DS2 series	P5-10	0010

1-4-5. Pulse sending complete flag notes

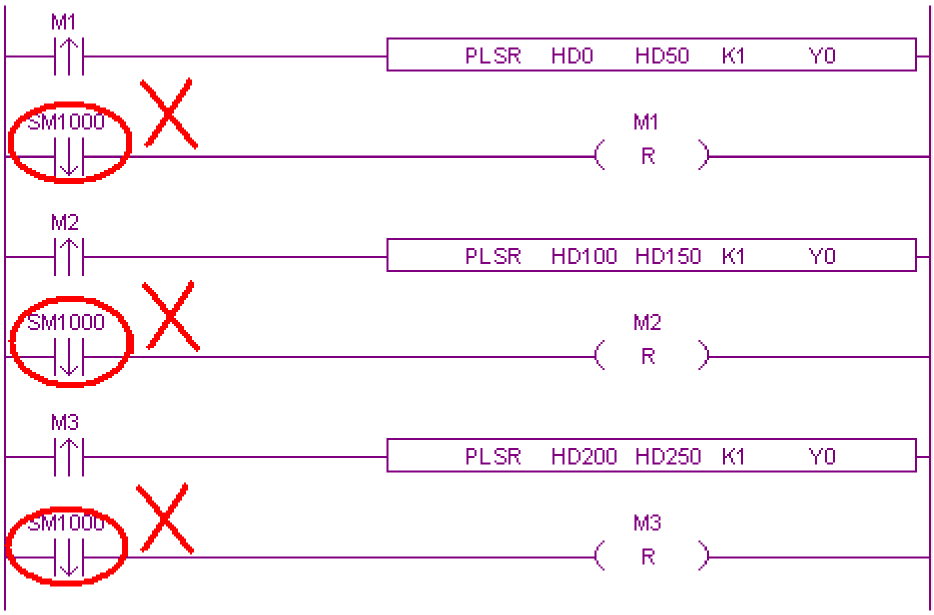
When the pulse sending flag SM1000, SM1020, SM1040 are changed from ON to OFF, it means that the action of instruction (pulse output action, etc.) is over. However, it does not mean that the action of the servo motor is over. In order to accurately grasp the end of the servo motor's operation, please correctly use the pulse sending flag.

Pulse sending flag:

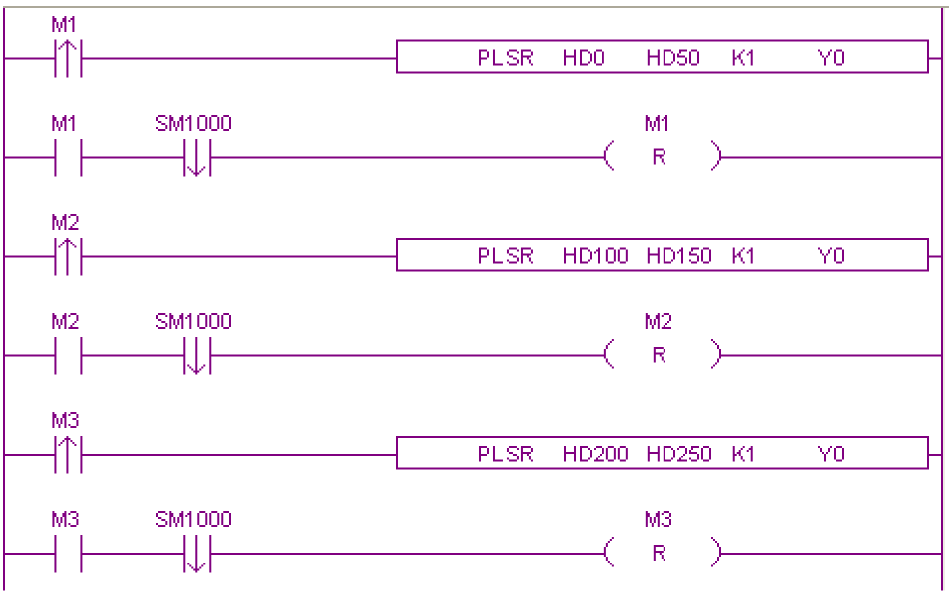
Flag	Axis	Explanation
SM1000	PULSE_1	When the pulse is sending, the coil is ON, and the OFF is set immediately after the pulse is sent. The falling edge of the coil is used to judge whether the pulse is sent or not.
SM1020	PULSE_2	
SM1040	PULSE_3	
SM1060	PULSE_4	
SM1080	PULSE_5	
SM1100	PULSE_6	
SM1120	PULSE_7	
SM1140	PULSE_8	
SM1160	PULSE_9	
SM1180	PULSE_10	

If multiple positioning instructions for the same pulse output port are written, then when the instructions are executed, the pulse flag SM1000, SM1020, SM1040 will change beween ON and OFF as each instructions. Therefore, if multiple instructions are executed, the sending pulse flag SM1000, SM1020, SM1040... are used in the same program at the same time, it is impossible to judge which instruction is executed, and at the same time, it is impossible to obtain the flag supported by each instruction.

Wrong writing is as below:



Correct writing is as below:



1-4-6. Cautions for triggering conditions of positioning instructions

XD/XL series of PLC positioning instructions are mainly PLSR (edge trigger), PLSF (normal open/close trigger), DRVI (edge trigger), DRVA (edge trigger), ZRN (edge trigger). Except PLSF instruction, all the other pulse instructions are edge trigger. In the process of executing a positioning instruction, the same pulse output port (such as Y0) is sending pulse, flag bit (SM1000) is always ON. The PLC will not respond to the pulse instruction triggered at the same pulse output port until the pulse output instructions being executed are sent out and the signal bit being sent is reset.

Since the conduction condition of PLSF pulse instruction is normally open/closed, when PLSF instruction is used, the conduction condition of PLSF instruction should be reset immediately when the pulse does not need to be executed (do not only set the pulse output frequency to 0 Hz, but not reset the pulse conduction condition).

1-4-7. Positioning Instruction and System Parameter Block Related Parameters

The following table sorts out the parameters setting of pulse output instruction and system parameter block:

System parameter	PLSR	PLSF	DRVI	DRVA	ZRN
Common parameter—pulse direction logic	**Must set**	**Must set**	×	×	**Must set**
Common parameter—enable soft limit	May not set	May not set	×	×	May not set
Common parameter — Default direction of mechanical return to origin	×	×	×	×	**Must set**
Common parameter —pulse unit	**Must set**	**Must set**	×	×	**Must set**
Common parameter — Interpolated coordinate mode	×	×	×	×	×
Common parameter — pulse send mode	**Must set**	**Must set**	×	×	**Must set**
Common parameter — pulse number(1 rotation)	May not set	May not set	×	×	May not set
Common parameter — offset(1 rotation)	May not set	May not set	×	×	May not set
Common parameter —pulse direction terminal	May not set	May not set	×	×	**Must set**
Common parameter —delay time of pulse direction	May not set	May not set	×	×	May not set
Common parameter —gear clearance positive compensation	May not set	May not set	×	×	May not set
Common parameter —gear clearance	May not	May not	×	×	May not

negative compensation	set	set			set
Common parameter —electric origin position	×	×	×	×	×
Common parameter —origin switch state setting	×	×	×	×	**Must set**
Common parameter —origin signal terminal setting	×	×	×	×	**Must set**
Common parameter —Z phase switch state setting	×	×	×	×	May not set
Common parameter — Z phase terminal setting	×	×	×	×	May not set
Common parameter —positive limit switch status setting	May not set	May not set	×	×	**Must set**
Common parameter —positive limit terminal setting	May not set	May not set	×	×	**Must set**
Common parameter —negative limit switch status setting	May not set	May not set	×	×	**Must set**
Common parameter —negative limit terminal setting	May not set	May not set	×	×	**Must set**
Common parameter —zero clear CLR signal output terminal setting	×	×	×	×	May not set
Common parameter —return speed VL	×	×	×	×	**Must set**
Common parameter —creeping speed VC	×	×	×	×	**Must set**
Common parameter — mechanical zero position	×	×	×	×	**Must set**
Common parameter — Z phase number	×	×	×	×	May not set
Common parameter —CLR signal delay time	×	×	×	×	May not set
Common parameter —grinding wheel radius(polar coordinate mode)	×	×	×	×	×
Common parameter — soft limit positive limit value					
Common parameter — soft limit negative limit value					
Group 1 parameter — pulse default speed	**Must set**	**Must set**	×	×	**Must set**
Group 1 parameter — acceleration time of pulse default speed	**Must set**	**Must set**	×	×	**Must set**
Group 1 parameter — deceleration time of pulse default speed	**Must set**	**Must set**	×	×	**Must set**

Group 1 parameter — Interval acceleration and deceleration time	May not set	May not set	×	×	May not set
Group 1 parameter —pulse acc/dec mode	**Must set**	**Must set**	×	×	**Must set**
Group 1 parameter —max speed	**Must set**	**Must set**	×	×	**Must set**
Group 1 parameter —start speed	**Must set**	**Must set**	×	×	**Must set**
Group 1 parameter —end speed	**Must set**	**Must set**	×	×	**Must set**

Note: group 2 to 4 parameters are same to group 1.

1-4-8. Troubleshooting of Servo Motor and Stepping Motor

When the servo motor and stepper motor do not work, please confirm the following items:

1) Please confirm the connection.

2) Please execute the positioning instructions to confirm the status of the following LED.

LED set as pulse output signal

LED set as pulse direction signal

3) Make sure that when the programmable controller executes the positioning instructions, the values of the accumulated pulse registers of each axis are changing.

The cumulative registers for each pulse output are shown in the following table:

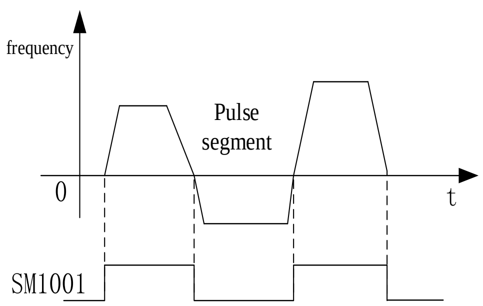
No.	Function	Notes	Axis
HSD0	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_1
HSD1	High 16-bit of cumulative pulse		
HSD2	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD3	High 16-bit of cumulative pulse		
HSD4	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_2
HSD5	High 16-bit of cumulative pulse		
HSD6	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD7	High 16-bit of cumulative pulse		
HSD8	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_3
HSD9	High 16-bit of cumulative pulse		
HSD10	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD11	High 16-bit of cumulative pulse		
HSD12	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_4
HSD13	High 16-bit of cumulative pulse		
HSD14	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD15	High 16-bit of cumulative pulse		
HSD16	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_5
HSD17	High 16-bit of cumulative pulse		
HSD18	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD19	High 16-bit of cumulative pulse		

HSD20	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_6
HSD21	High 16-bit of cumulative pulse		
HSD22	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD23	High 16-bit of cumulative pulse		
HSD24	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_7
HSD25	High 16-bit of cumulative pulse		
HSD26	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD27	High 16-bit of cumulative pulse		
HSD28	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_8
HSD29	High 16-bit of cumulative pulse		
HSD30	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD31	High 16-bit of cumulative pulse		
HSD32	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_9
HSD33	High 16-bit of cumulative pulse		
HSD34	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD35	High 16-bit of cumulative pulse		
HSD36	Low 16-bit of cumulative pulse	Pulse number is the unit	PULSE_10
HSD37	High 16-bit of cumulative pulse		
HSD38	Low 16-bit of cumulative pulse	Pulse equivalent is the unit	
HSD39	High 16-bit of cumulative pulse		

4) Make sure that the pulse output form of the programmable controller side and the servo amplifier (driving unit) is consistent.

5) Make sure that the stop bit of the pulse output is in action.

The pulse output flags of each pulse are shown in the table below.

No.	Coil	Axis	Note
1	SM1001	PULSE_1	When the pulse value is positive, the coil is ON; when the pulse value is negative, the coil is OFF. 
2	SM1021	PULSE_2	
3	SM1041	PULSE_3	
4	SM1061	PULSE_4	
5	SM1081	PULSE_5	
6	SM1101	PULSE_6	
7	SM1121	PULSE_7	
8	SM1141	PULSE_8	
9	SM1161	PULSE_9	
10	SM1181	PULSE_10	

6) Please confirm whether the limit (positive limit and reverse limit) is in action.

7) Please confirm the action sequence of positioning instruction.

When the pulse flag bit is ON, the positioning instruction or the pulse output instruction using the same output terminal can not be executed.

1-4-9. Troubleshooting of incorrect stop position of servo motor and stepper motor

When the stop position is incorrect, please confirm the following items:
1) Make sure that the setting of the electronic gear of the servo amplifier (driving unit) is correct.
2) Please confirm whether the origin position is offset.

A. When designing the origin signal, consider that there is enough time for ON to slow down to crawling speed.

The ZRN instruction begins to decelerate to stop at the front end of the origin, delays and reverse accelerates to crawl speed, stops when it leaves the origin, and clears the current value register. Failure to slow down to crawl speed in front of the back end of the origin will cause stop position offset.

B. Please make the crawling speed slow enough.

The stop of the origin regression instruction is not decelerated, so if the crawling speed is too fast, the stop position will be offset due to inertia.

C. Soft components for origin signals.

The origin signal terminal can select all the input points on the PLC; but if the selected input point is the external interrupt terminal on the PLC main unit, the process of returning to the mechanical origin will be handled according to the interrupt, which can further improve the accuracy of returning to the mechanical origin (if Z phase is used to return to the origin, it will not affect); and the selected input point is the external interrupt terminal on PLC extention module, in the process of mechanical origin, it will be affected by the scanning cycle of PLC (if Z phase is used to return to the origin, it will not be affected).

3) After the forward and reverse rotation (round-trip action), the stop position deviates.

Because of the contact gap between the worktable and the ball screw, when the worktable switches from the forward movement to the reverse movement, the reverse actual movement distance is less than the set distance; when the worktable switches from the reverse movement to the forward movement, the forward actual movement distance is less than the set distance.

It can be corrected by forward gear clearance compensation and reverse gear clearance compensation.

1-5. Positioning instruction example programs

This section mainly introduces the use of PLSR, PLSF, DRVA, DRVI, ZRN instructions through several sample programs.

Action	Instruction	Program example	
		Sequential ladder chart	Process ladder chart
Multi section pulse positioning	PLSR	1-5-4 1-5-6	1-5-5 1-5-7
Variable frequency pulse output	PLSF	1-5-2 1-5-4	1-5-3 1-5-5
Relative single section positioning	DRVI	1-5-2 1-5-6	1-5-3 1-5-7
Absolute single section positioning	DRVA	1-5-2 1-5-6	1-5-3 1-5-7
Mechanical origin regression	ZRN	1-5-2 1-5-4 1-5-6	1-5-3 1-5-5 1-5-7

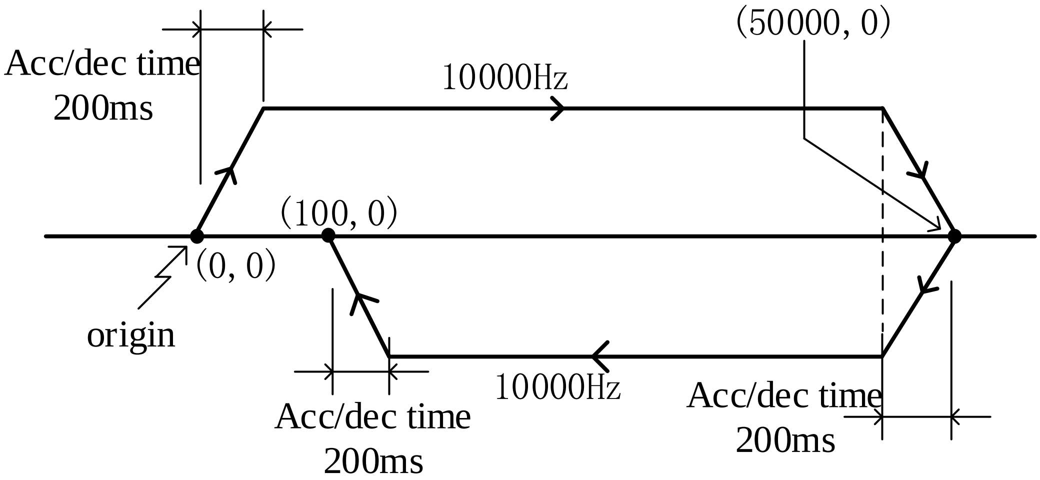
1-5-1. I/O point assignment

The pulse output Y0 (axis 1) is used in the program example. When using other pulse output terminals, please modify the corresponding soft components of the pulse axis.

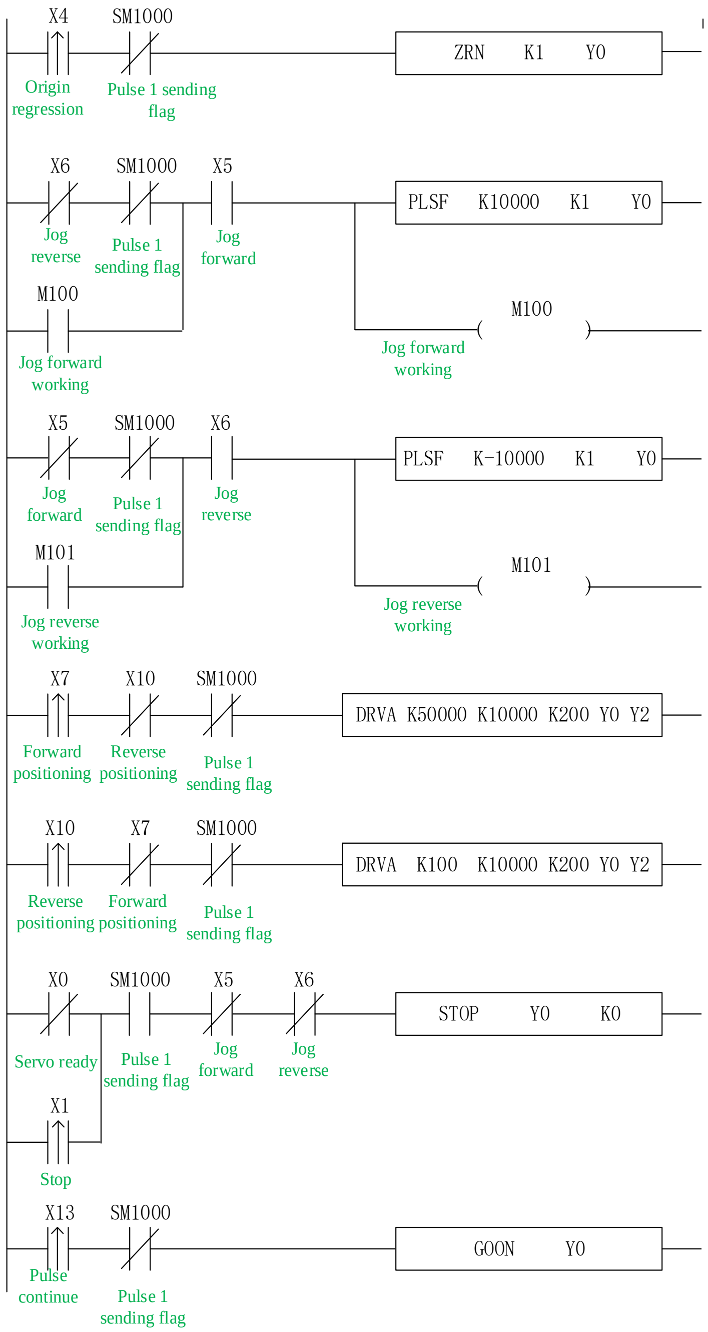
Signal name	I/O points	Notes
Pulse output port	Y0	
Pulse direction port	Y2	
CLR zero clear signal	Y3	
Servo ready	X0	
Stop	X1	
Pulse continue	X13	
Origin regression	X4	
Jog forward	X5	
Jog reverse	X6	
Forward rotation positioning	X7	
Reverse rotation positioning	X10	
Close origin input terminal	X2	
Origin input terminal	X3	External interruption terminal
Forward limit switch	X11	
Reverse limit switch	X12	

1-5-2. Forward and reverse rotation sequence control sample program 【PLSF, DRVI, DRVA, ZRN】

Example 1: According to the following figure, use the absolute single section positioning method.



Firstly, the ladder chart program is shown as follows:



//mechanical origin regression (use group 1 parameters)

//jog forward control, speed is 10000Hz (use group 1 parameters)

//jog reverse control, speed is -10000Hz (use group 1 parameters)

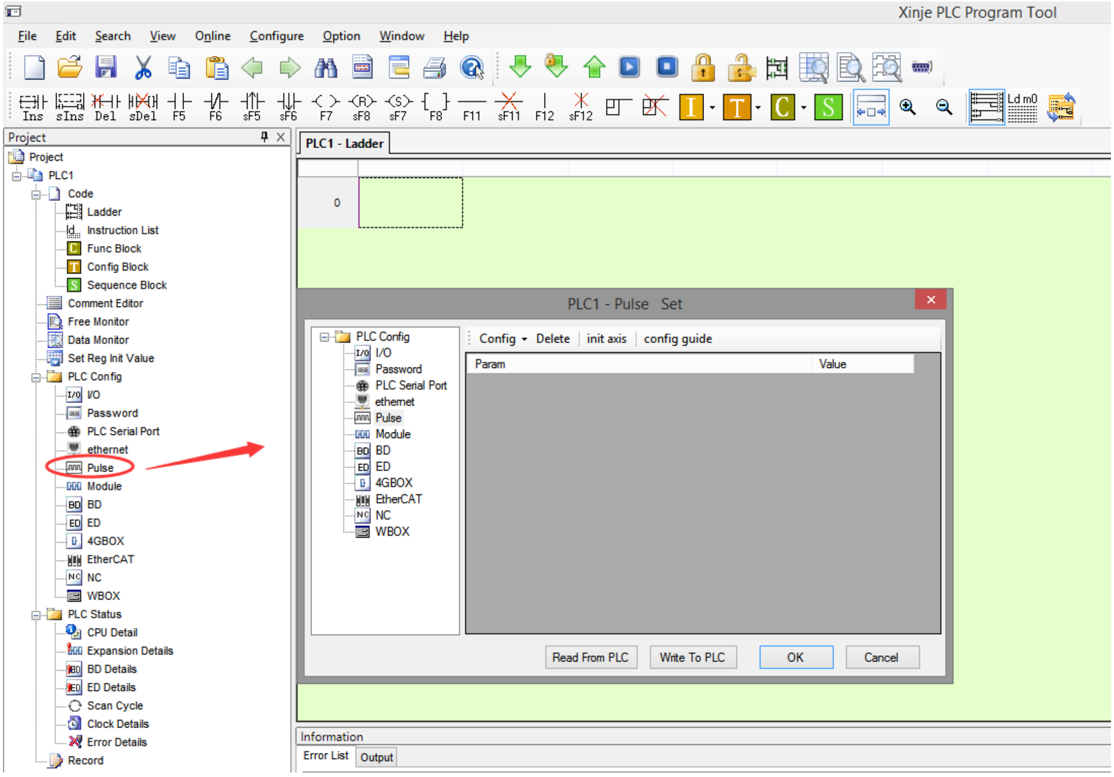
//forward rotation positioning control

//reverse rotation positioning control

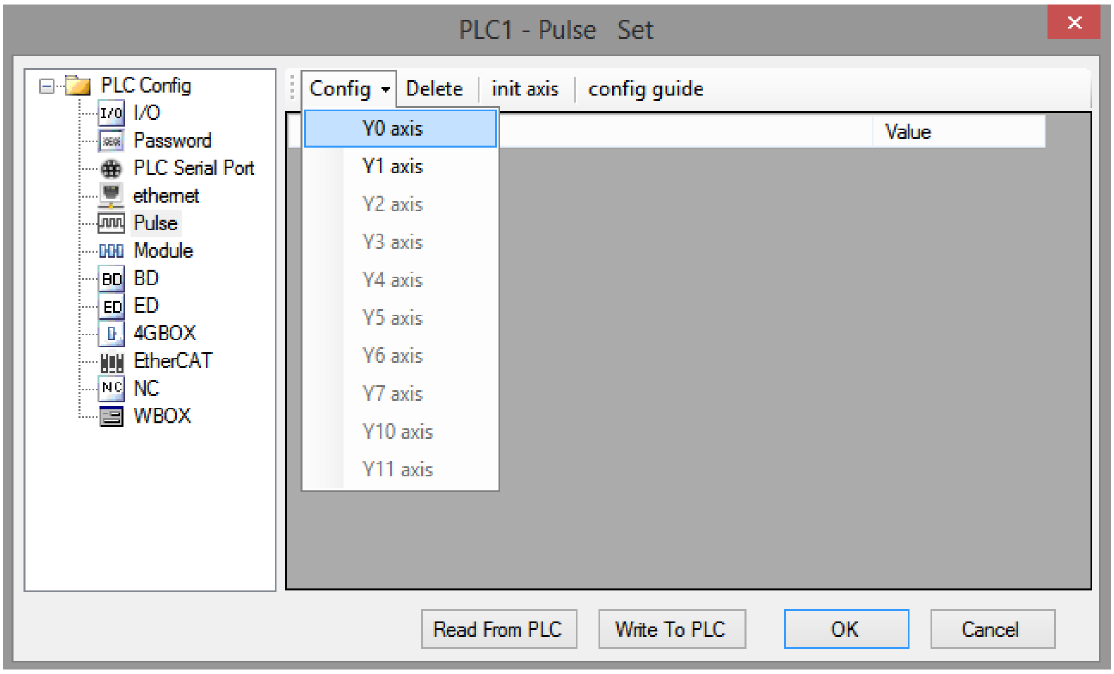
//pulse stop sending (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

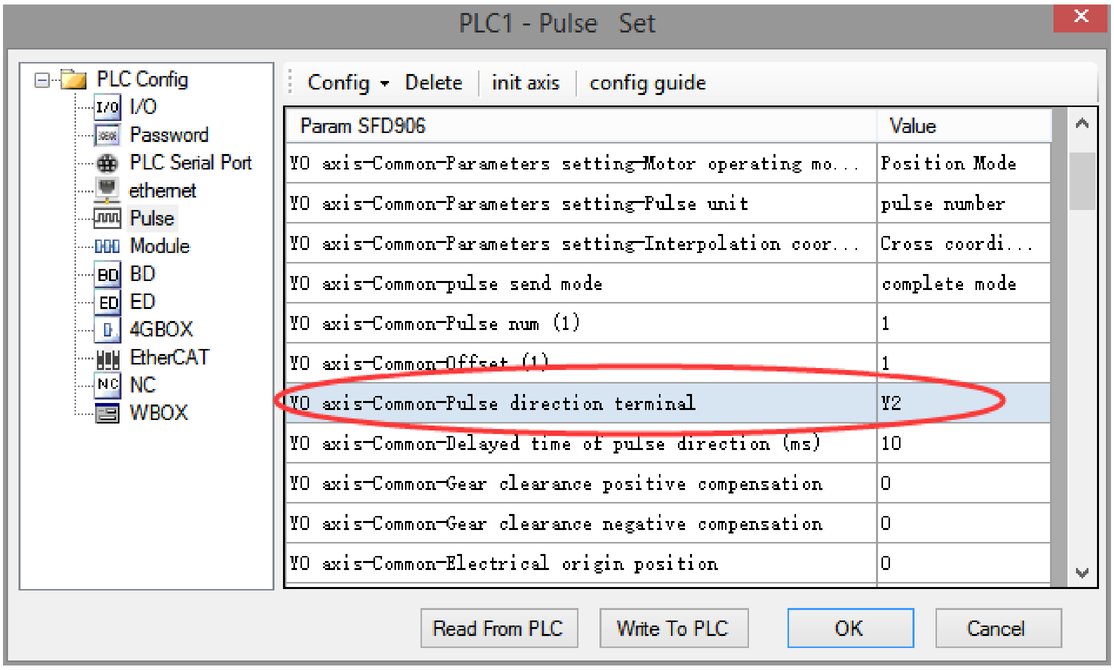
In the sample program, except DRVI and DRVA, all the system parameters used in the pulse instructions are group 1 parameters. So we click the "pulse configuration parameters" in the PLC programming software, as follows:

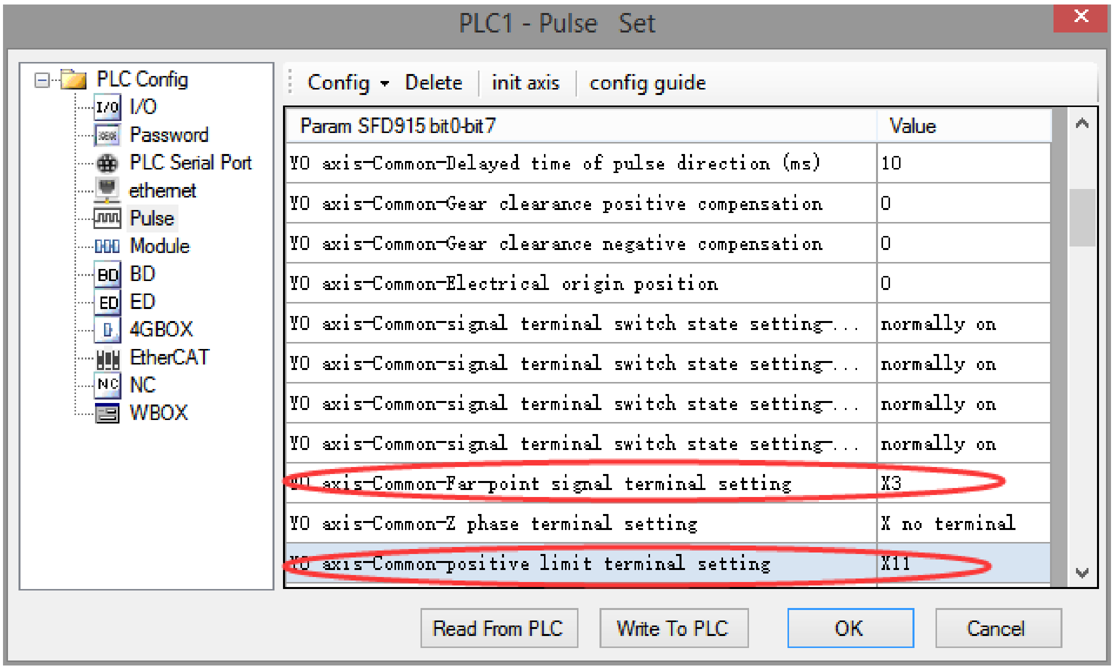


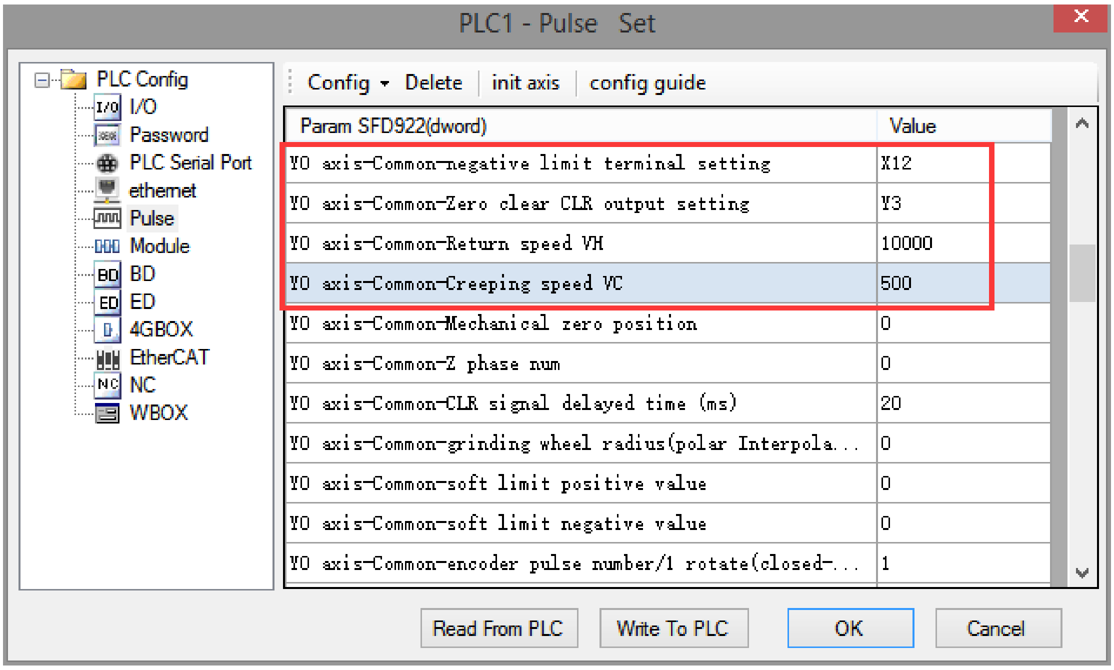
Click config, then select Y0 axis.

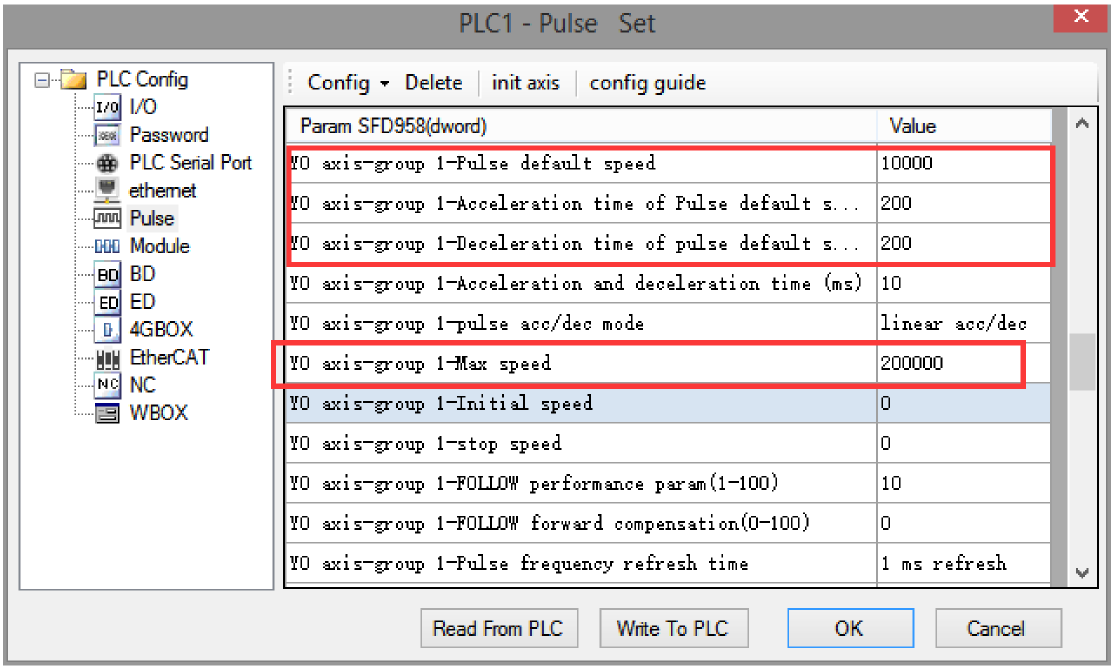


In the parameter configuration table, configure as follows (circled parameters need to be modified):





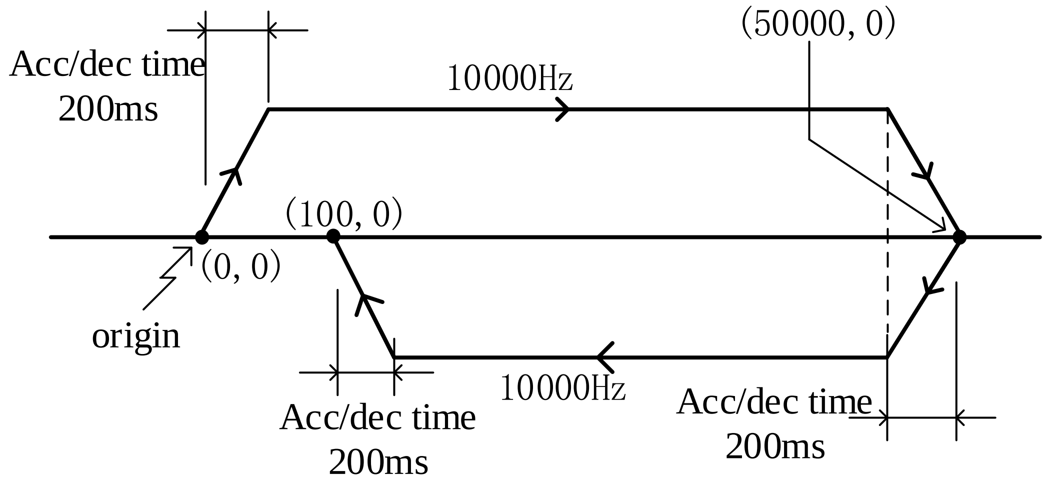




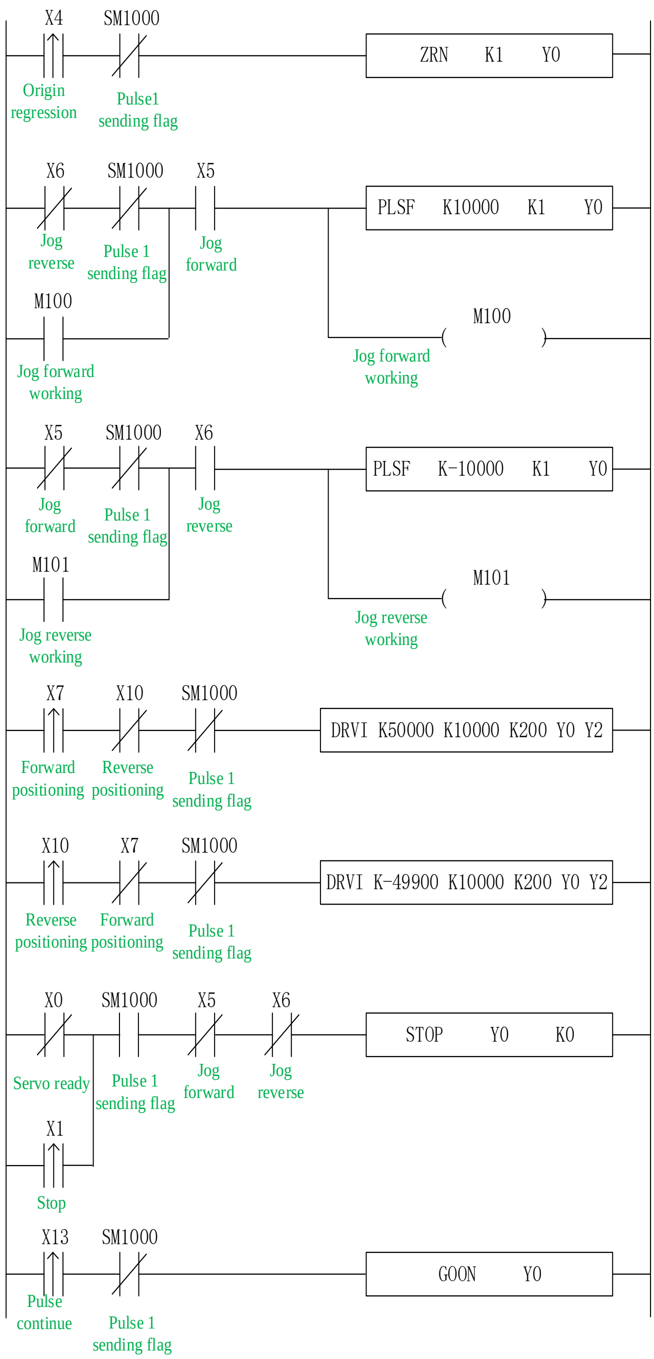
After configuring the parameters, click the "Write to PLC" button to write the parameters into the PLC. After downloading the program, power off the PLC and then power on again.

Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

Example 2: According to the following figure, use the relative single segment positioning method.



Firstly, make the ladder chart as follows:



//mechanical origin regression (use group 1 parameters)

//jog forward control, speed is 10000Hz (use group 1 parameters)

//jog reverse control, speed is -10000Hz (use group 1 parameters)

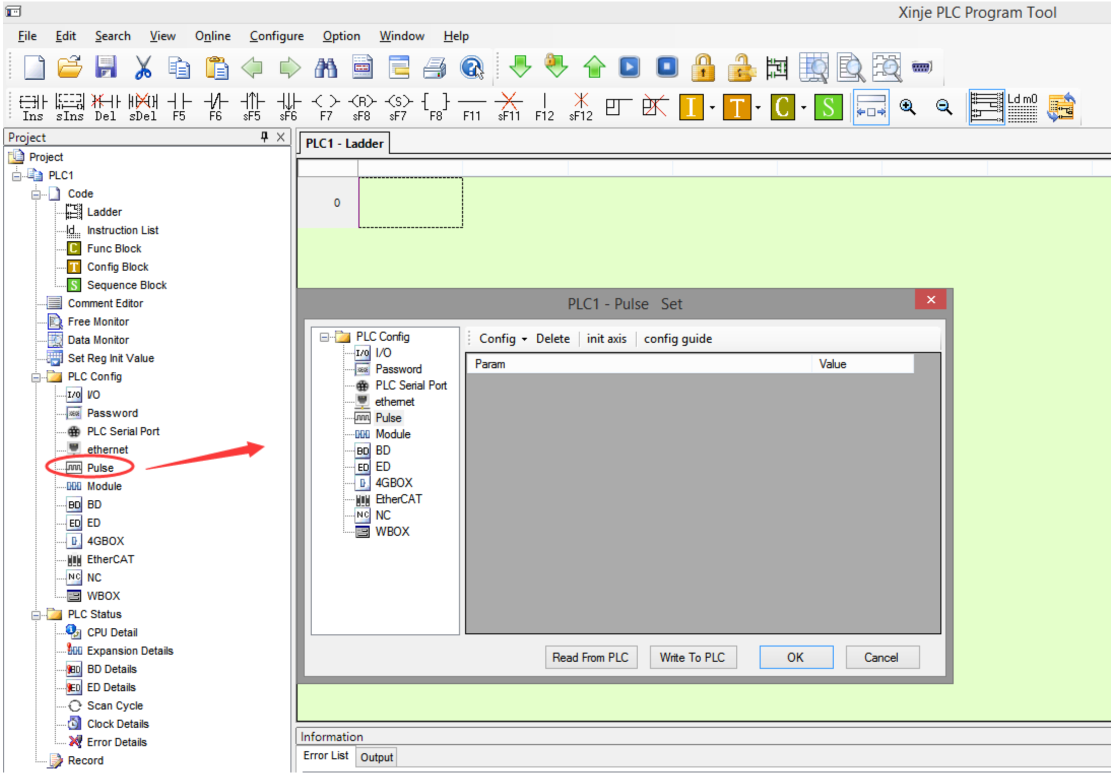
//forward rotation positioning control

//reverse rotation positioning control

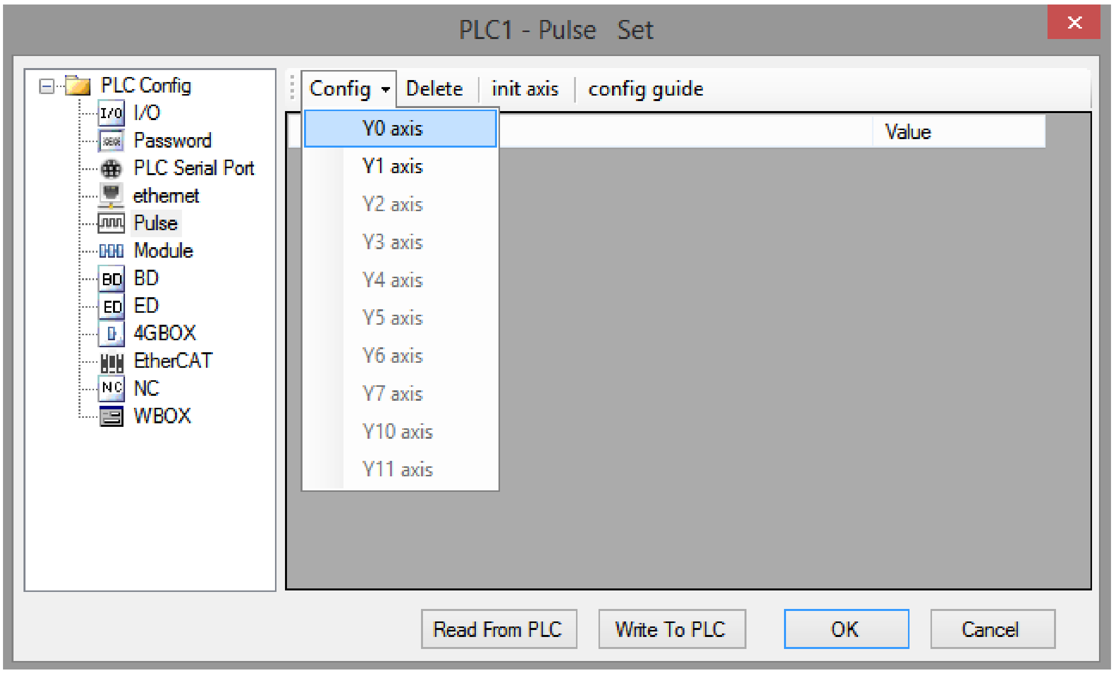
//pulse stop sending (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

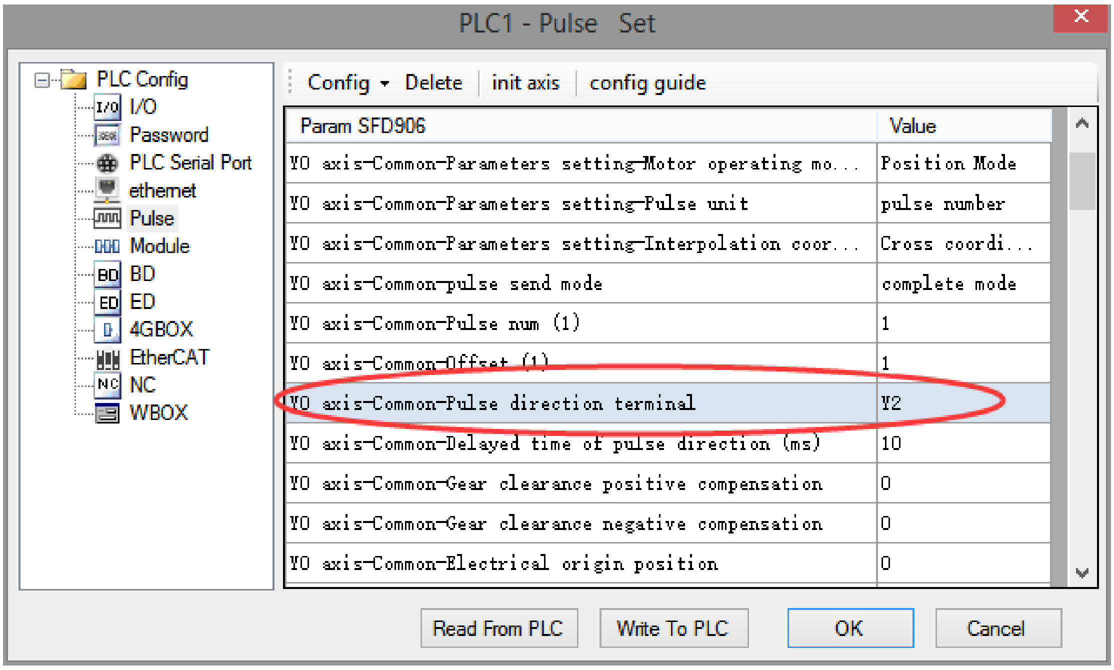
In the sample program, except DRVI and DRVA, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

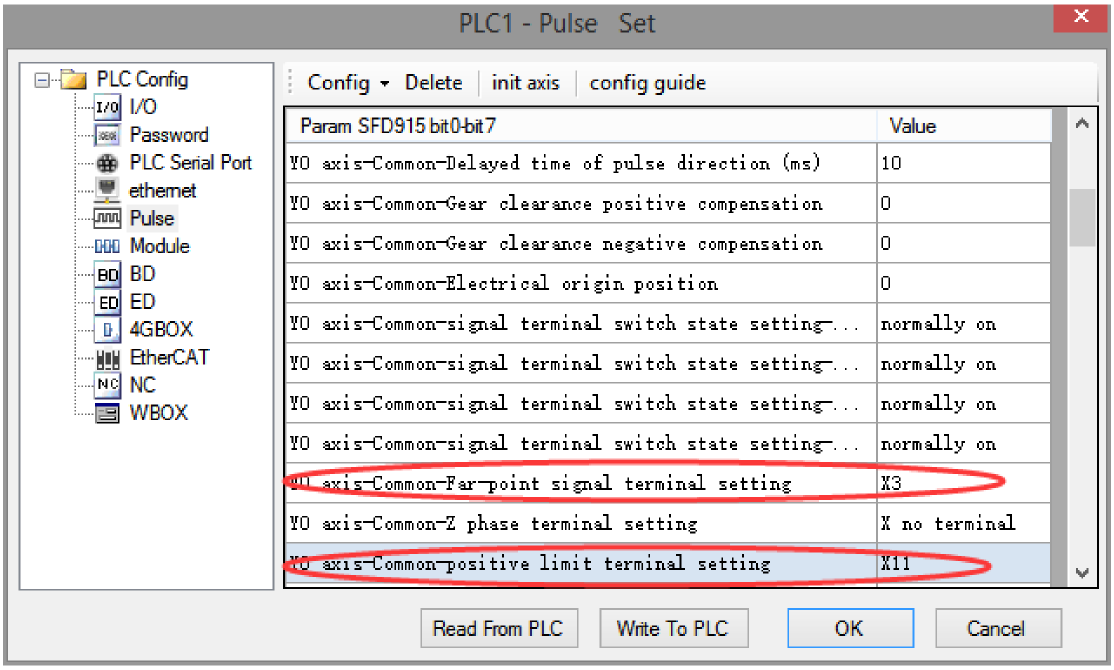


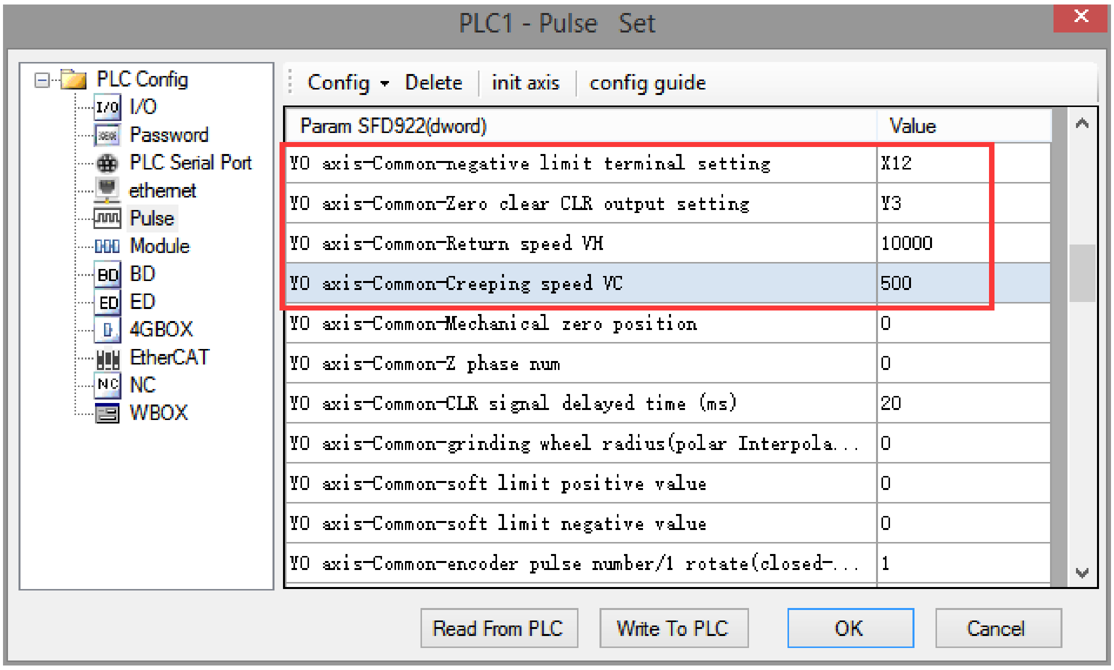
Click config, then select Y0 axis.

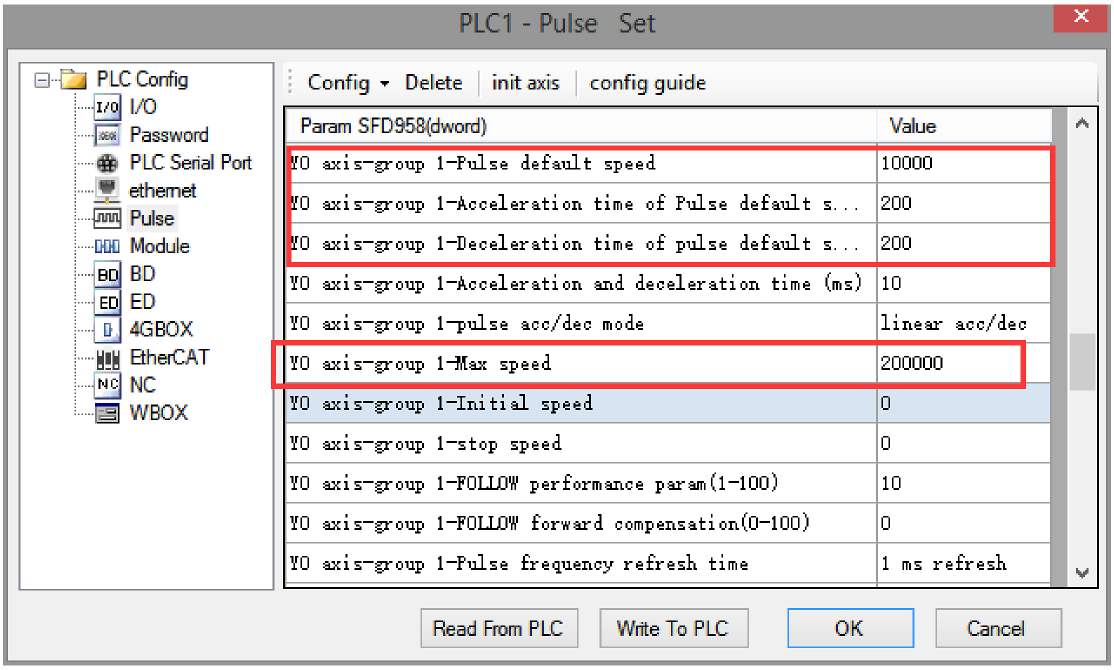


In the parameter configuration table, configure as follows (circled parameters need to be modified):







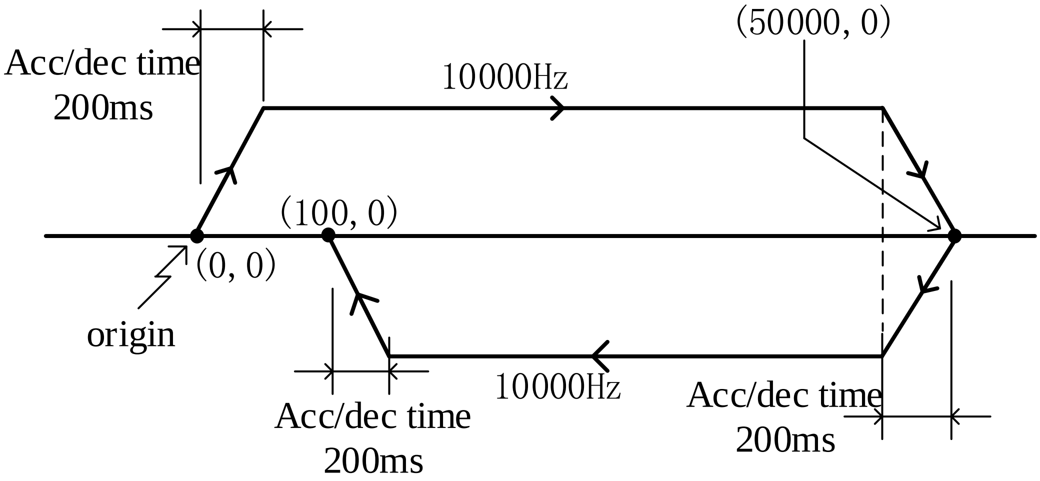


After configuring the parameters, click the "Write to PLC" button to write the parameters into the PLC. After downloading the program, power off the PLC and then power on again.

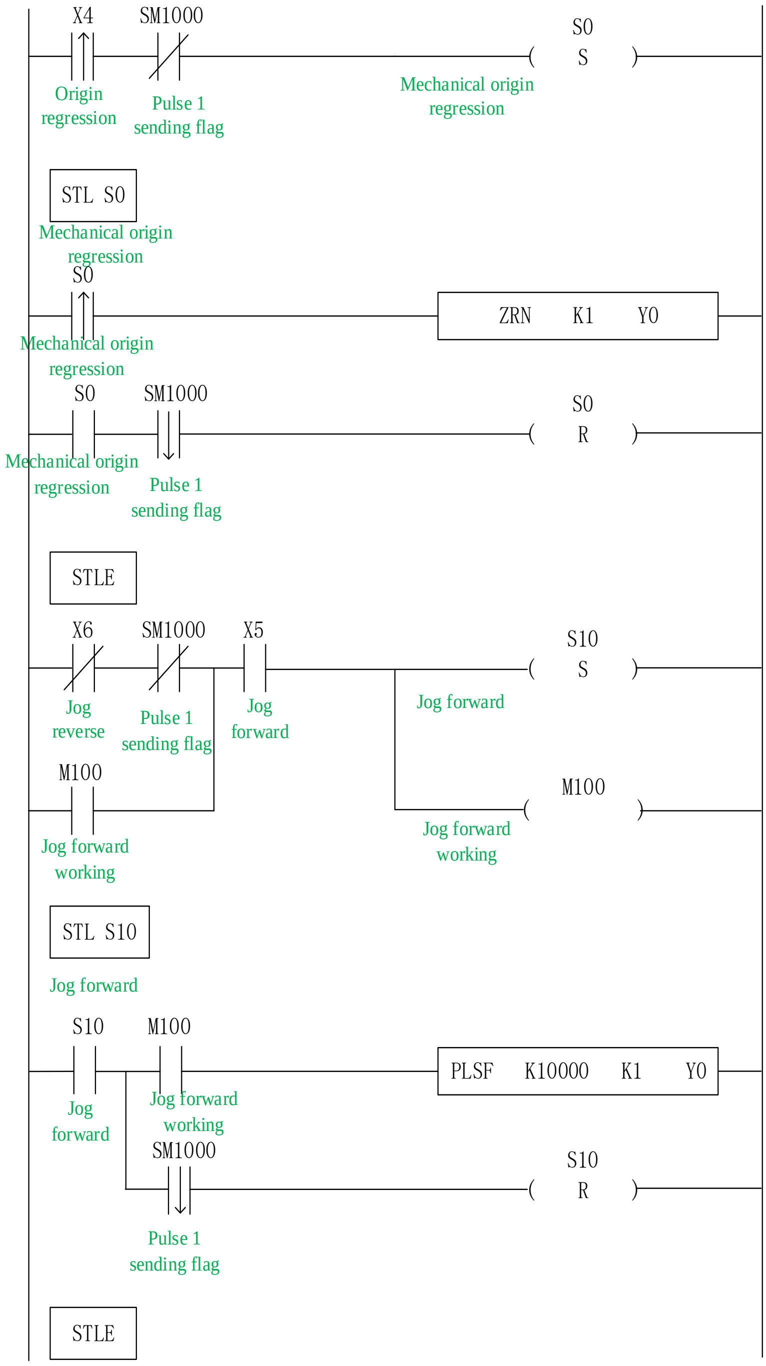
Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

1-5-3. Forward and reverse rotation process program 【PLSF, DRVI, DRVA, ZRN】

Example 1: According to the following figure, use the absolute single segment positioning method.

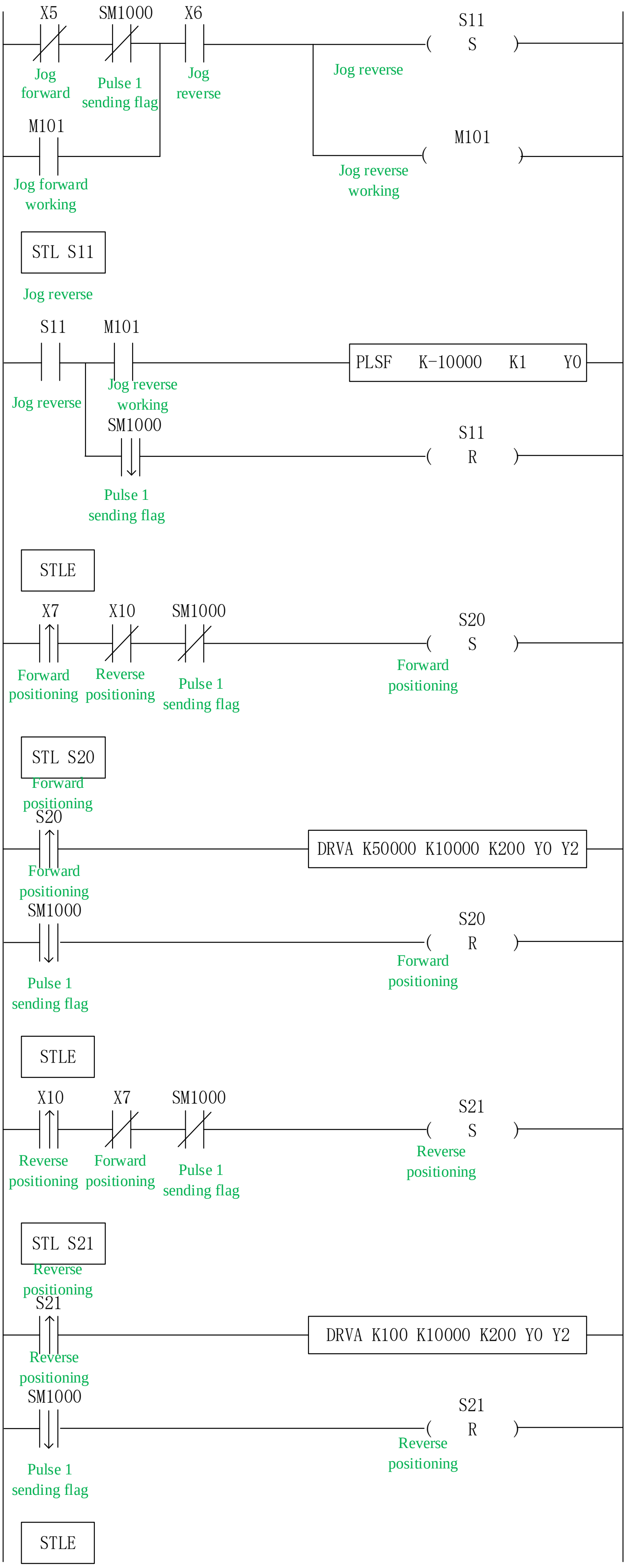


Firstly, make the ladder chart as follows:



//mechanical origin regression instruction (group 1 parameters)

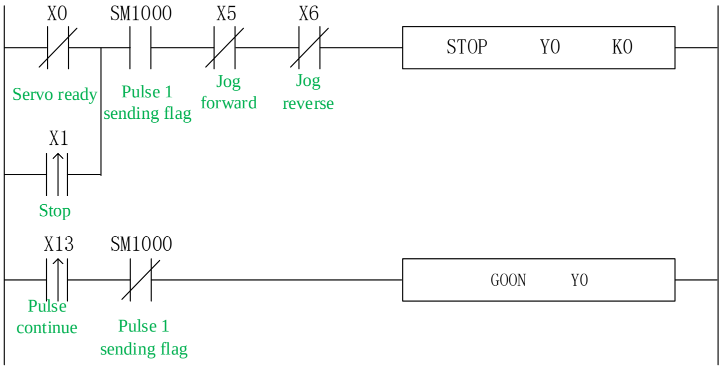
//jog forward control, speed is 10000Hz (group 1 parameters)



//jog reverse control, the speed is 10000Hz (group 1 parameters)

//forward positioning control

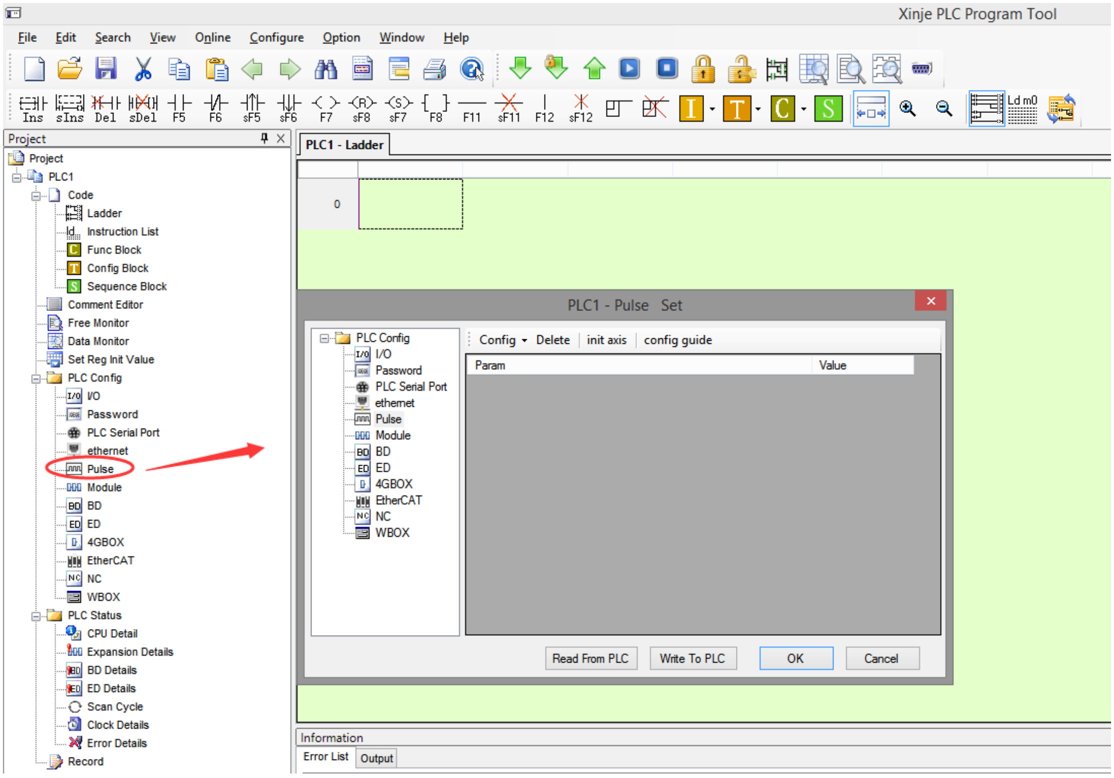
//reverse positioning control



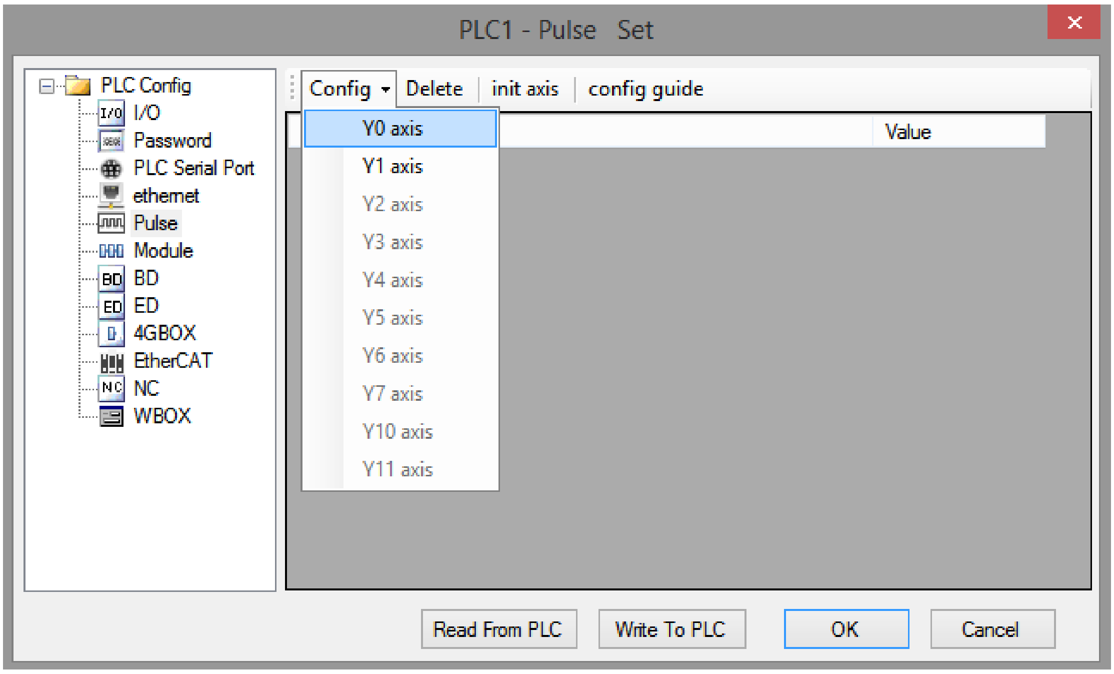
//pulse stop (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

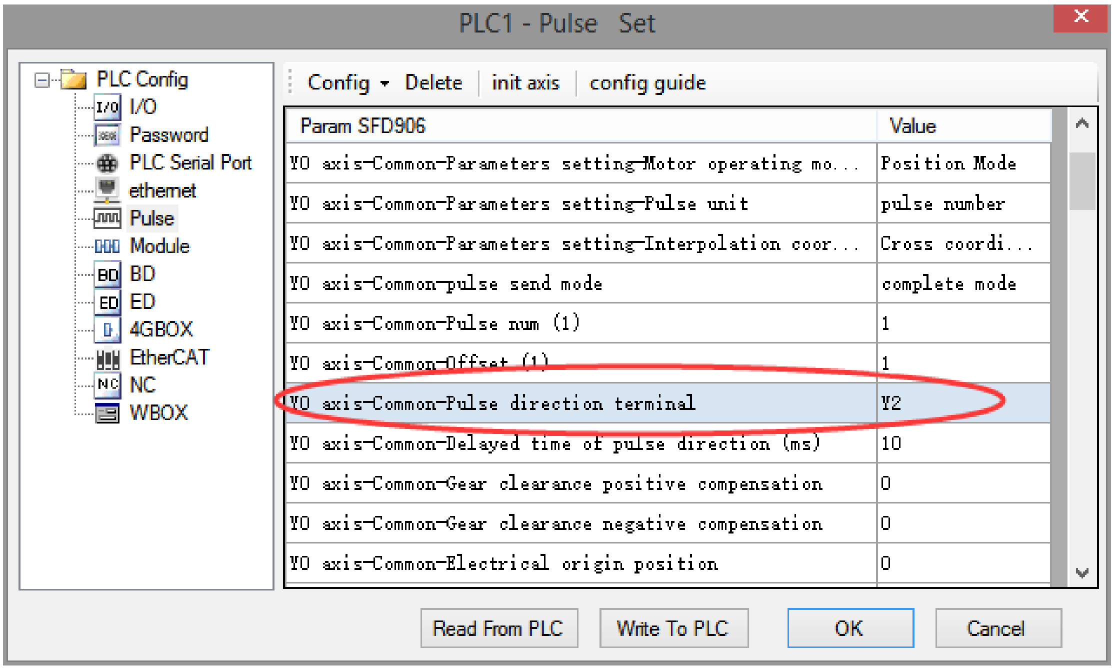
In the sample program, except DRVI and DRVA, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

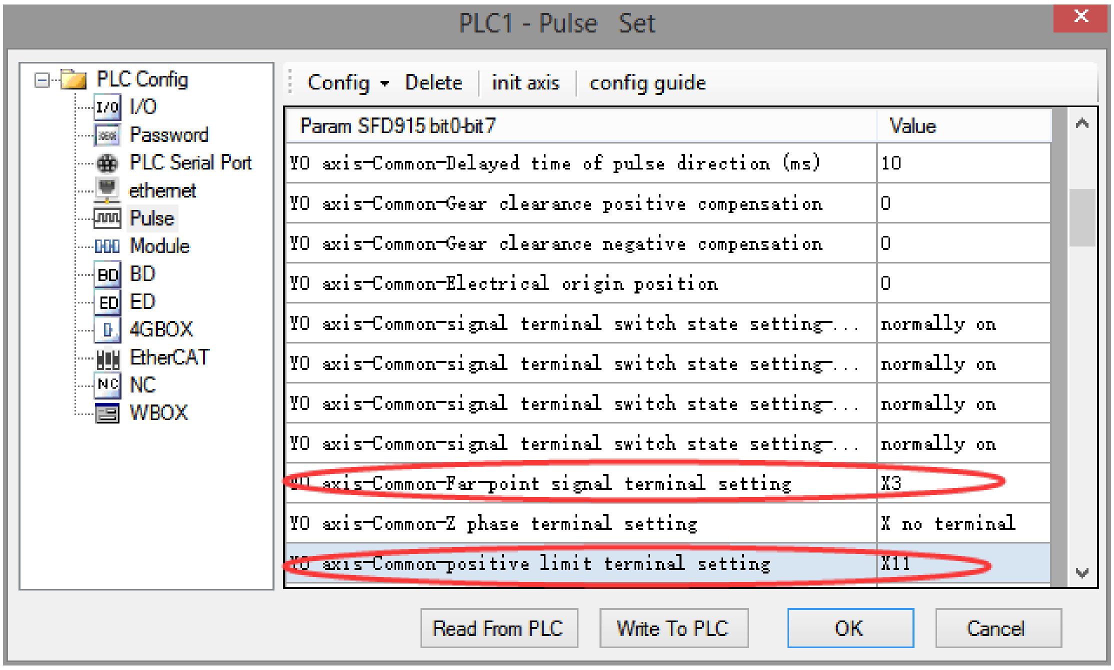


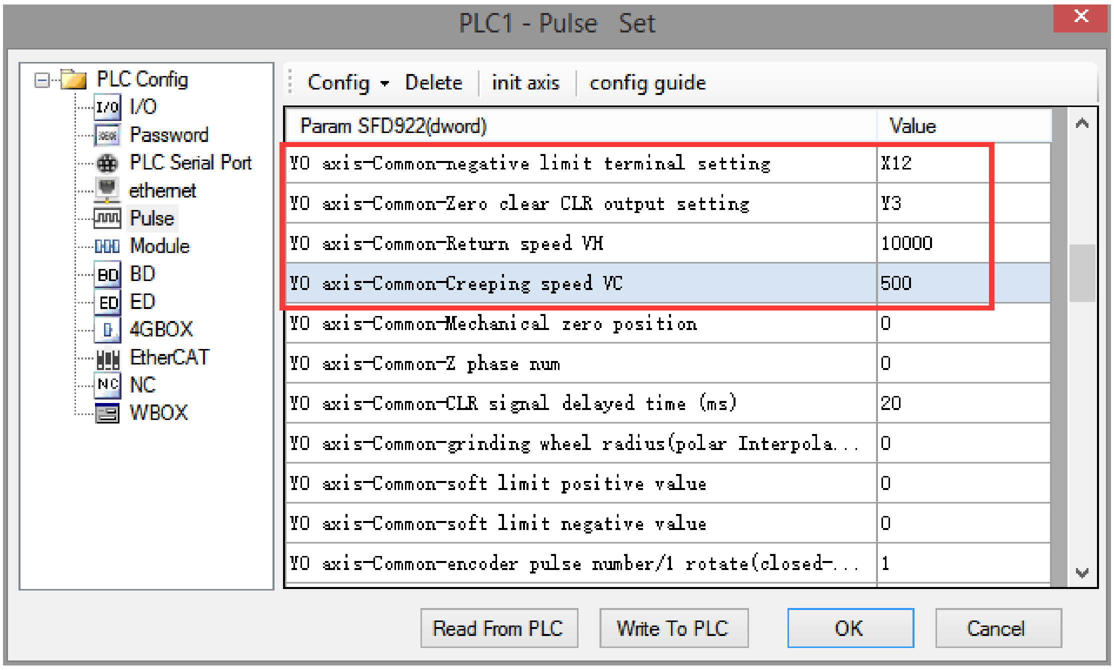
Click config, then select Y0 axis.

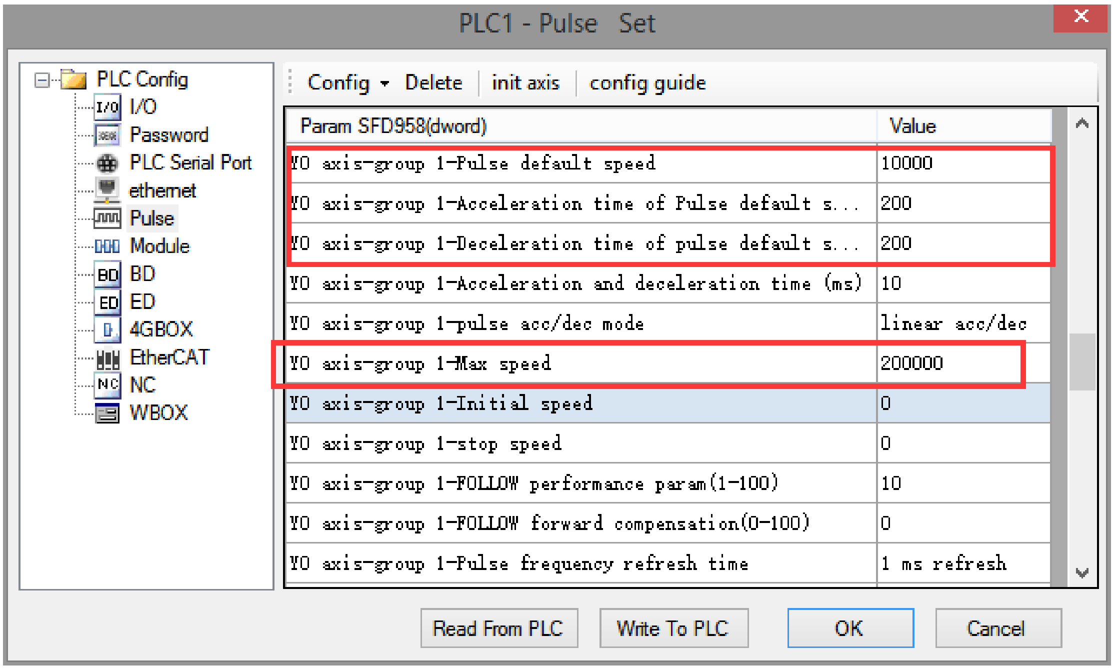


In the parameter configuration table, configure as follows (circled parameters need to be modified):




PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel

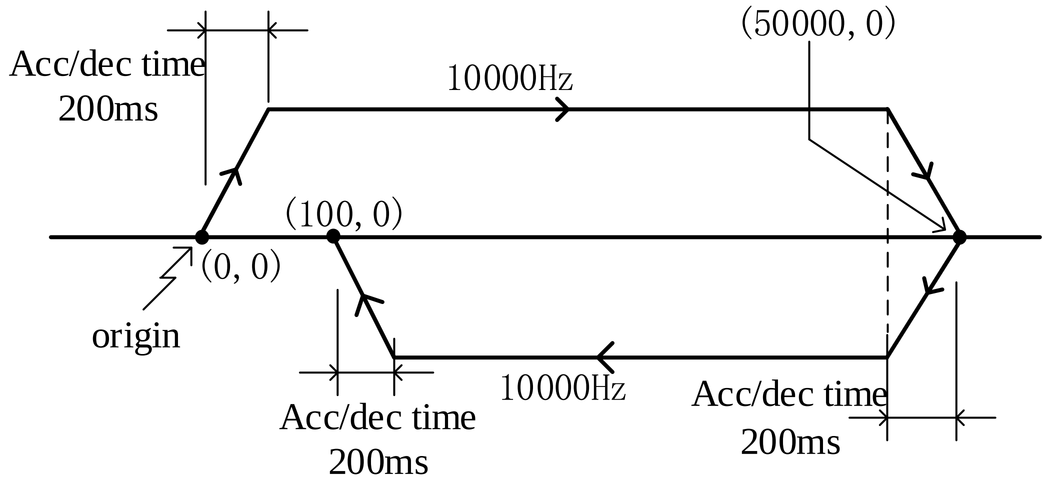

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD922(dword)
Value
Y0 axis-Common-negative limit terminal setting
X12
Y0 axis-Common-Zero clear CLR output setting
Y3
Y0 axis-Common-Return speed VH
10000
Y0 axis-Common-Creeping speed VC
500
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
20
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...
1
Read From PLC
Write To PLC
OK
Cancel



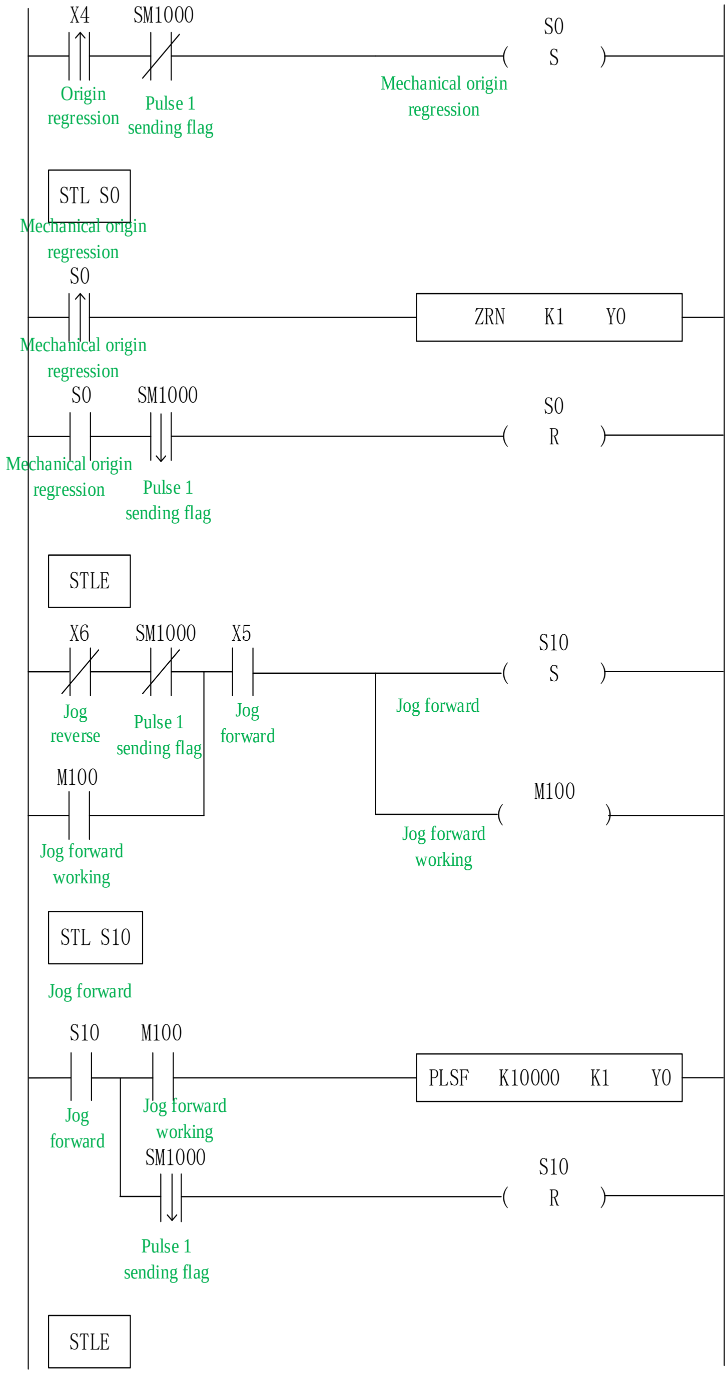
After configuring the parameters, click the "Write to PLC" button to write the parameters into the PLC. After downloading the program, power off the PLC and then power on again.

Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

Example 2: According to the following figure, use the relative single segment positioning method.

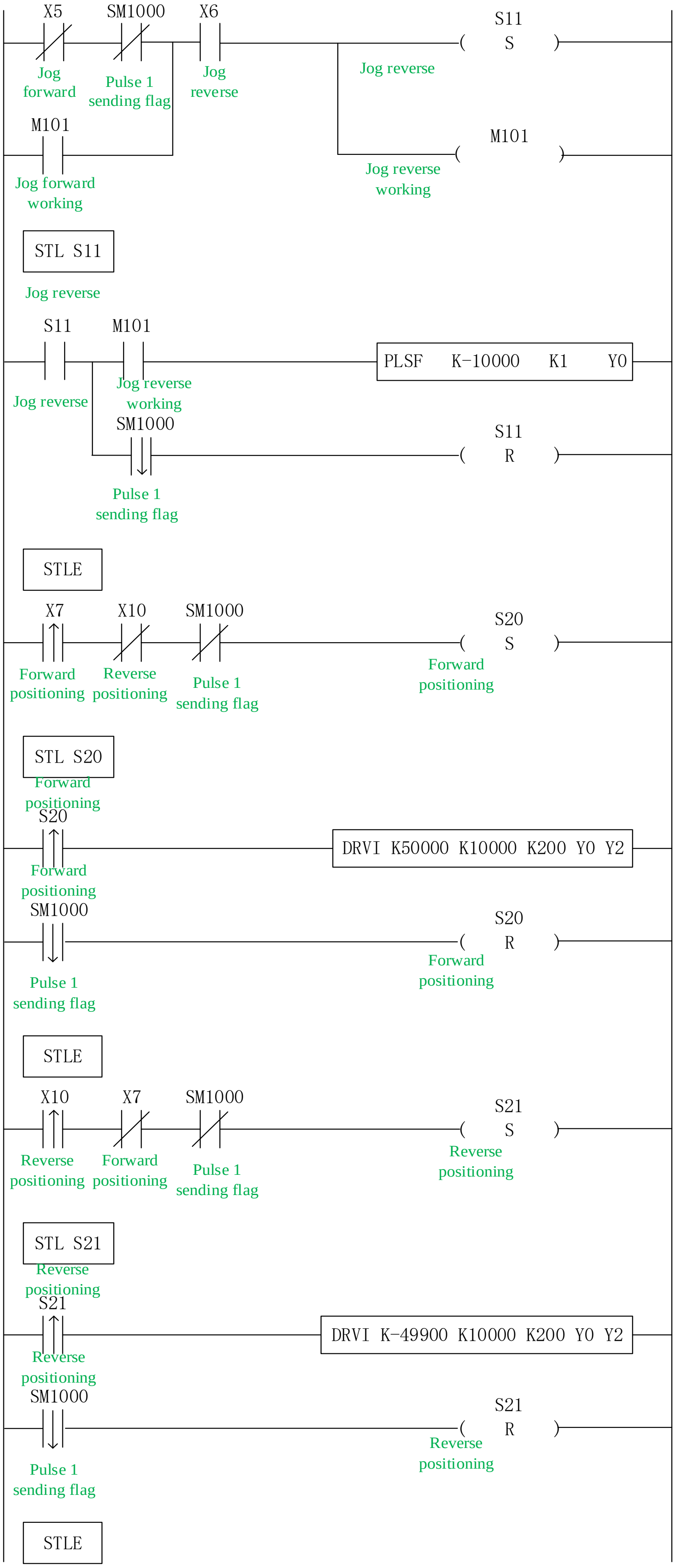


Firstly, make the ladder chart as follows:



//mechanical origin regression instruction (use group 1 parameters)

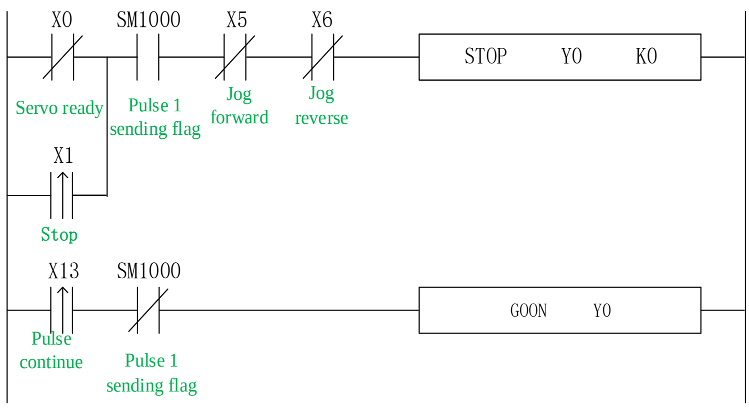
//jog forward control, the speed is 10000Hz (use group 1 parameters)



//jog reverse control, the speed is 10000Hz (use group 1 parameters)

//forward positioning control

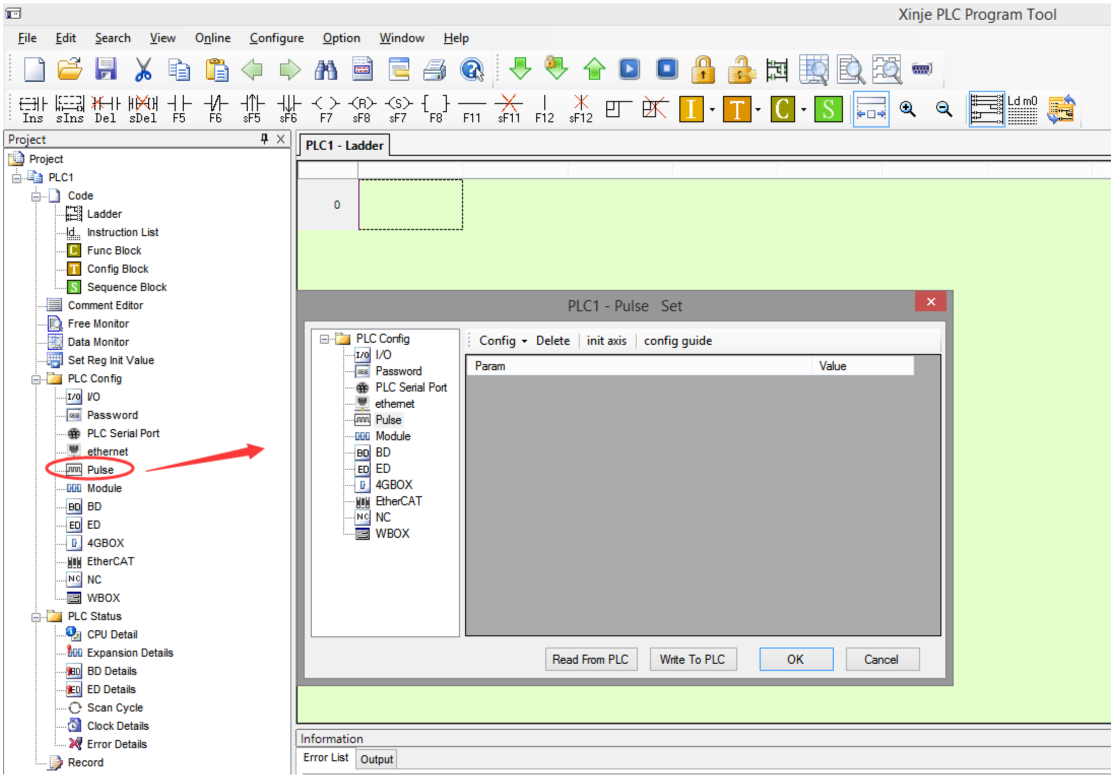
//reverse positioning control



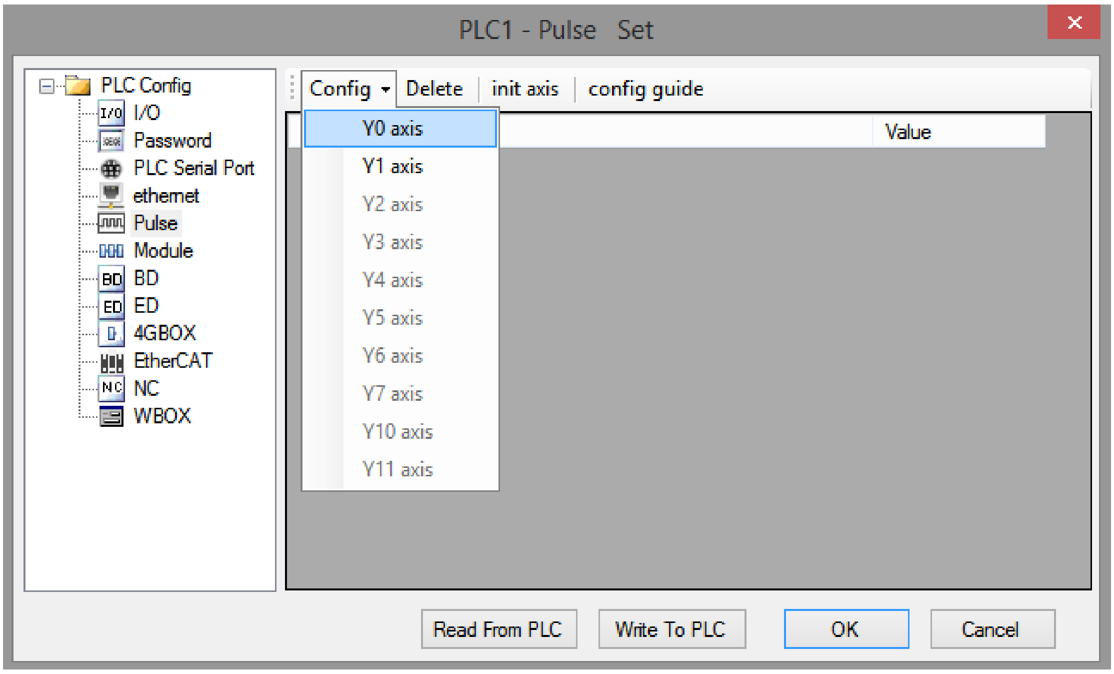
//pulse stop (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

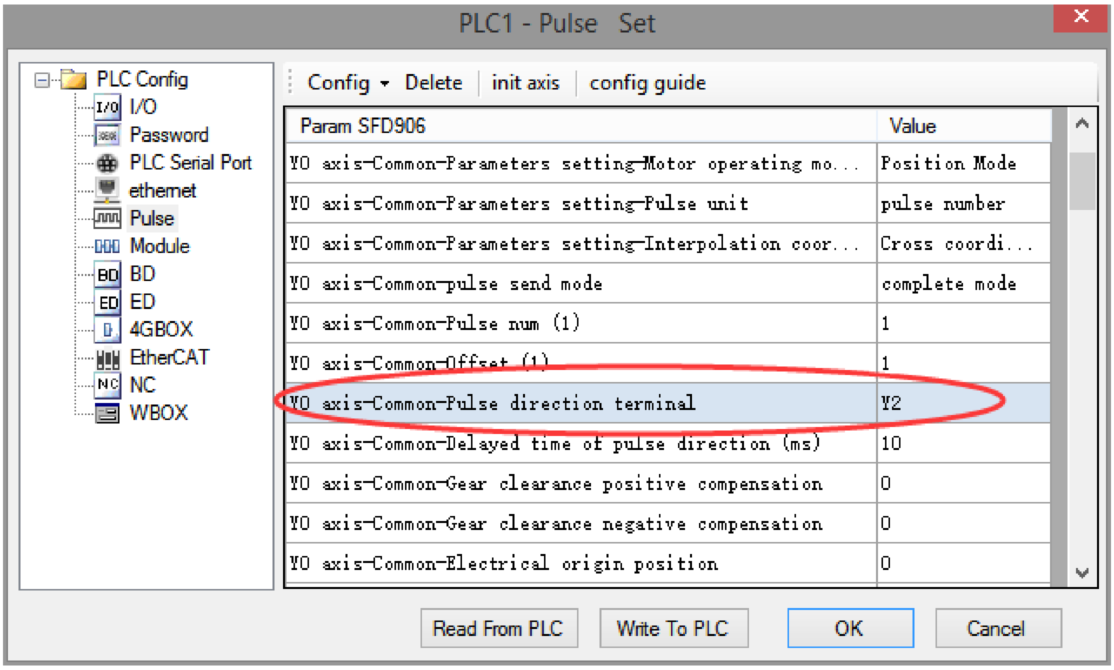
In the sample program, except DRVI and DRVA, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

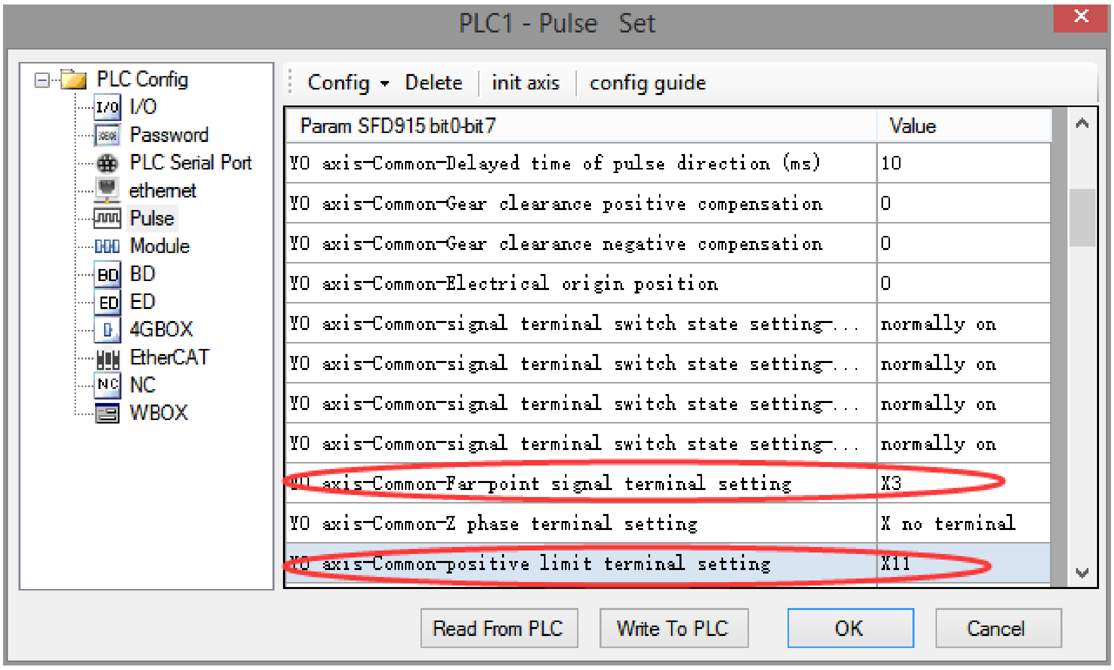


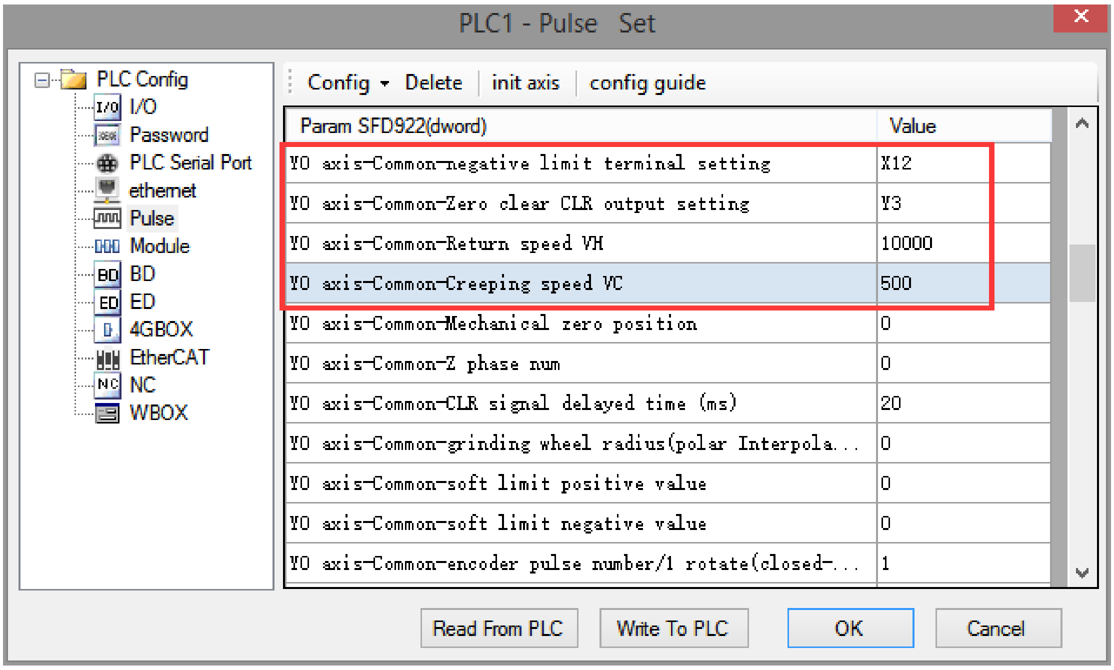
Click config, then select Y0 axis.

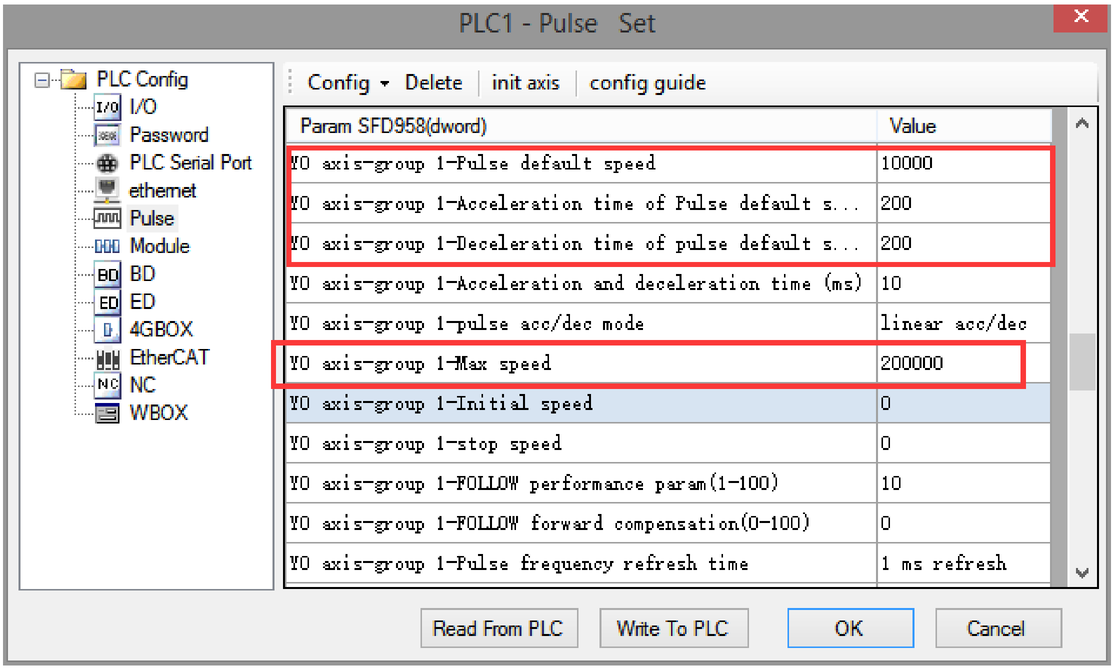


In the parameter configuration table, configure as follows (circled parameters need to be modified):




PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD922(dword)
Value
Y0 axis-Common-negative limit terminal setting
X12
Y0 axis-Common-Zero clear CLR output setting
Y3
Y0 axis-Common-Return speed VH
10000
Y0 axis-Common-Creeping speed VC
500
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
20
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...
1
Read From PLC
Write To PLC
OK
Cancel

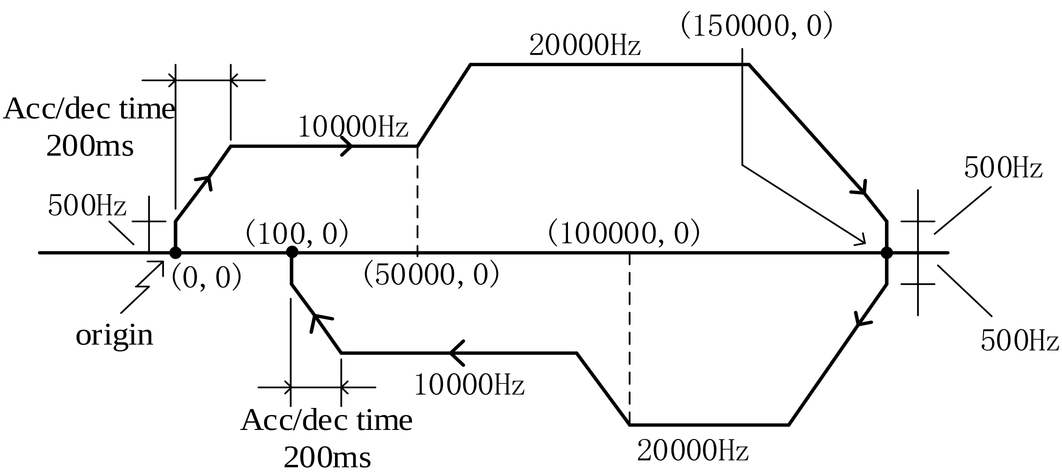


After configuring the parameters, click the "Write to PLC" button to write the parameters into the PLC. After downloading the program, power off the PLC and then power on again.

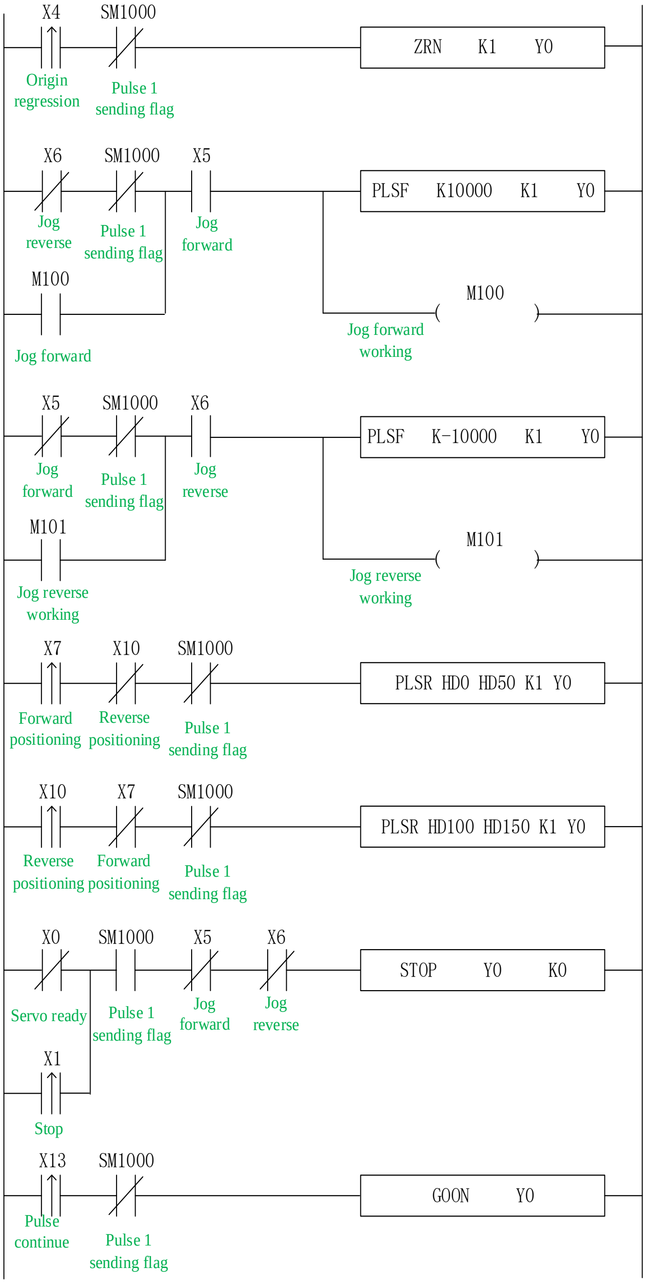
Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

1-5-4. Forward and reverse rotation multi-section process program 【PLSF, PLSR, ZRN】

Example 1: According to the following figure, use multi-segment absolute positioning mode.



Firstly, make the ladder chart as follows:



//mechanical origin regression instruction (use group 1 parameters)

//Jog forward control, the speed is 10000Hz (use group 1 parameters)

//jog reverse control, the speed is -10000Hz (use group 1 paramreters)

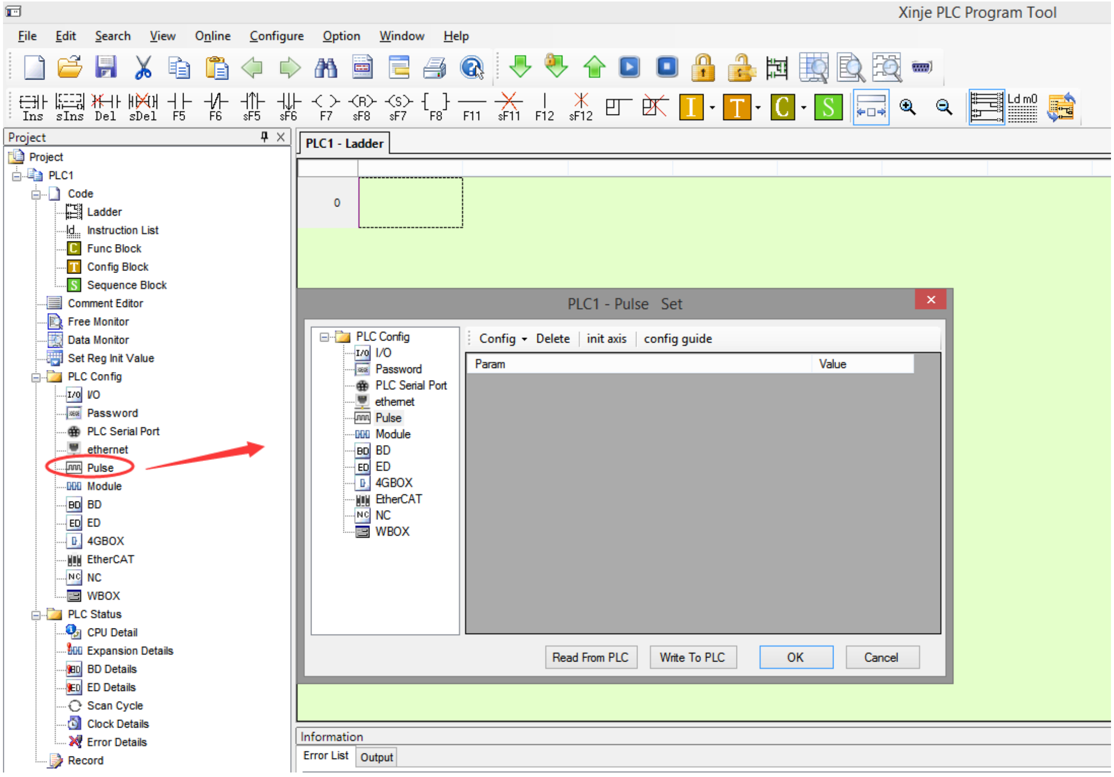
//forward positioning control (use group 1 parameters)

// reverse positioning control (use group 1 parameters)

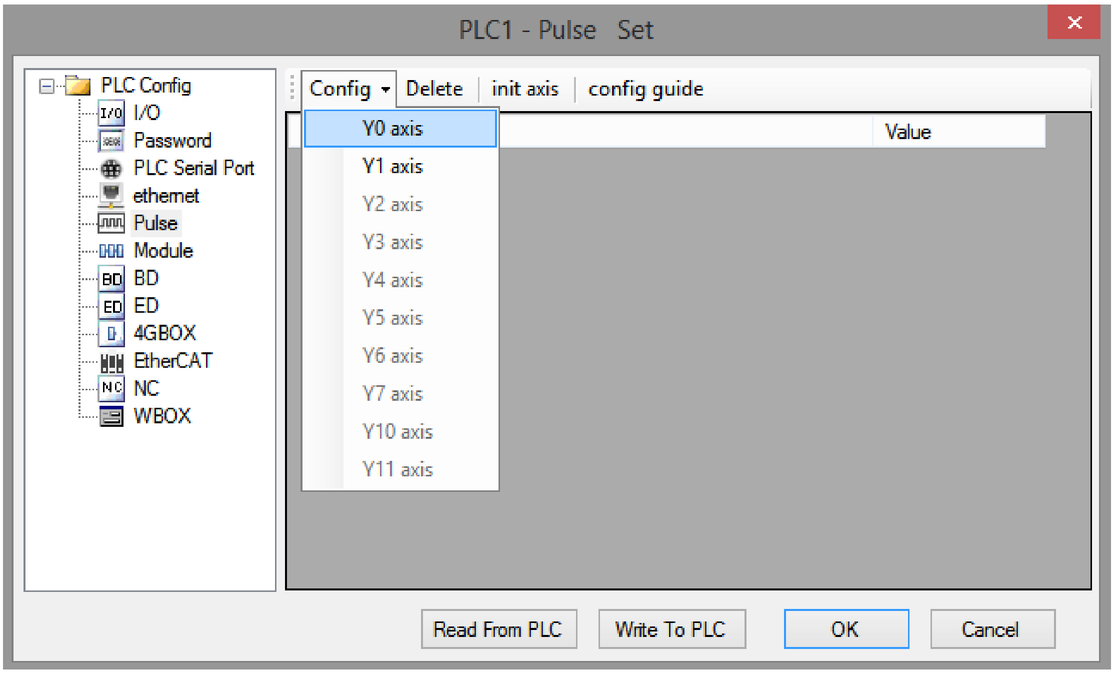
//pulse stop (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

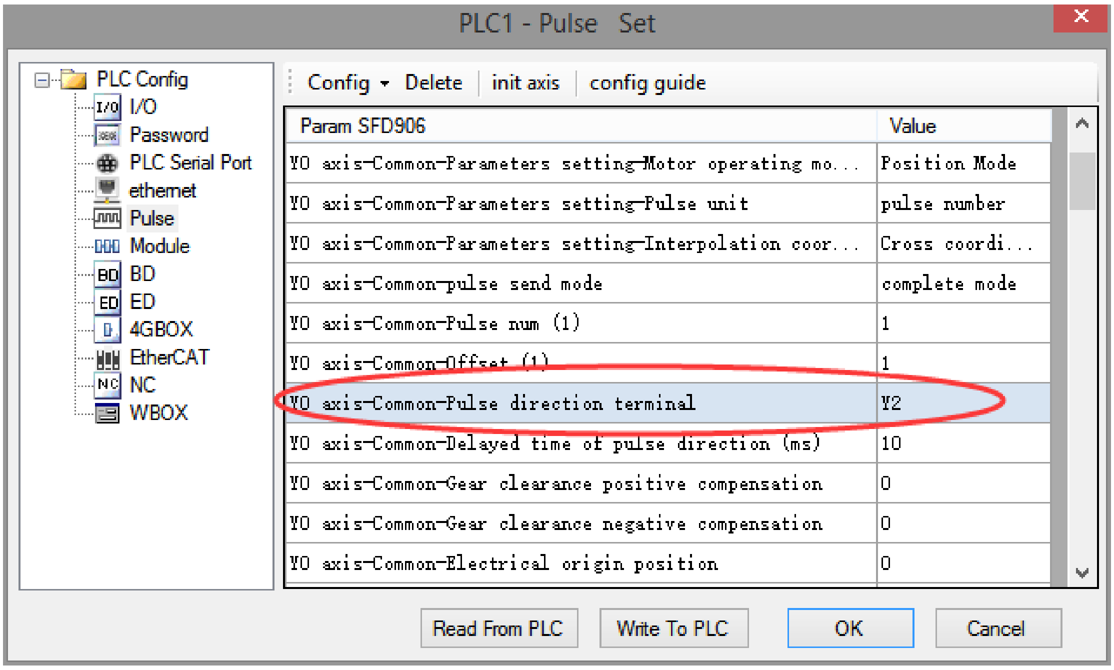
In the sample program, except DRVI and DRVA, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

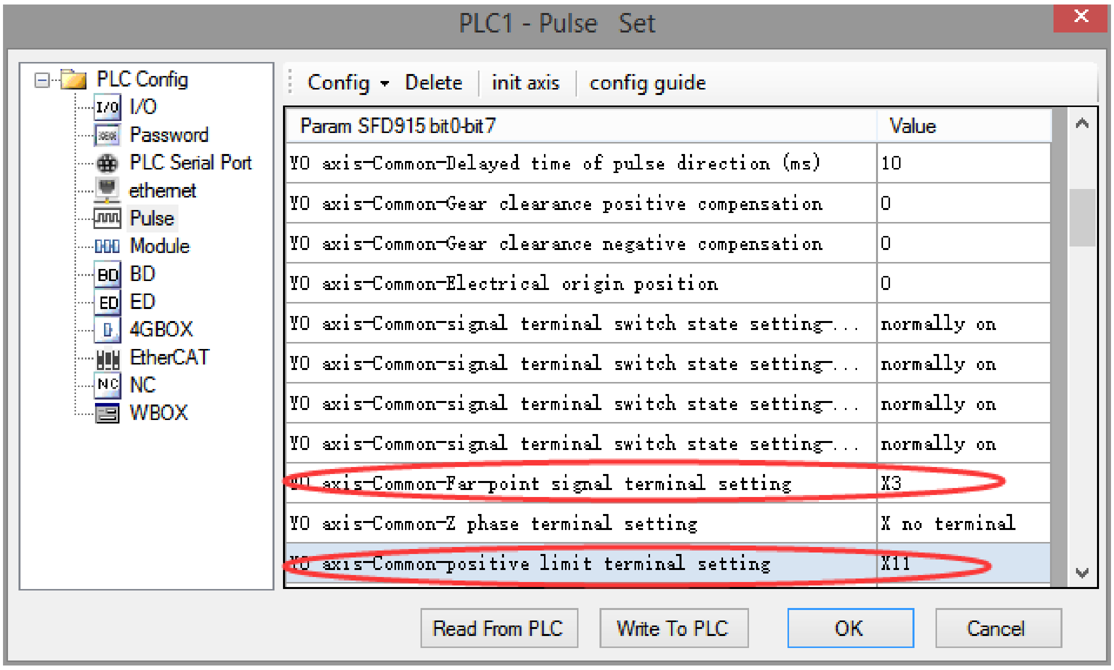


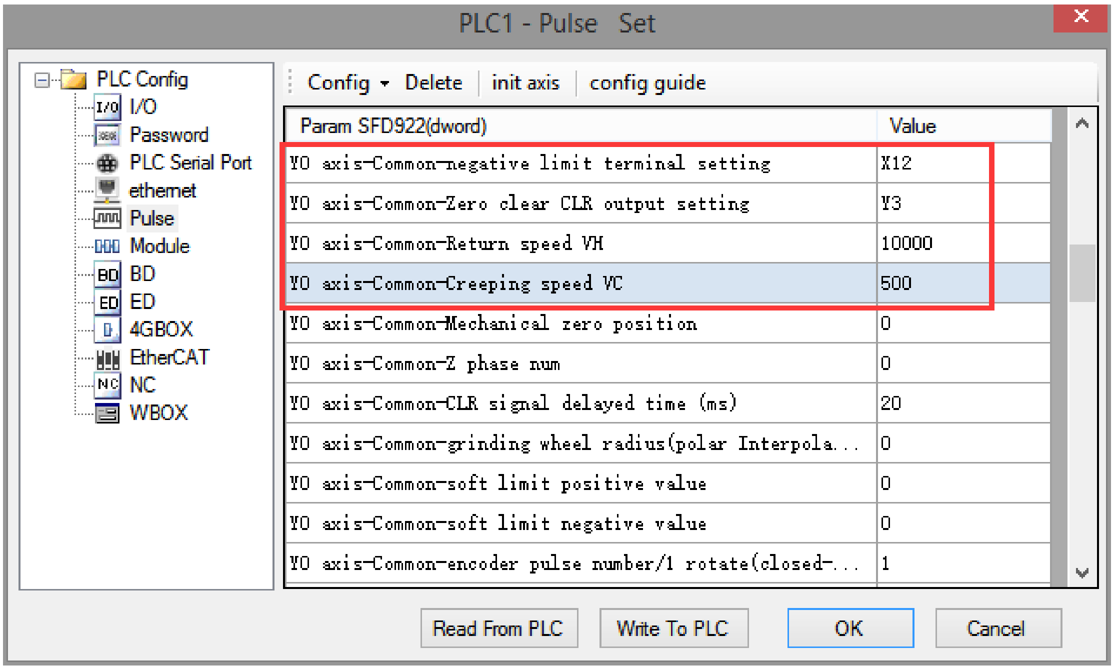
Click config, then select Y0 axis.

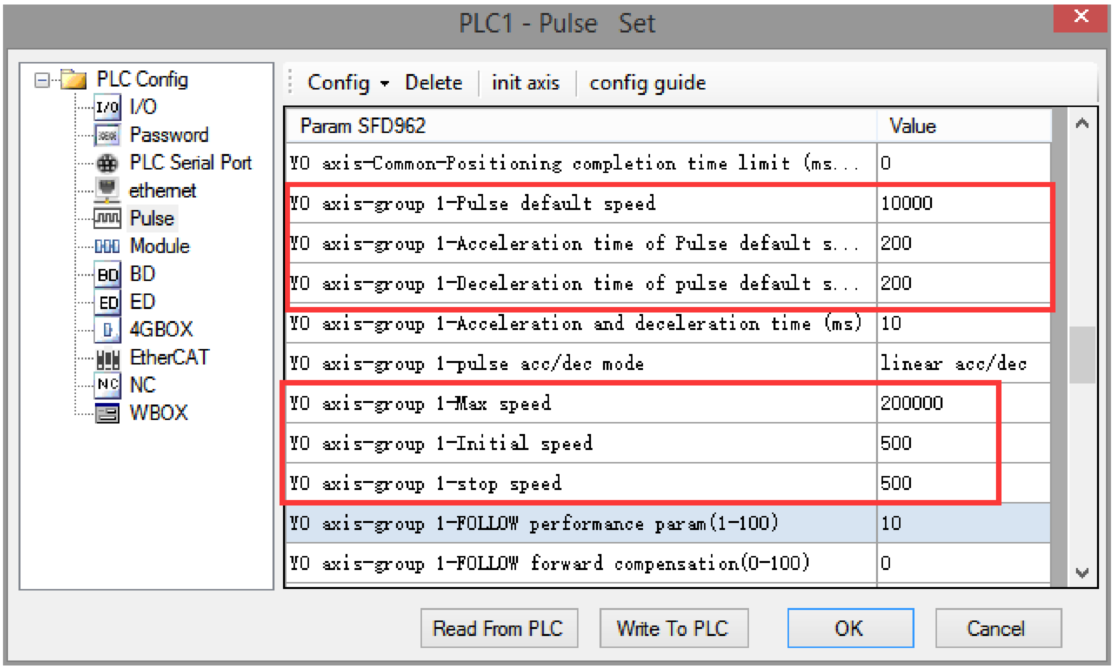


In the parameter configuration table, configure as follows (circled parameters need to be modified):


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD906
Value
Y0 axis-Common-Parameters setting-Motor operating mo...
Position Mode
Y0 axis-Common-Parameters setting-Pulse unit
pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...
Cross coordi...
Y0 axis-Common-pulse send mode
complete mode
Y0 axis-Common-Pulse num (1)
1
Y0 axis-Common-Offset (1)
1
Y0 axis-Common-Pulse direction terminal
Y2
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Read From PLC
Write To PLC
OK
Cancel

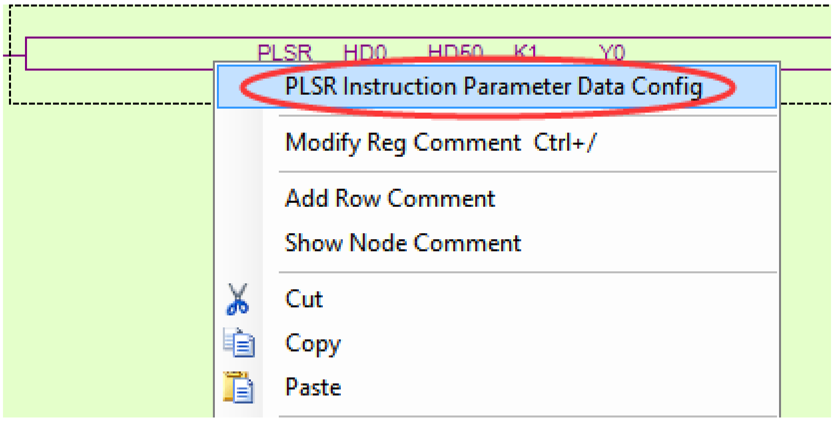

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel



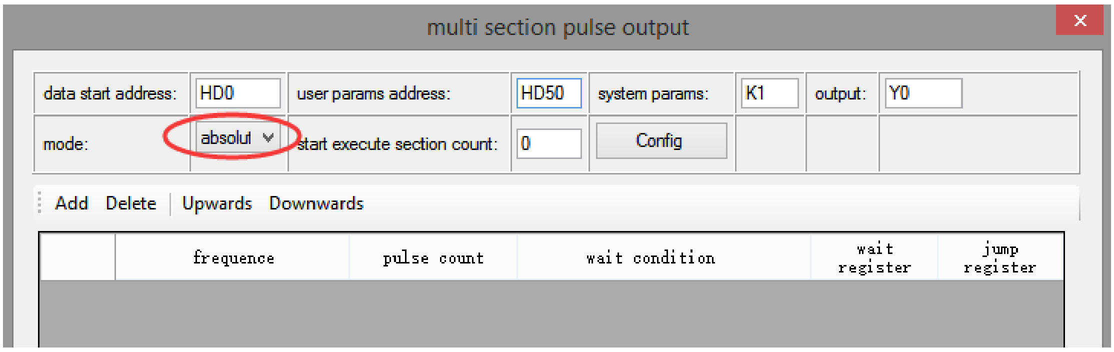


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

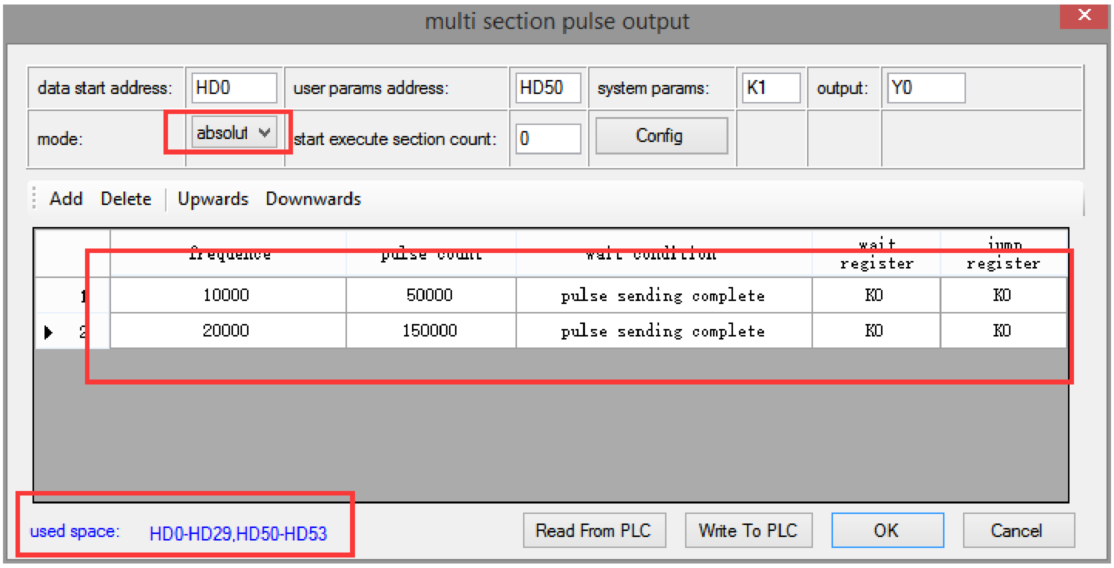
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:

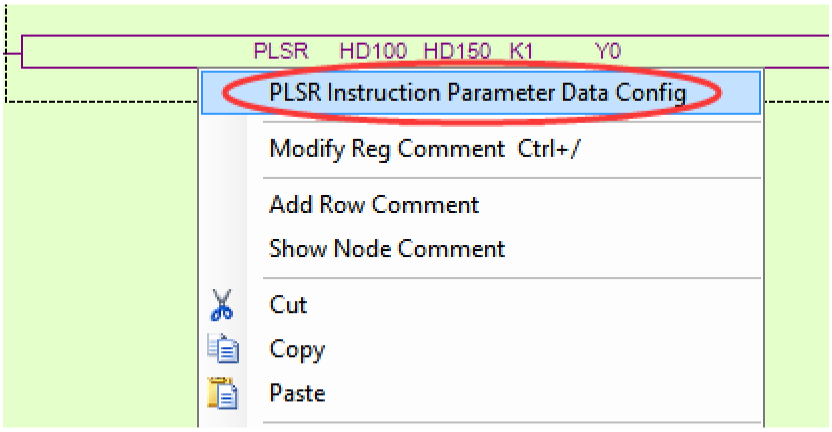


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

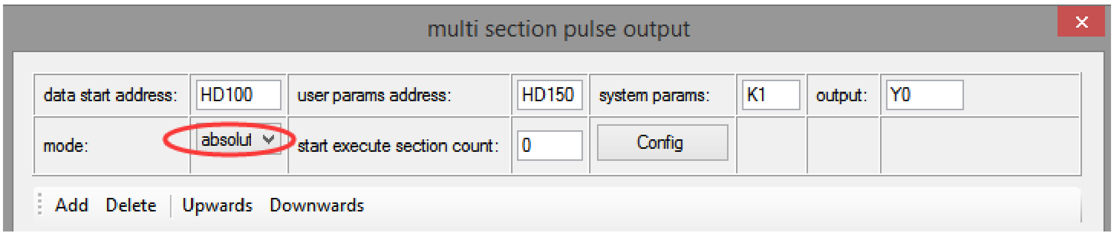


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

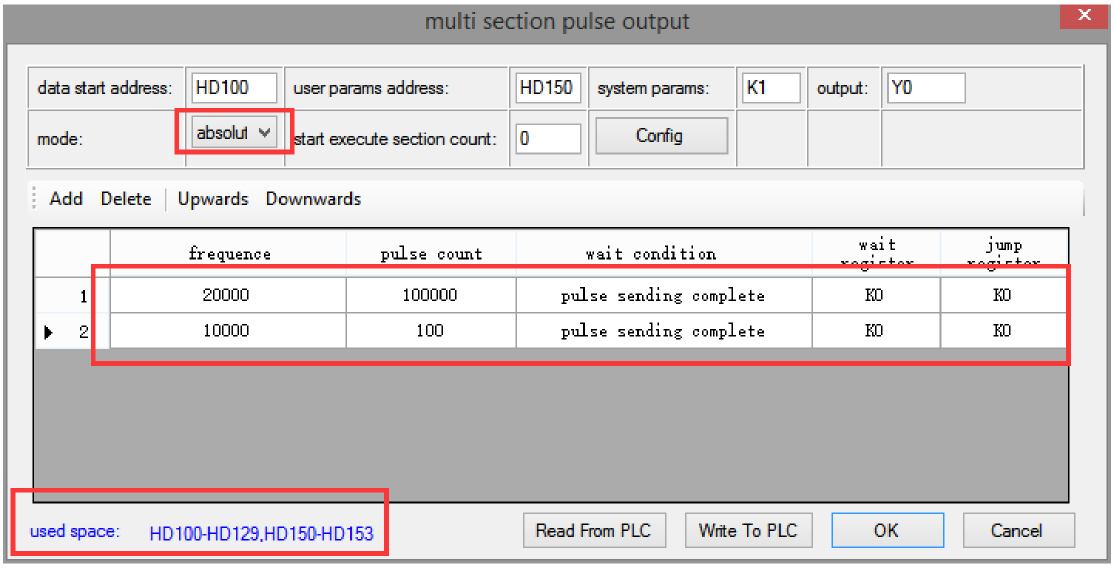
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:



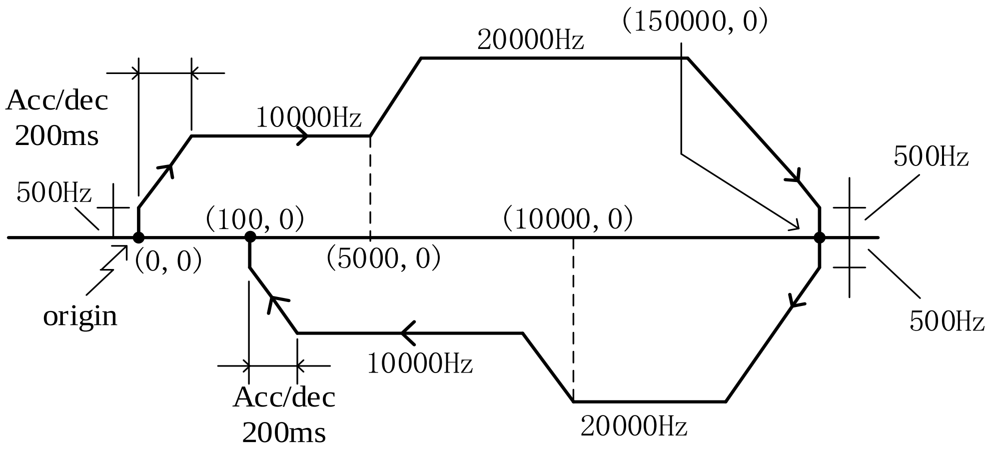
After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:



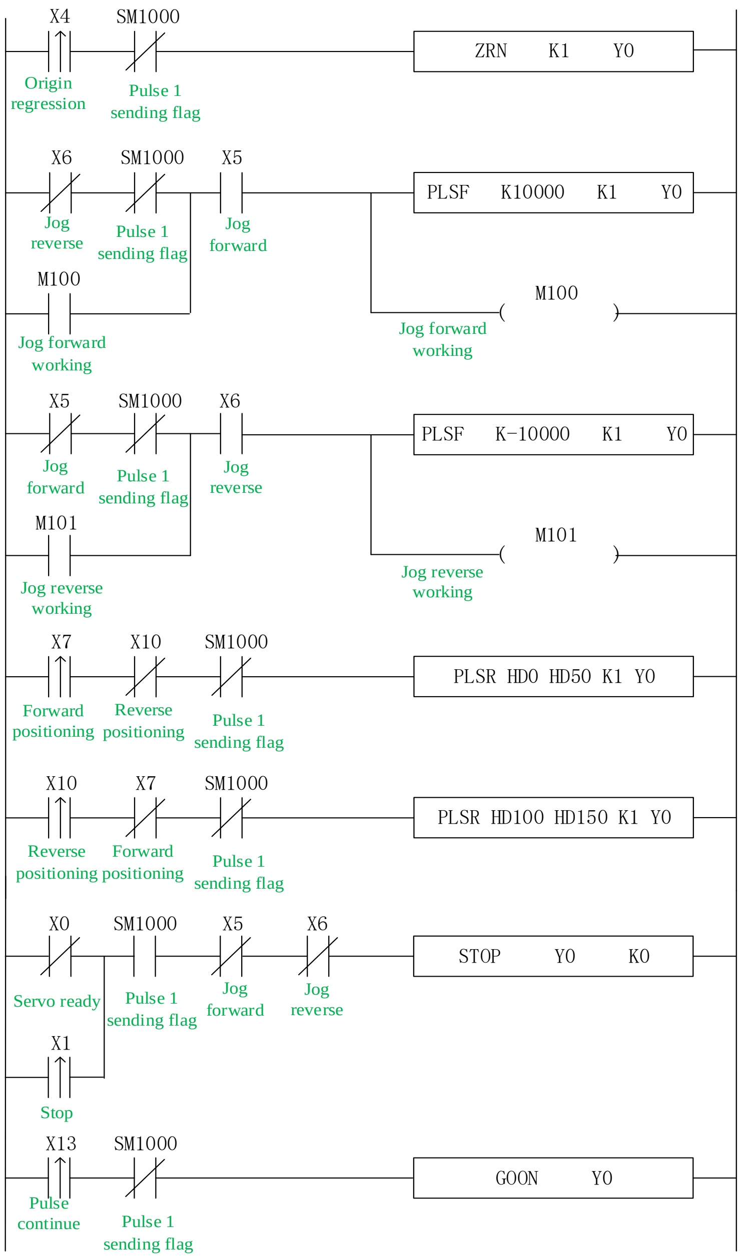
Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

After downloading the program, power off the PLC and then re-energize it.
Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

Example 2: According to the following figure, multi-segment relative positioning method is used.



Firstly, make the ladder chart as follows:



//mechanical origin regression instruction (use group 1 parameters)

//Jog forward control, the speed is 10000Hz (use group 1 parameters)

//jog reverse control, the speed is -10000Hz (use group 1 paramreters)

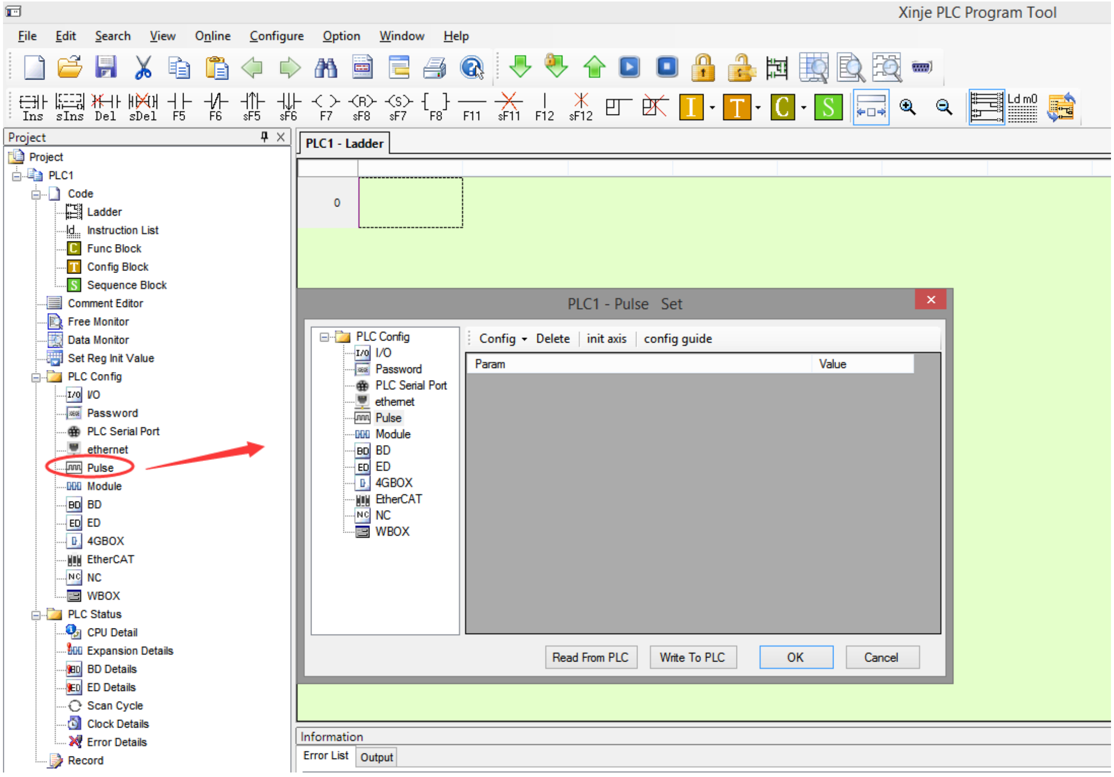
//forward positioning control (use group 1 parameters)

// reverse positioning control (use group 1 parameters)

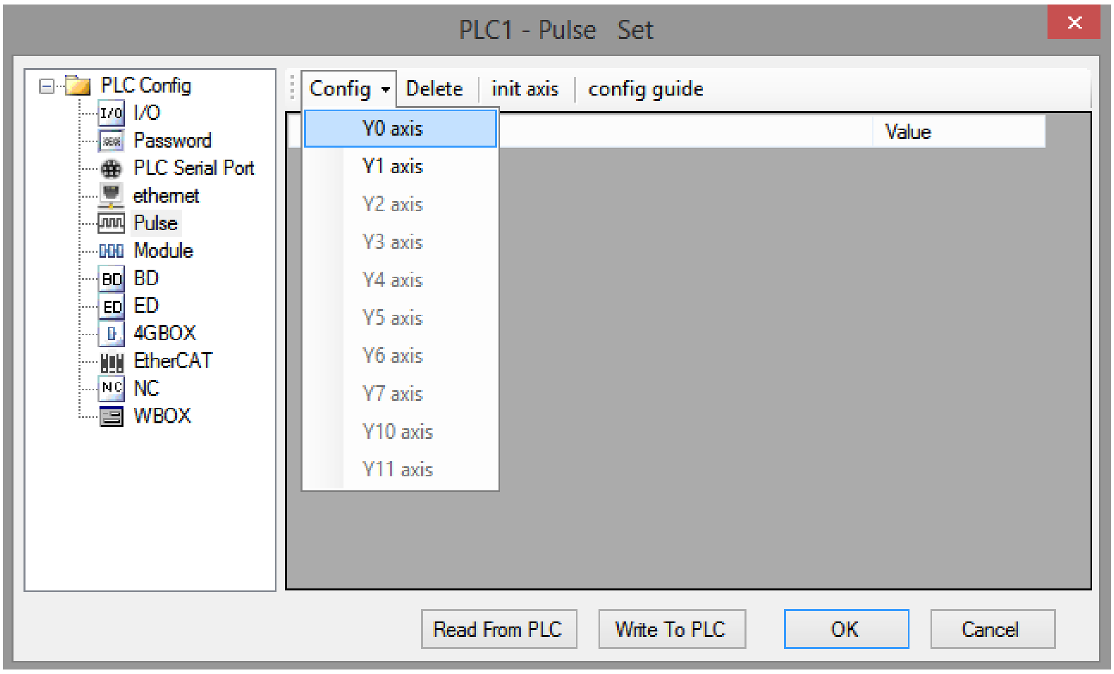
//pulse stop (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

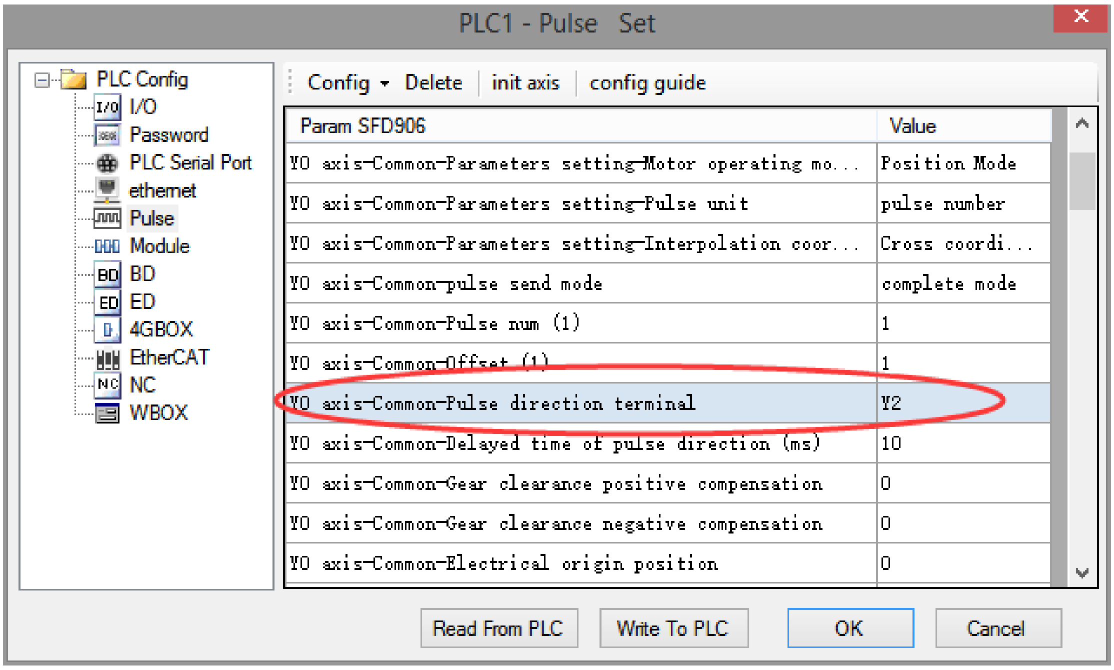
In the sample program, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

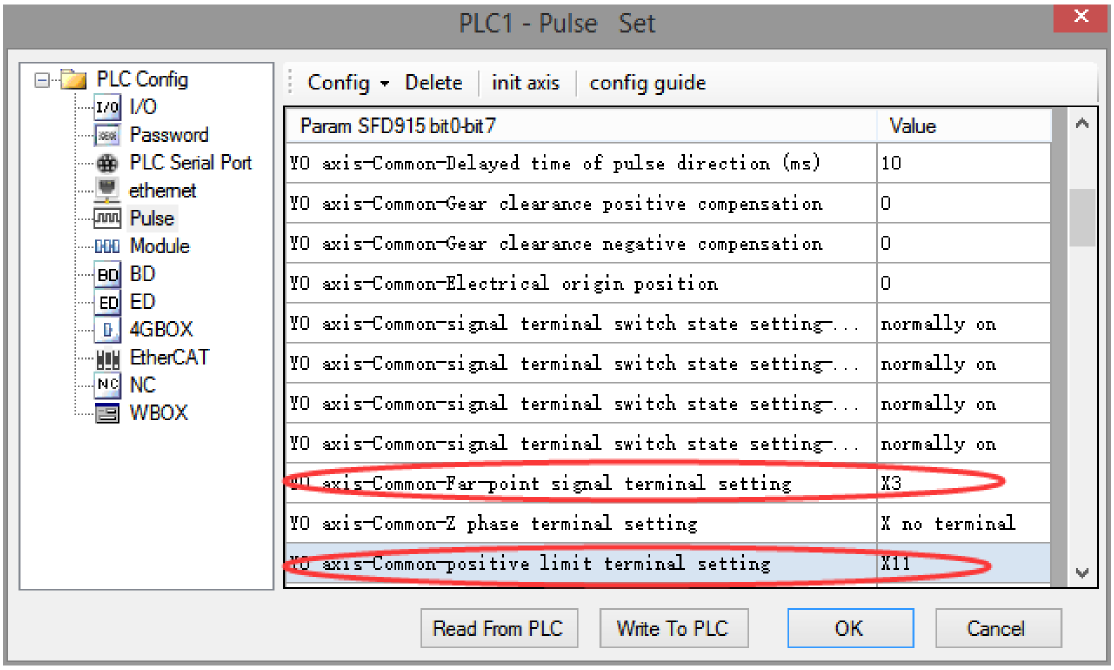


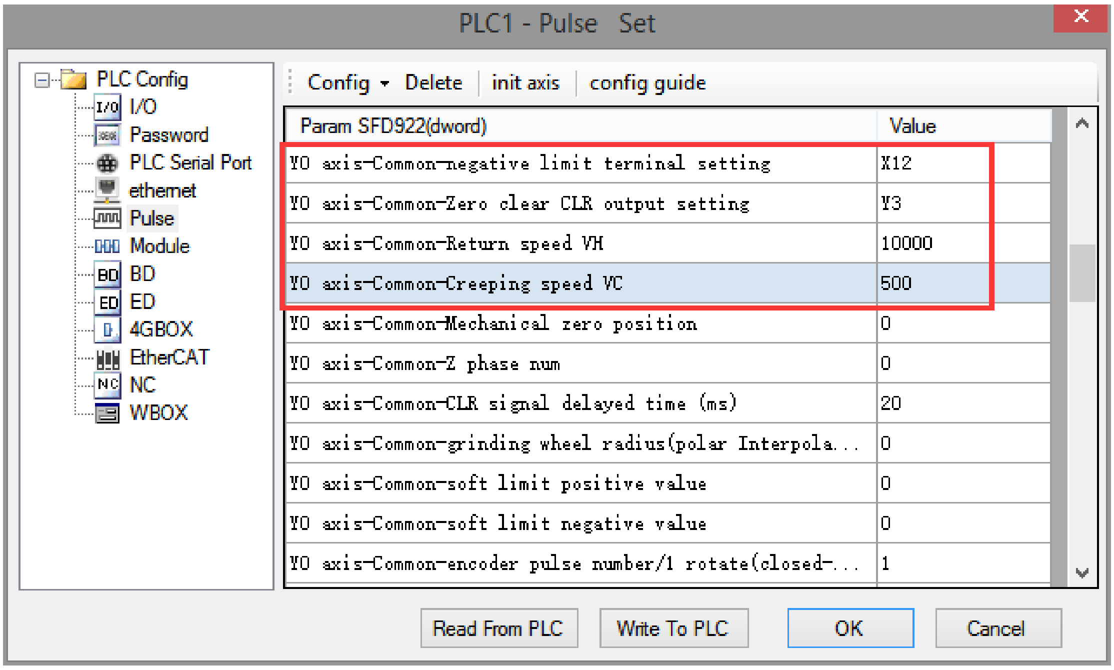
Click config, then select Y0 axis.

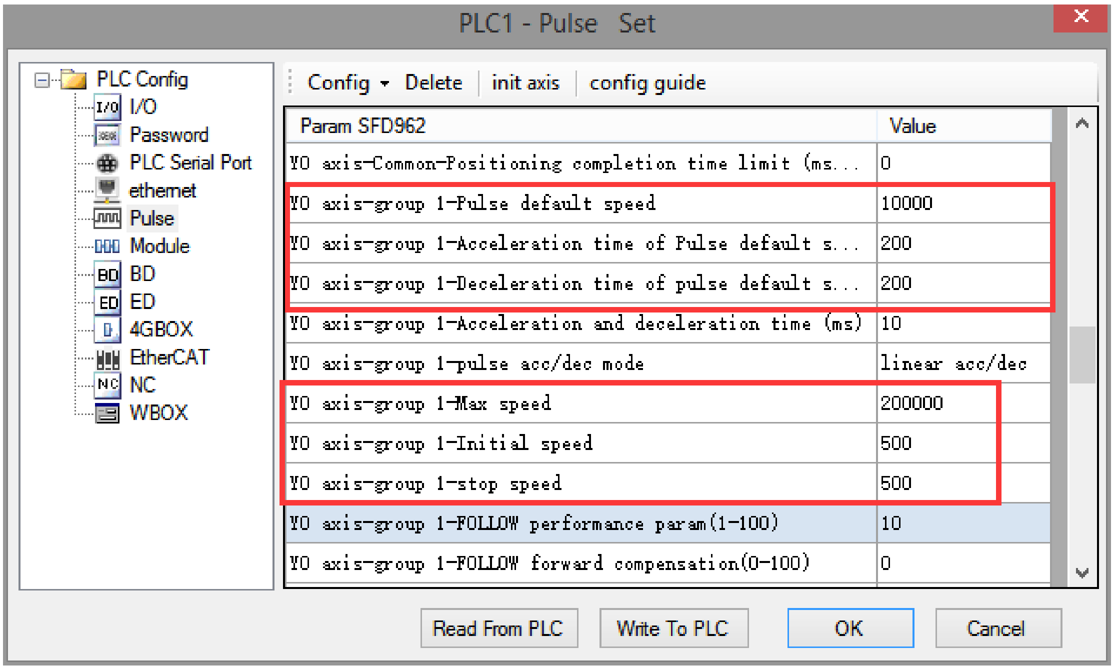


In the parameter configuration table, configure as follows (circled parameters need to be modified):


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD906
Value
Y0 axis-Common-Parameters setting-Motor operating mo...
Position Mode
Y0 axis-Common-Parameters setting-Pulse unit
pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...
Cross coordi...
Y0 axis-Common-pulse send mode
complete mode
Y0 axis-Common-Pulse num (1)
1
Y0 axis-Common-Offset (1)
1
Y0 axis-Common-Pulse direction terminal
Y2
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Read From PLC
Write To PLC
OK
Cancel

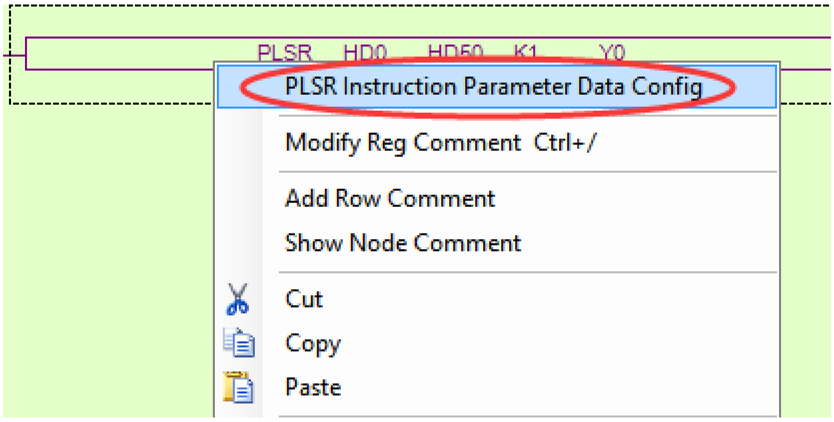

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel



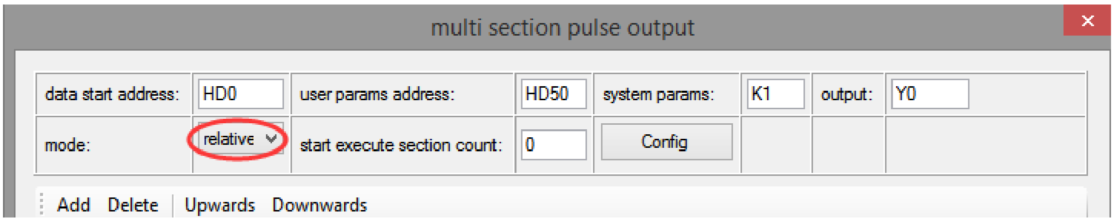


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

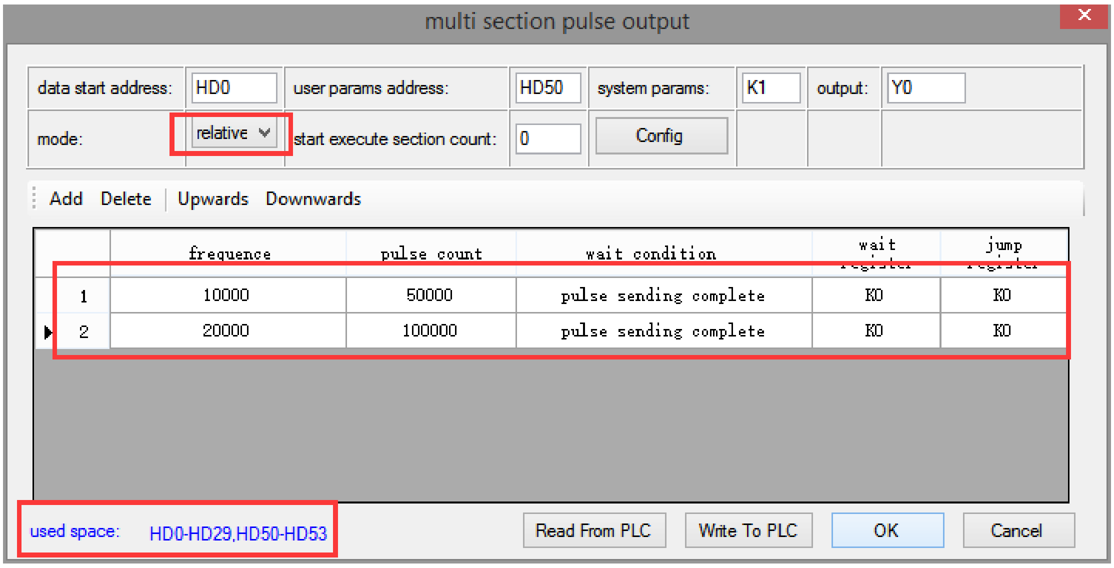
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:

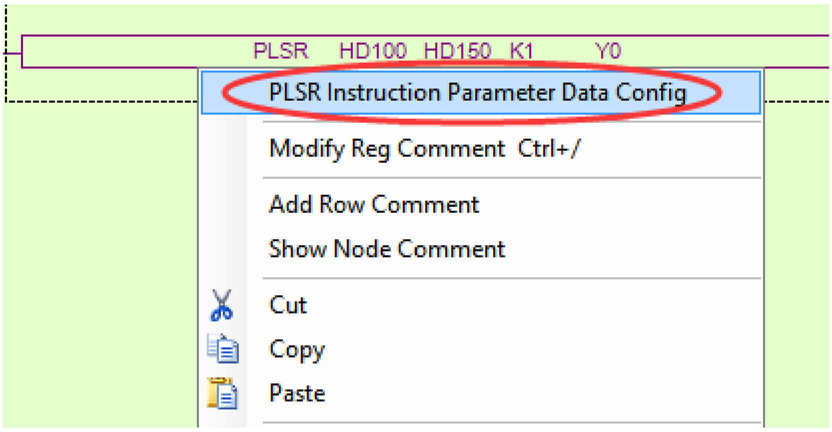


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

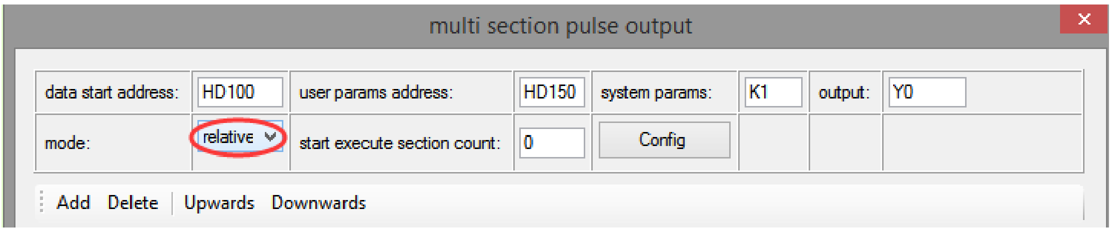


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

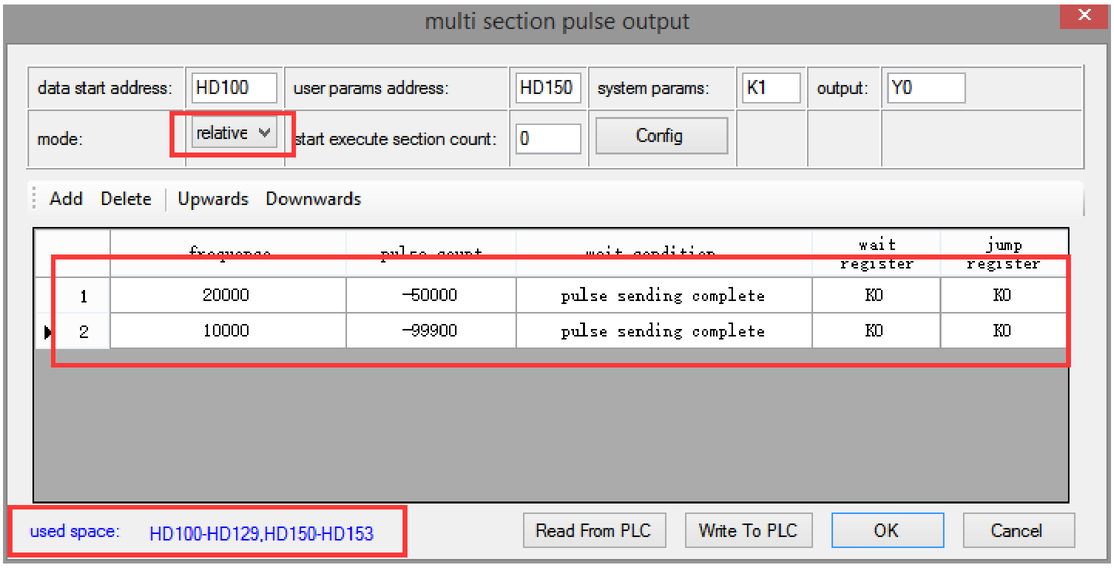
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:



After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of reverse rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

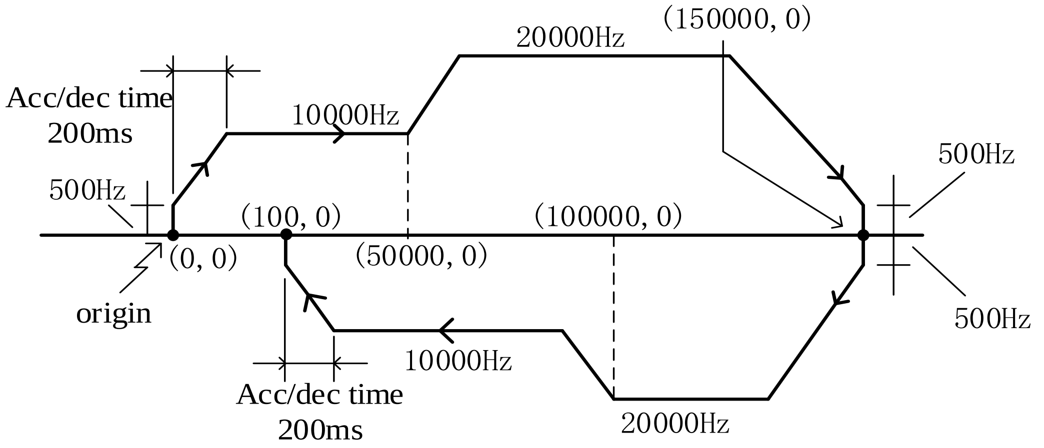


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

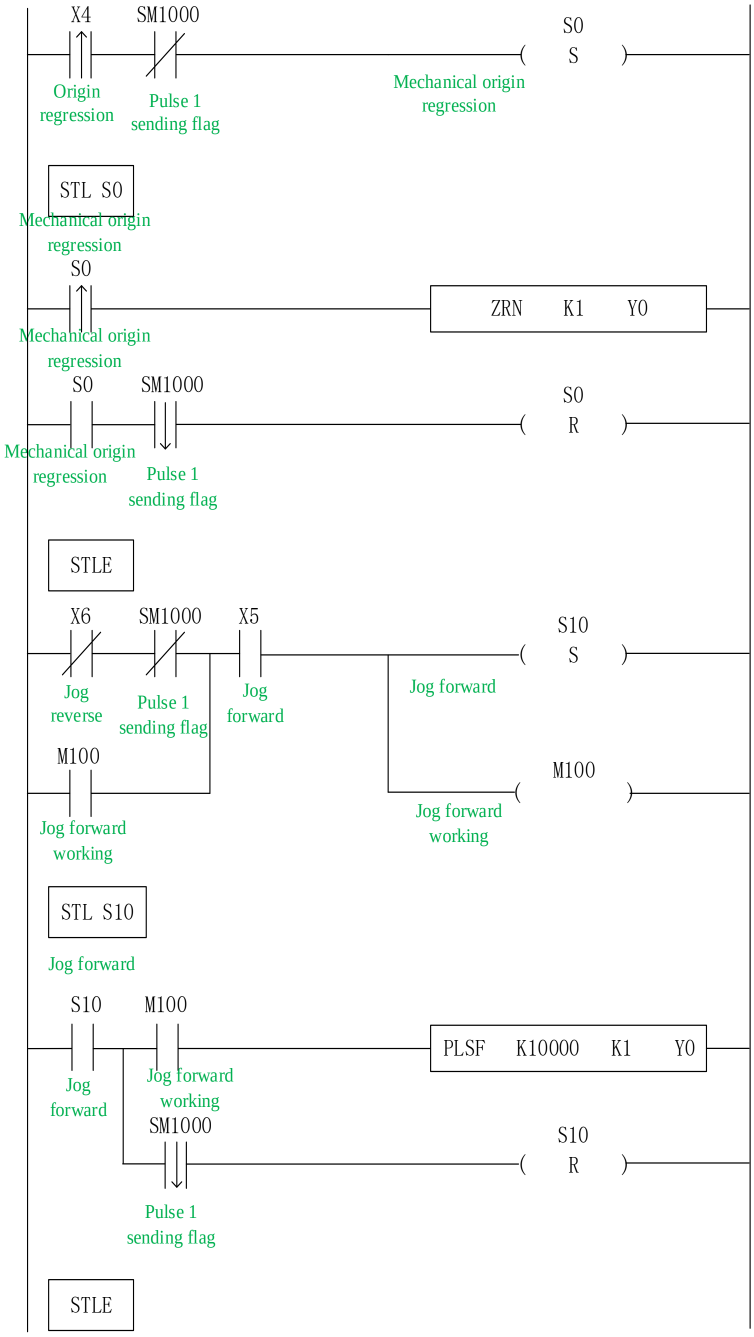
After downloading the program, power off the PLC and then re-energize it.
Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

1-5-5. Forward reverse multi-segment process program【PLSF, PLSR, ZRN】

Example 1: According to the following figure, multi-segment absolute positioning is used.

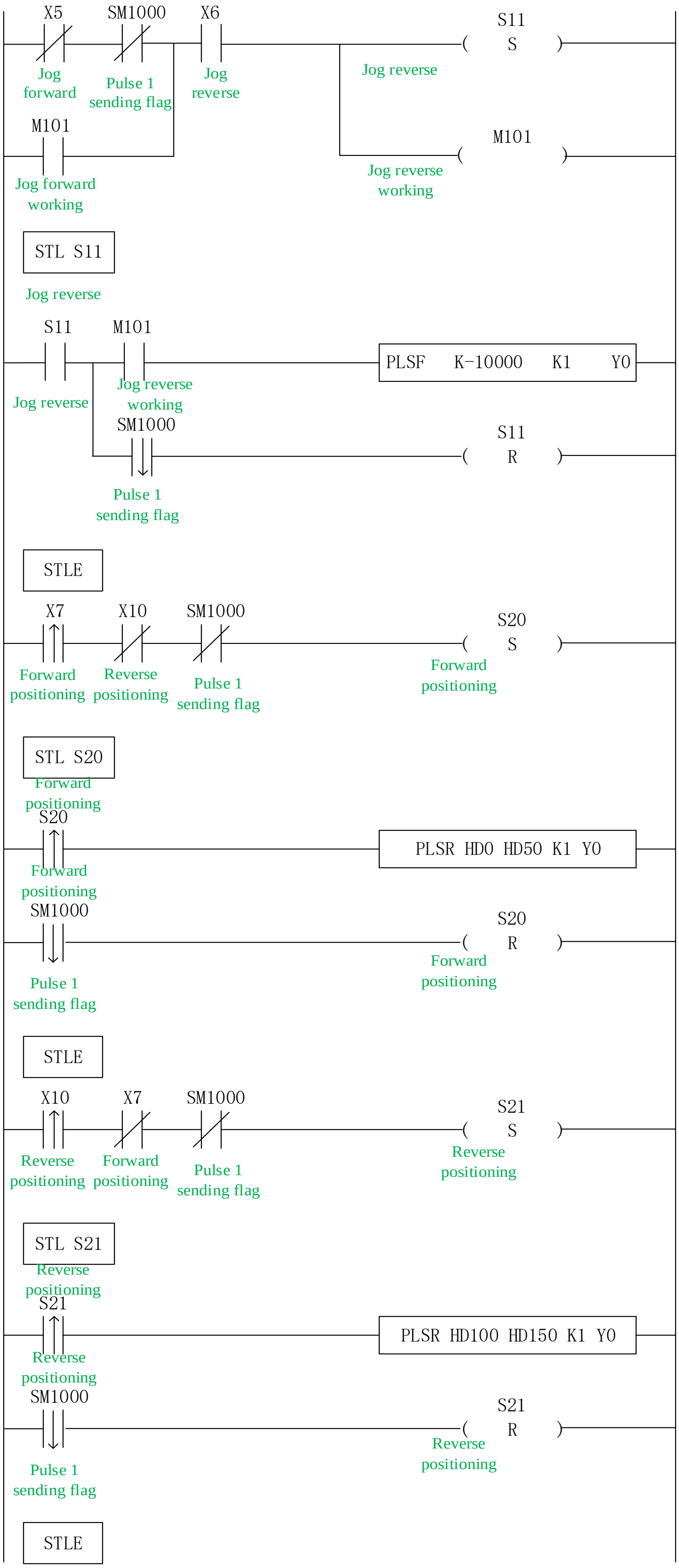


Firstly, make the ladder chart as follows:



//mechanical origin regression (use group 1 parameters)

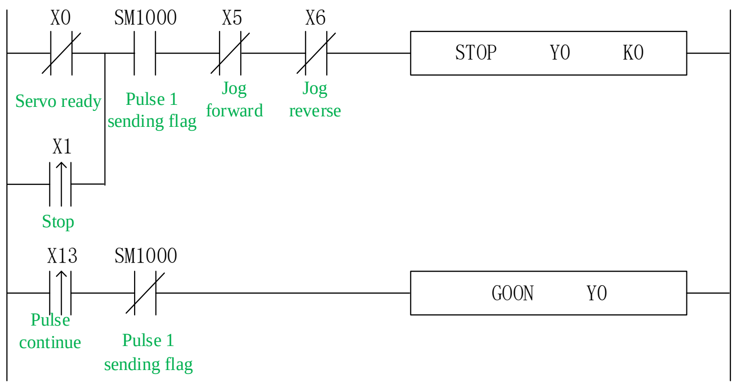
//jog forward control (use group 1 parameters)



//jog reverse control (use group 1 parameters)

//forward positioning control (use group 1 parameters)

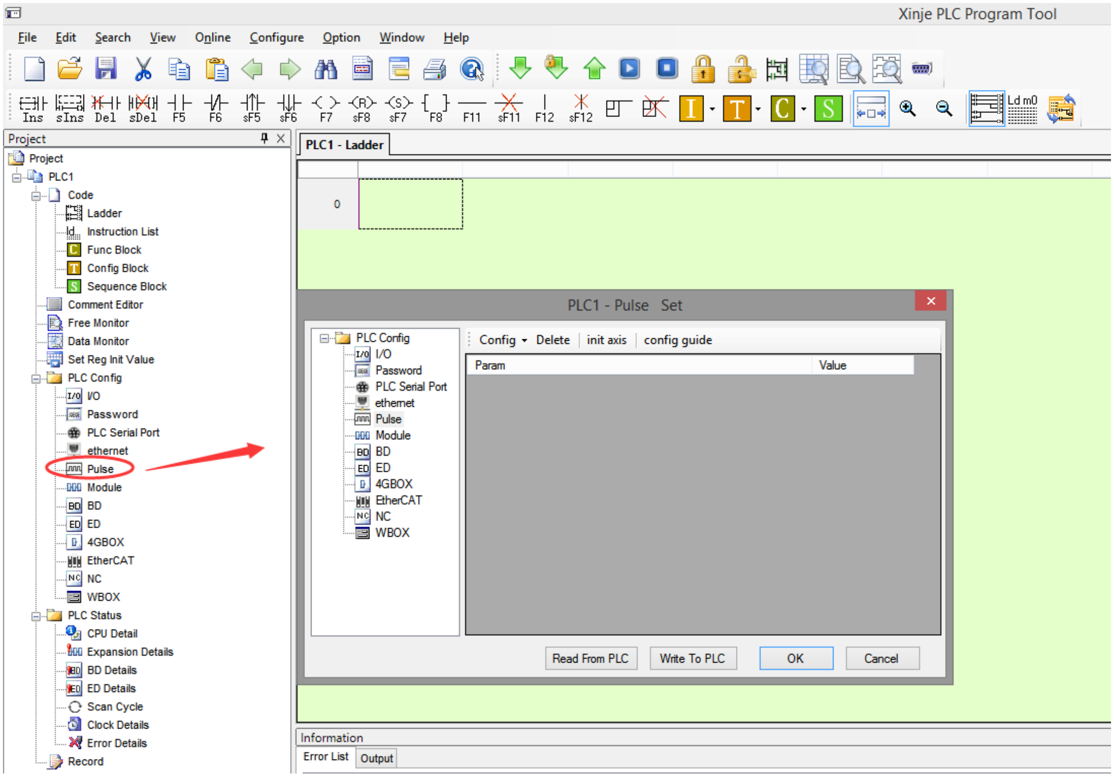
//reverse positioning control (use group 1 parameters)



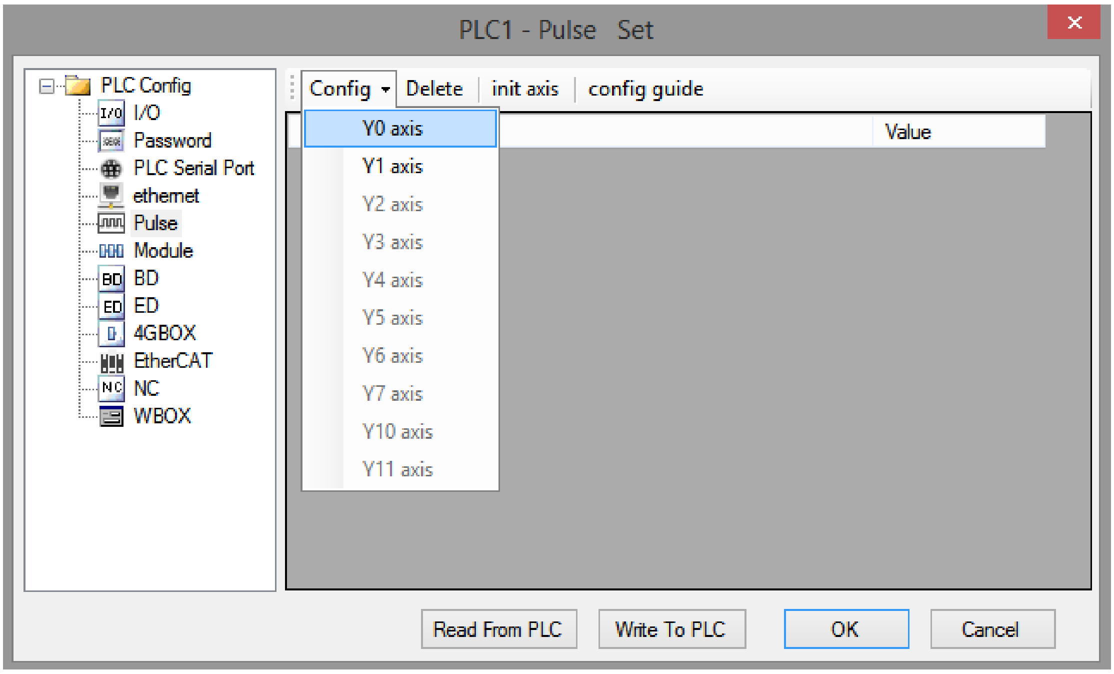
//stop pulse sending (slow stop)

// If the pulse has not been sent out, the execution pulse will continue to send out the remaining pulse.

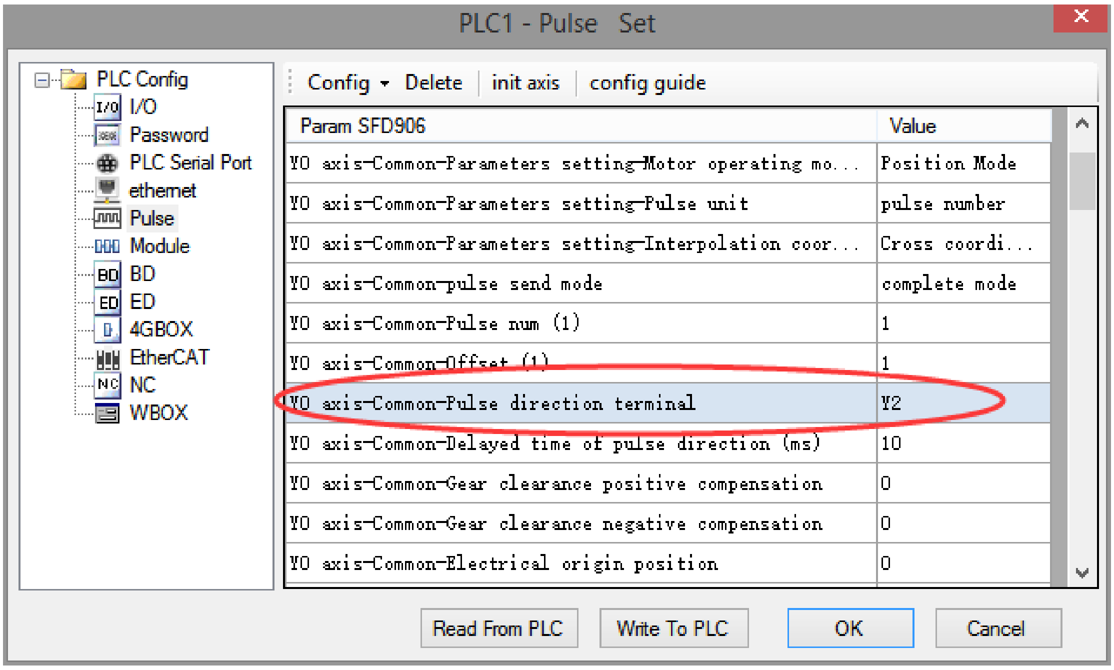
In the sample program, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

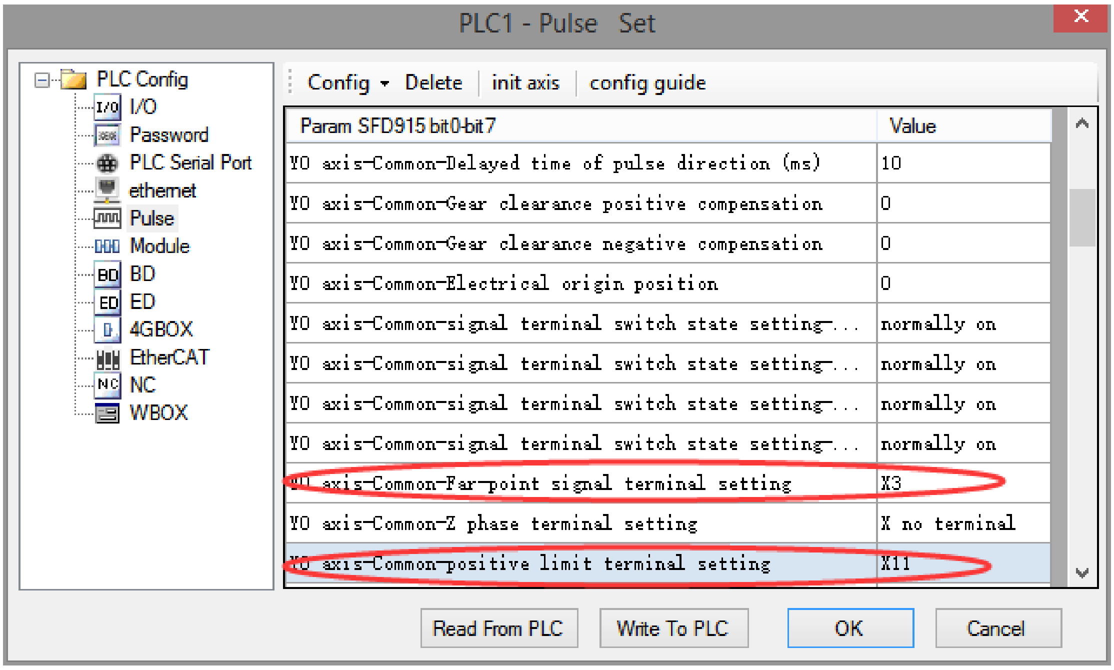


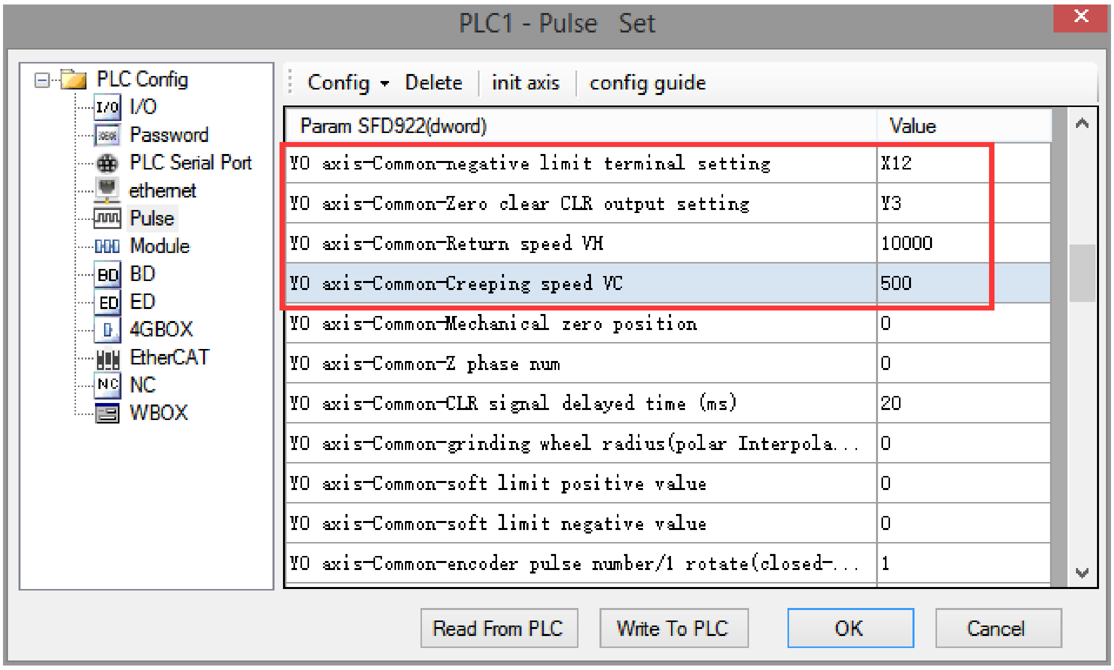
Click config, then select Y0 axis.

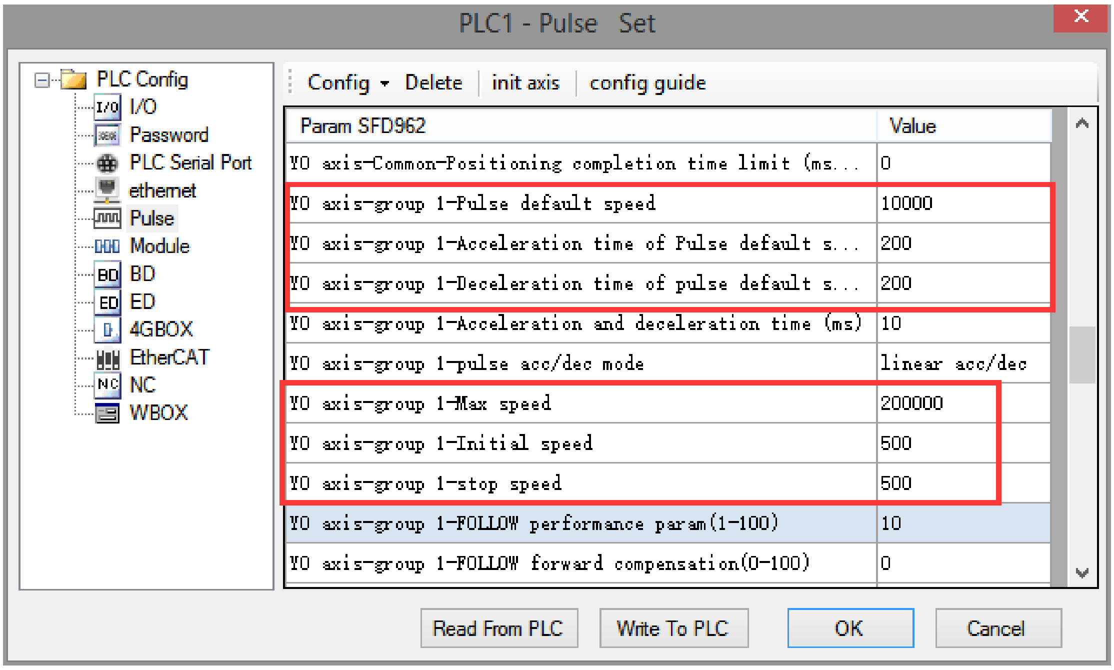


In the parameter configuration table, configure as follows (circled parameters need to be modified):



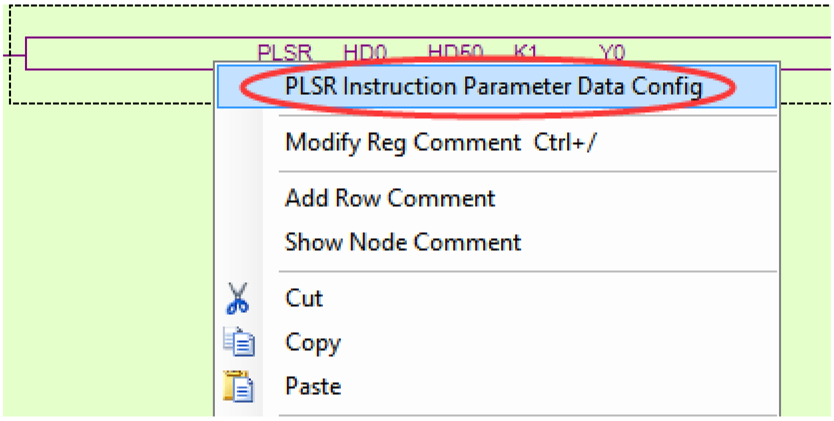

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD922(dword)
Value
Y0 axis-Common-negative limit terminal setting
X12
Y0 axis-Common-Zero clear CLR output setting
Y3
Y0 axis-Common-Return speed VH
10000
Y0 axis-Common-Creeping speed VC
500
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
20
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...
1
Read From PLC
Write To PLC
OK
Cancel

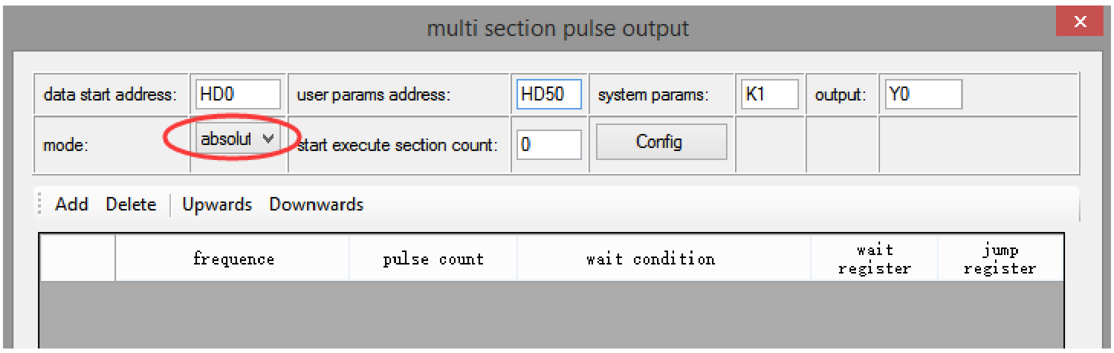


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

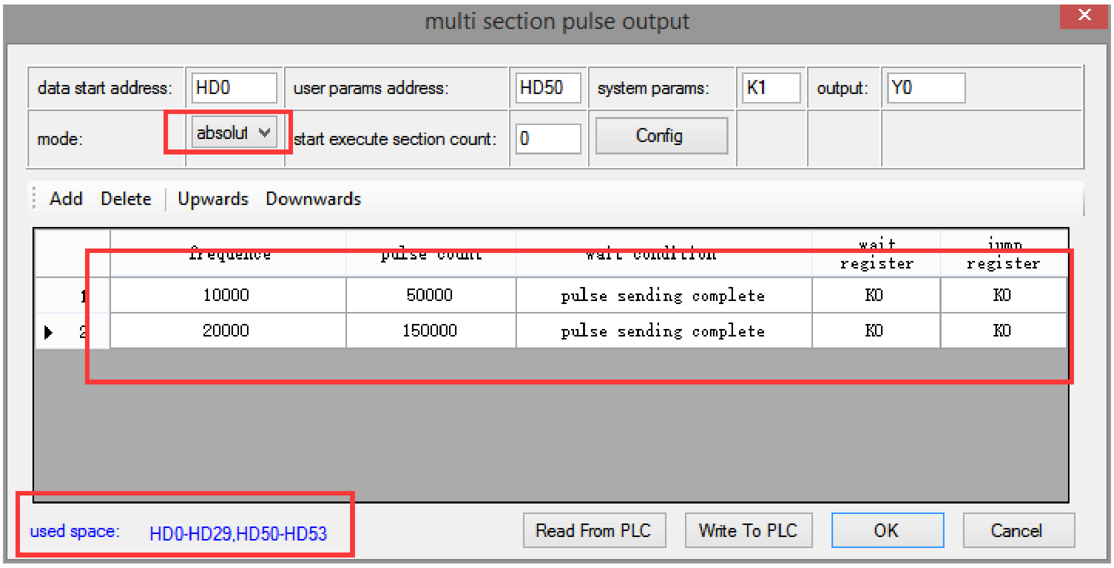
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:

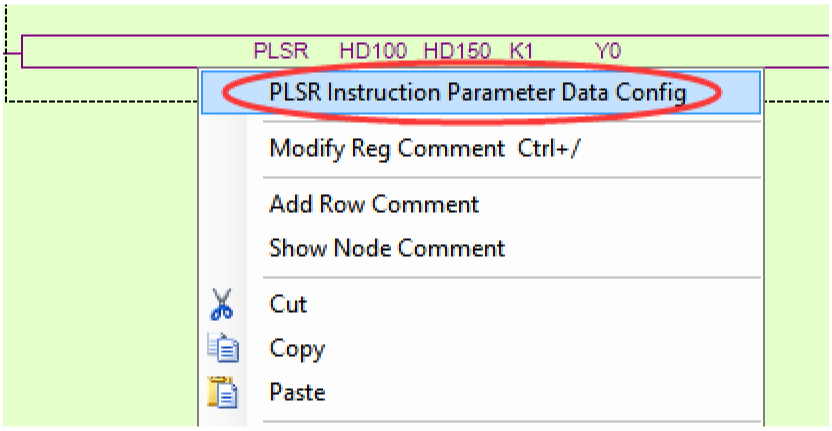


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

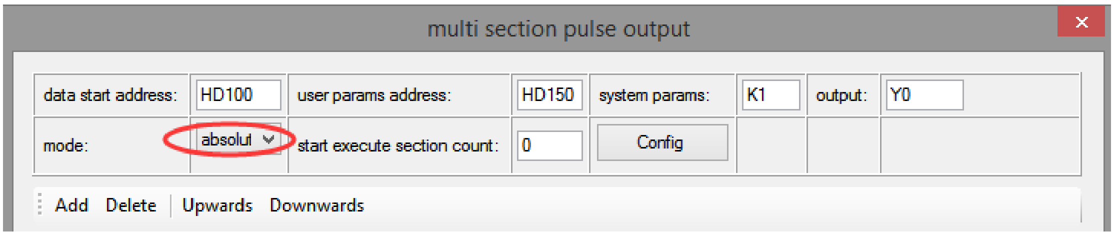


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

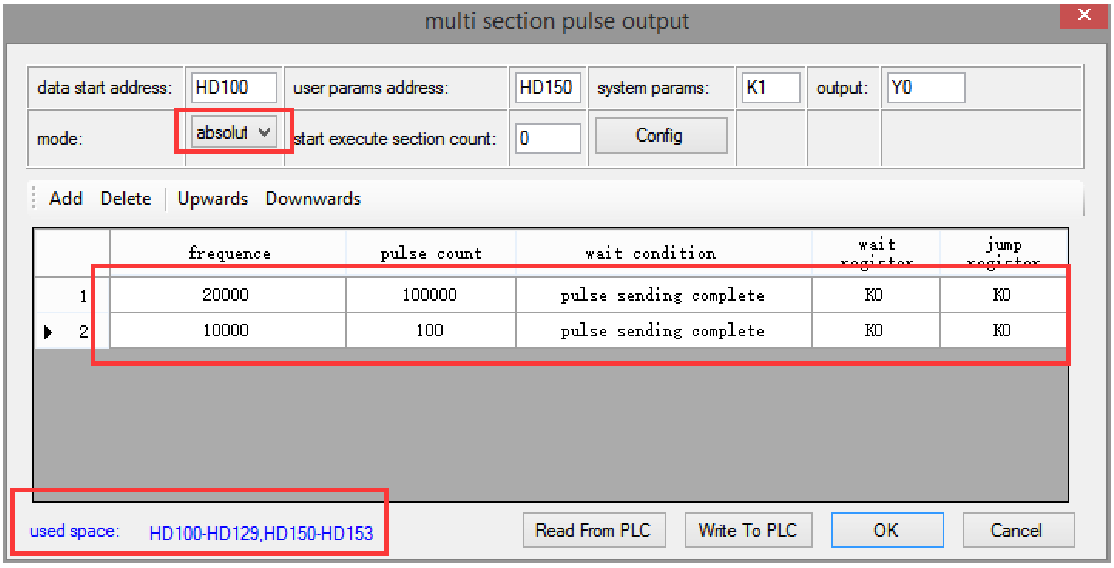
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:



After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

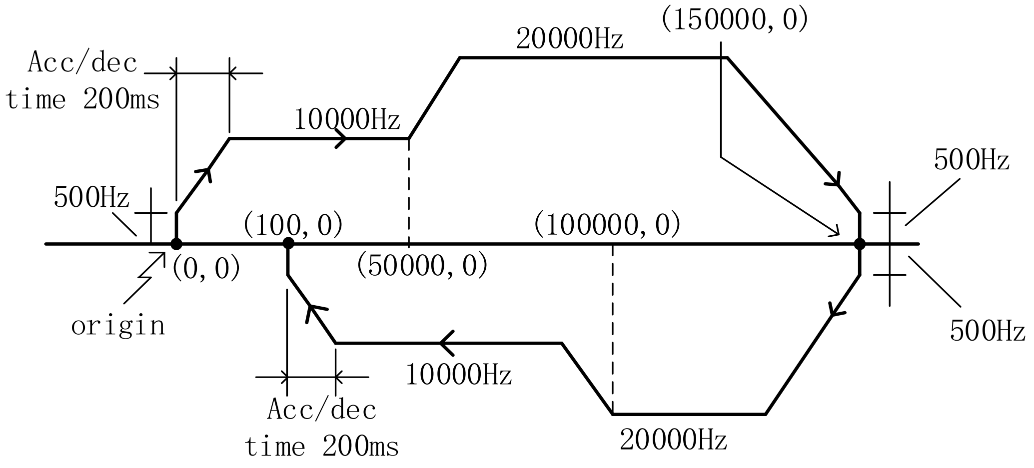


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

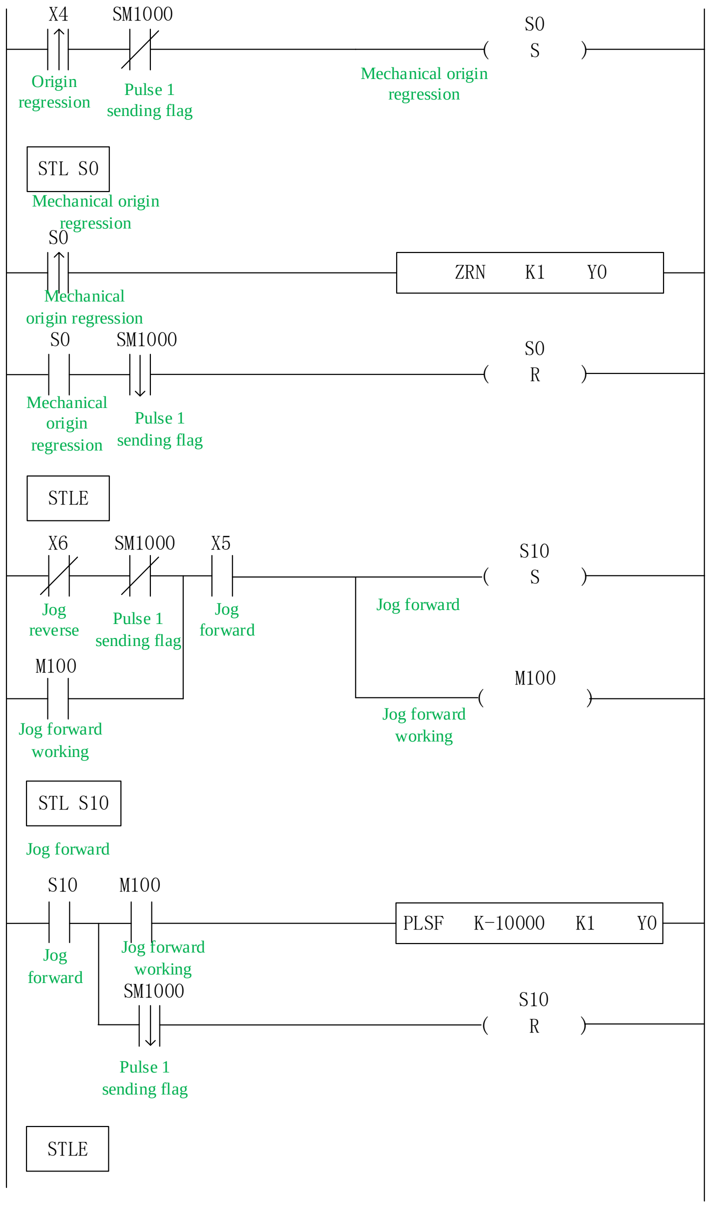
After downloading the program, power off the PLC and then re-energize it.

Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

Example 2: According to the following figure, multi-segment absolute positioning mode is adopted.

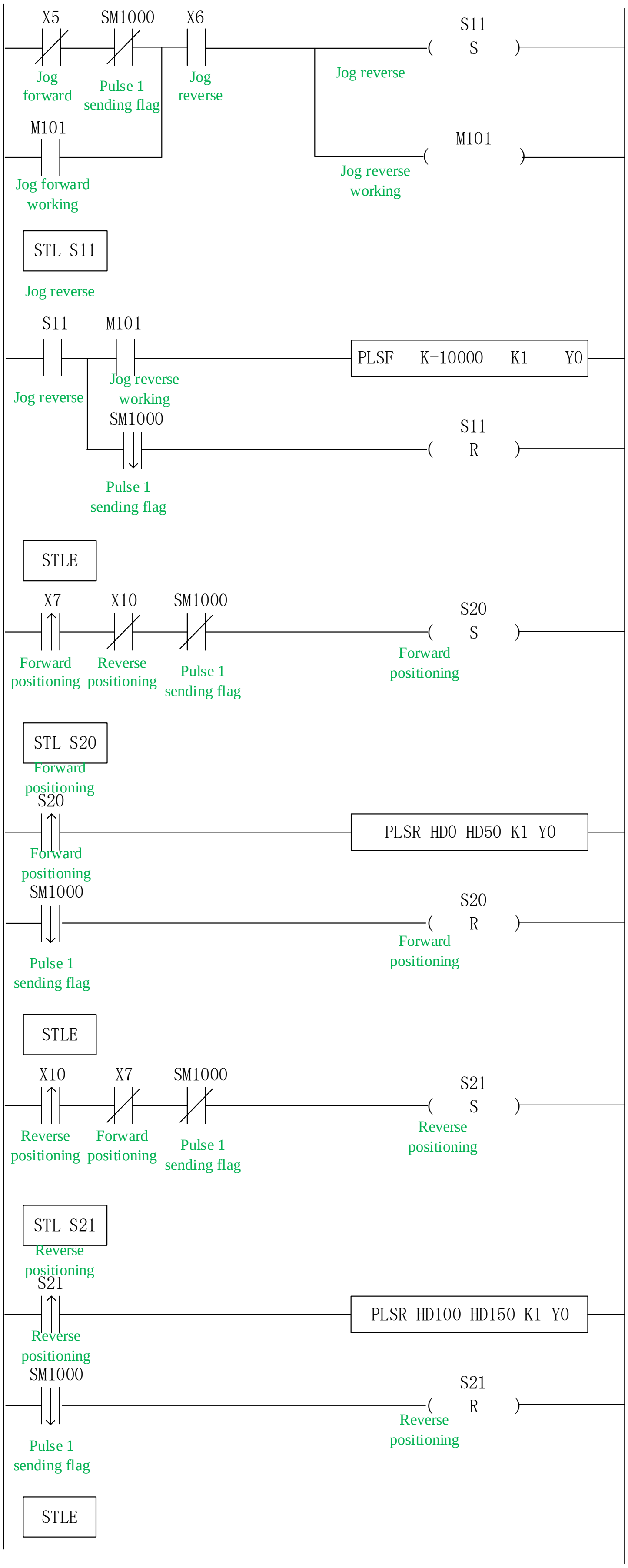


Firstly, make the ladder chart as follows:



//mechanical origin regression (use group 1 parameters)

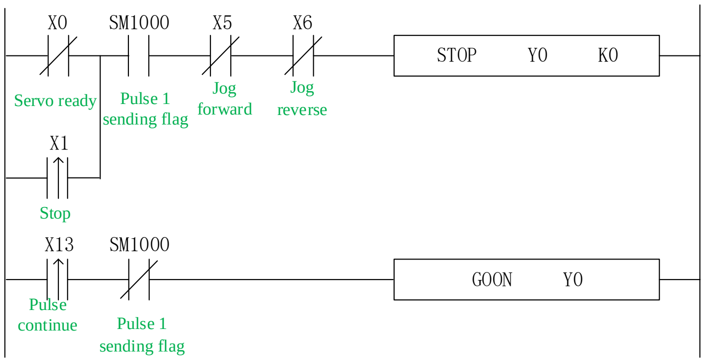
//jog forward control (use group 1 parameters)



//jog reverse control (use group 1 parameters)

//forward positioning control (use group 1 parameters)

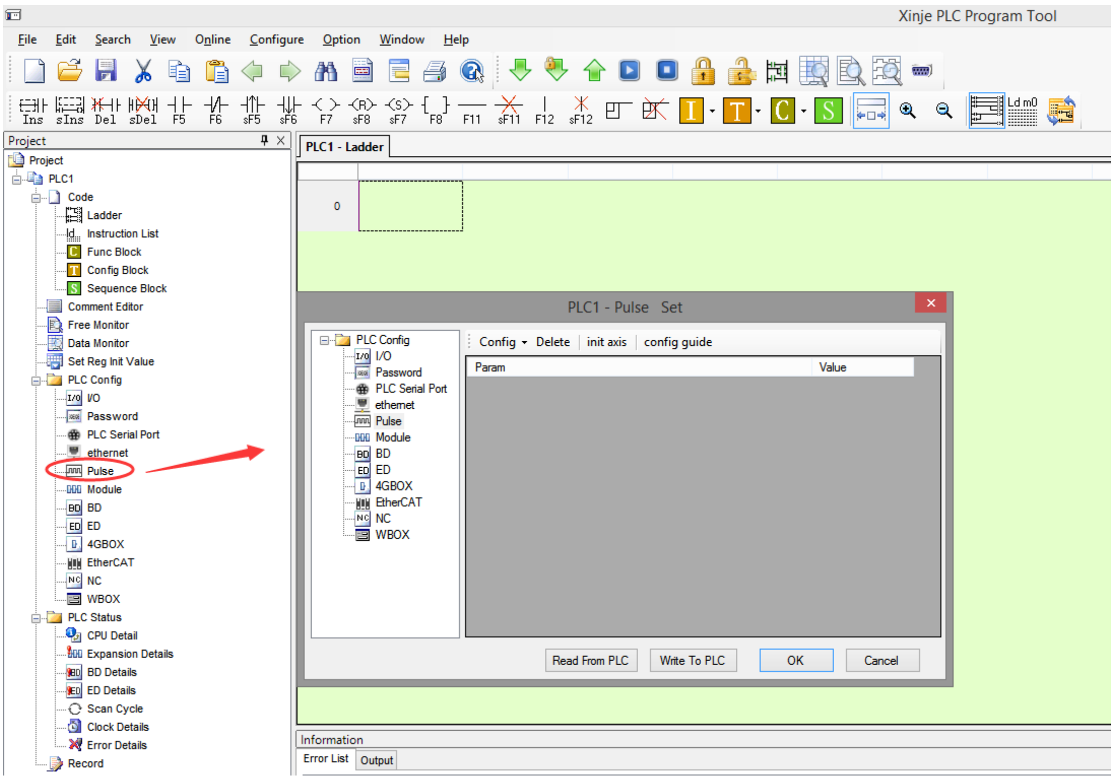
//reverse positioning control (use group 1 parameters)



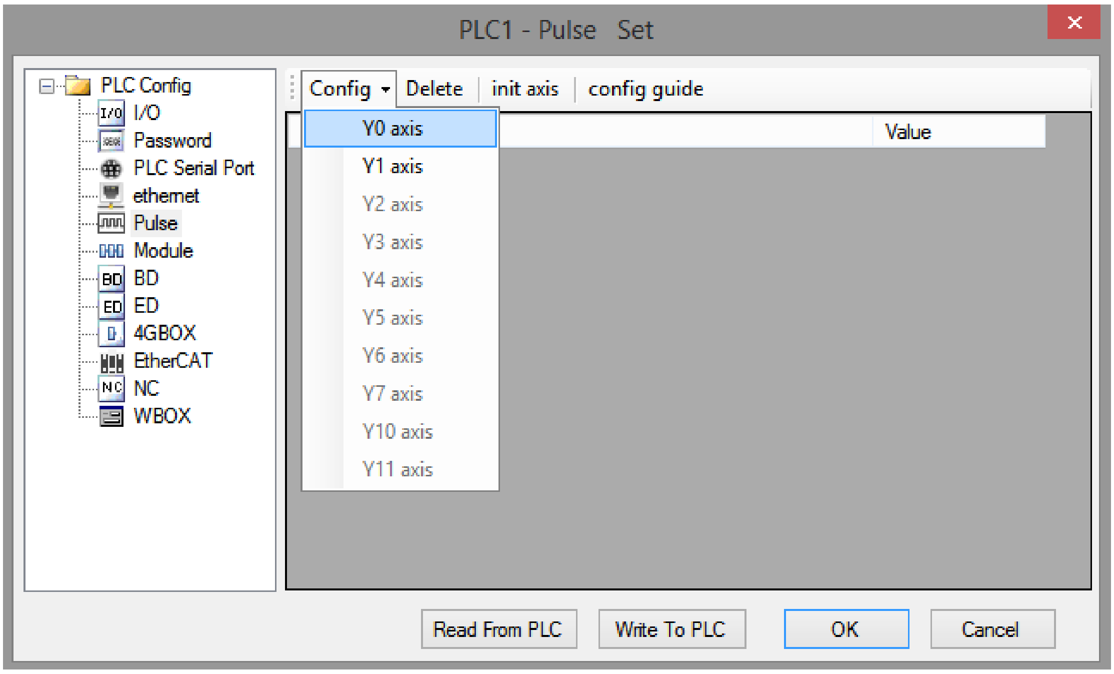
//stop pulse sending (slow stop)

// If the pulse has not been sent out, the execution pulse will continue to send out the remaining pulse.

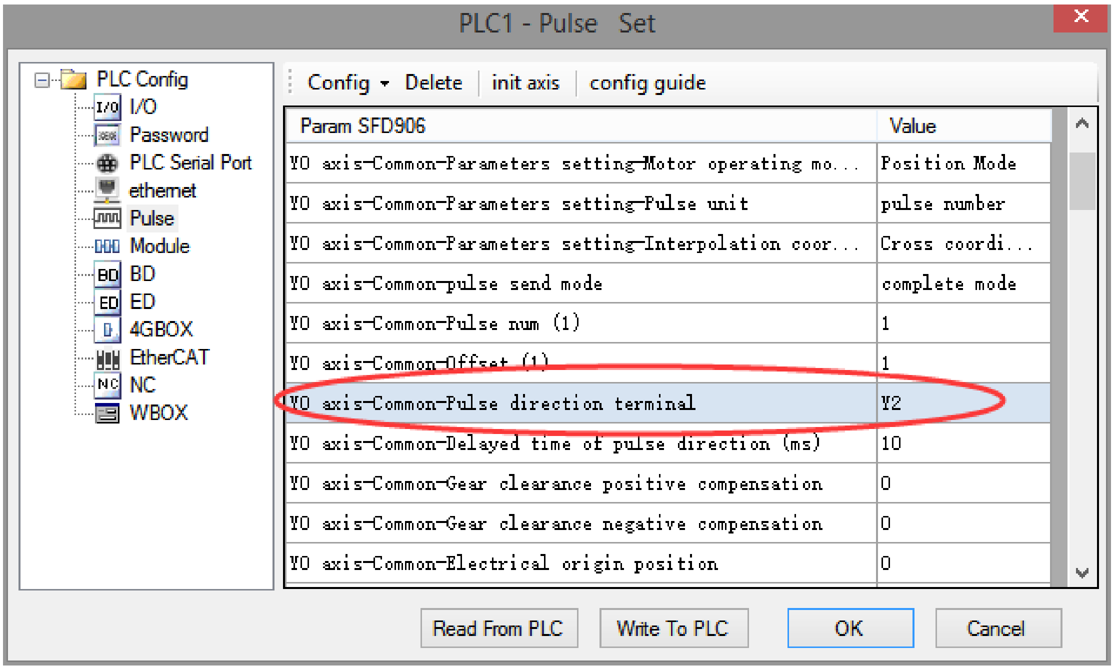
In the sample program, all the system parameters used in the pulse instructions are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

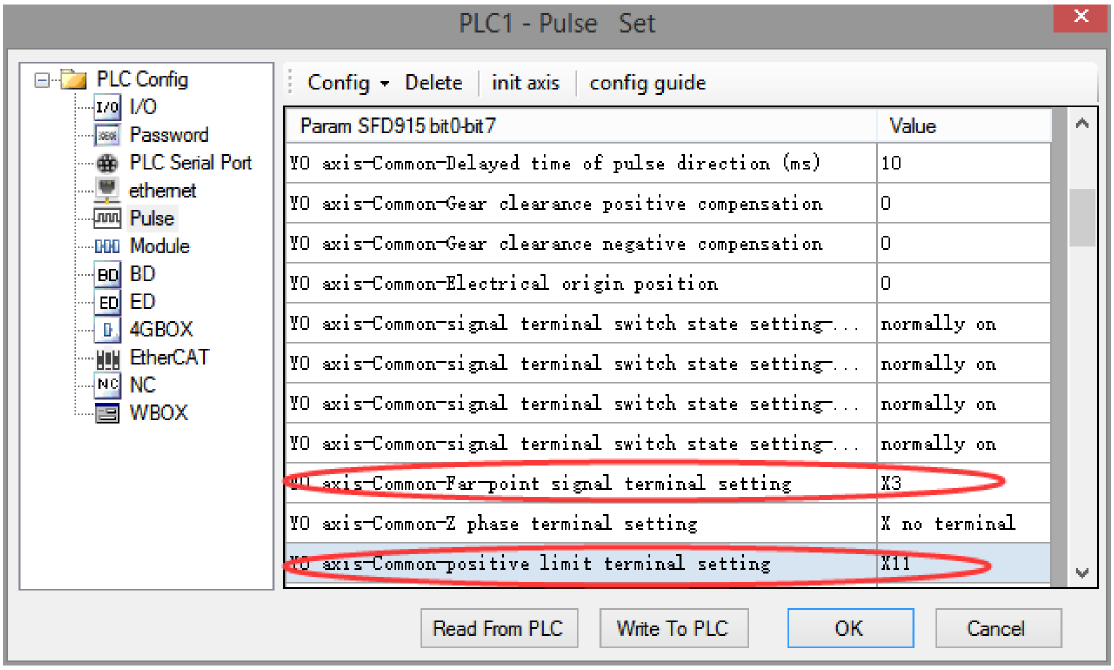


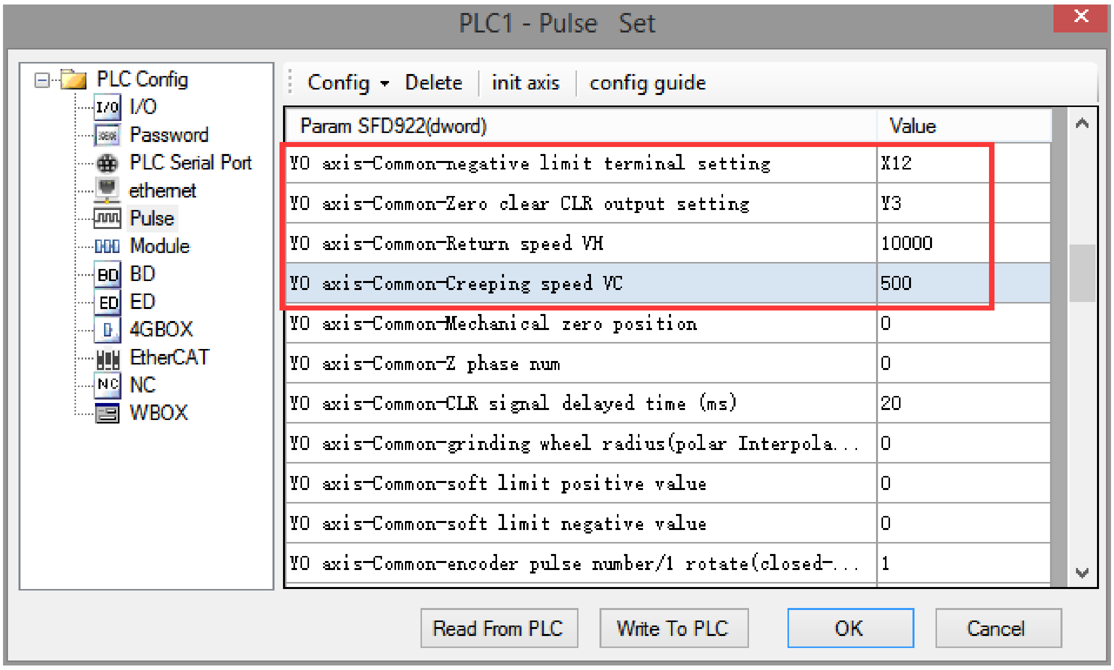
Click config, then select Y0 axis.

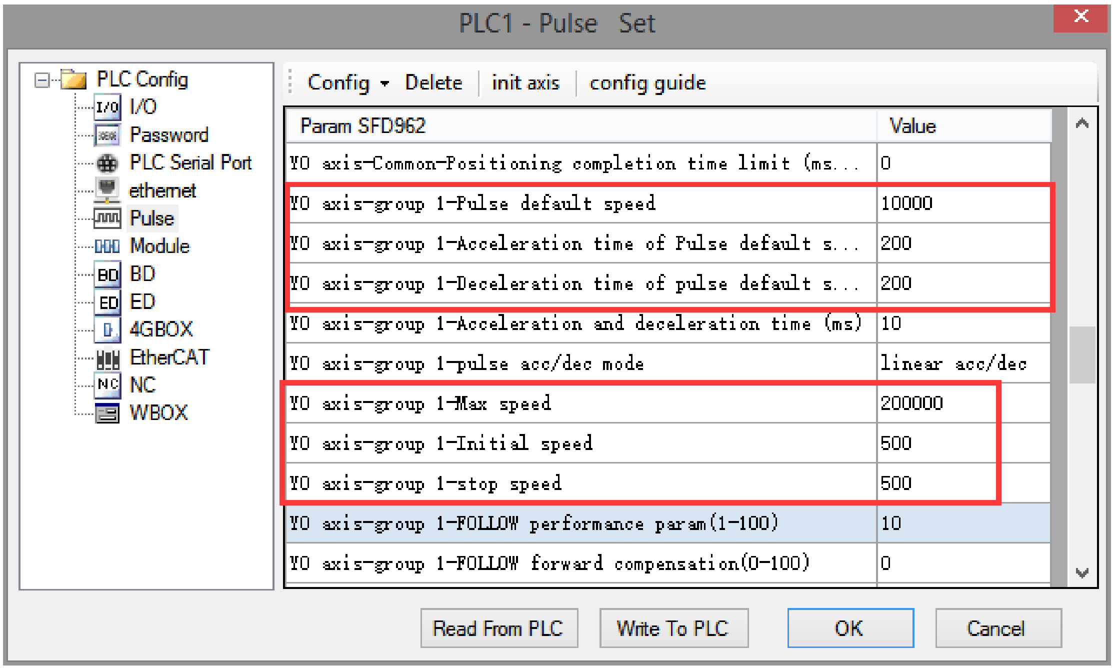


In the parameter configuration table, configure as follows (circled parameters need to be modified):



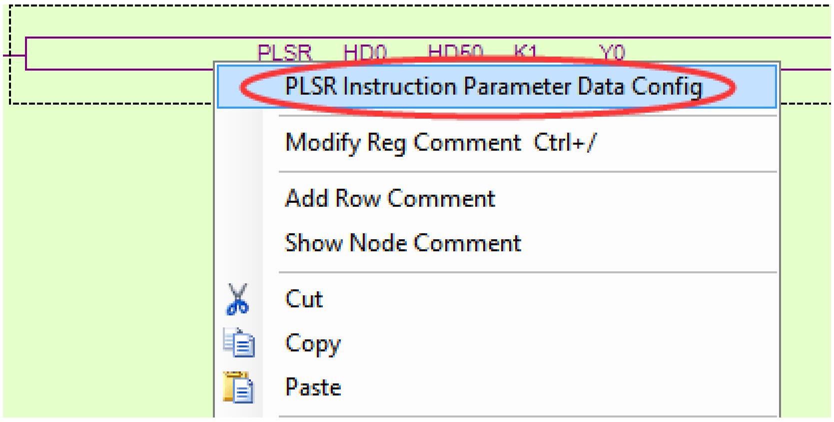

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD922(dword)
Value
Y0 axis-Common-negative limit terminal setting
X12
Y0 axis-Common-Zero clear CLR output setting
Y3
Y0 axis-Common-Return speed VH
10000
Y0 axis-Common-Creeping speed VC
500
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
20
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...
1
Read From PLC
Write To PLC
OK
Cancel

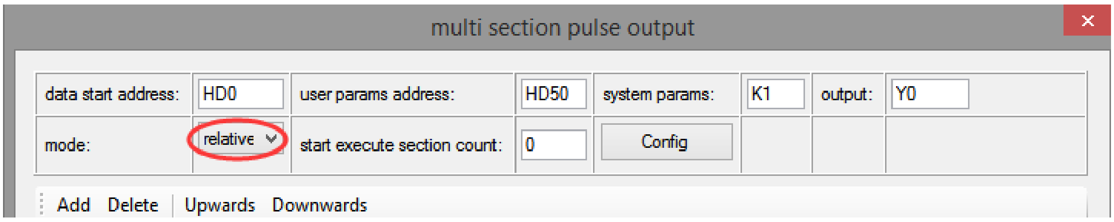


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

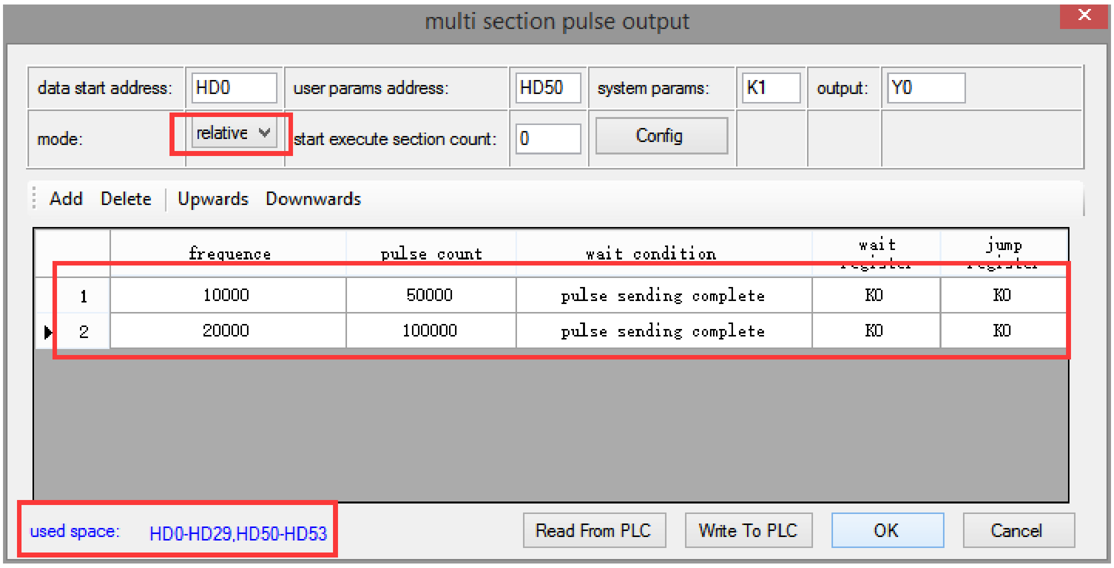
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:

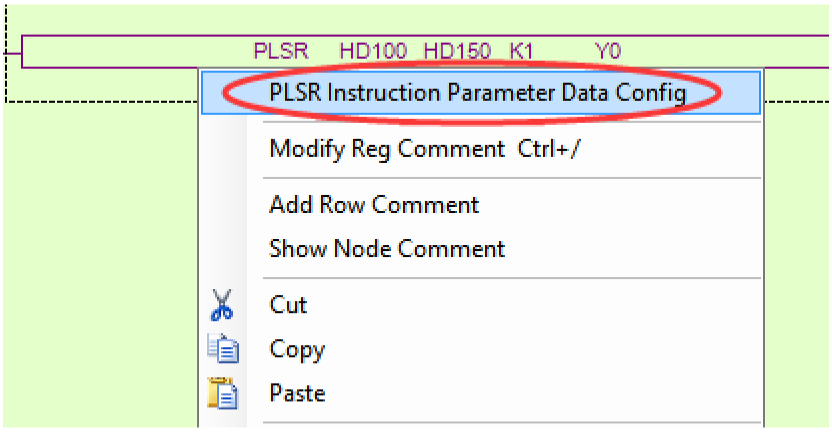


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

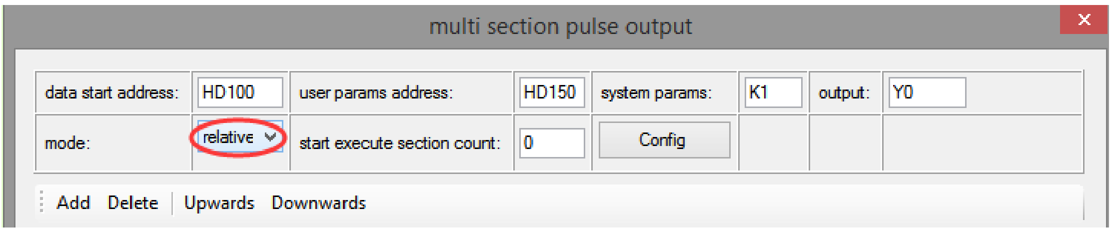


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

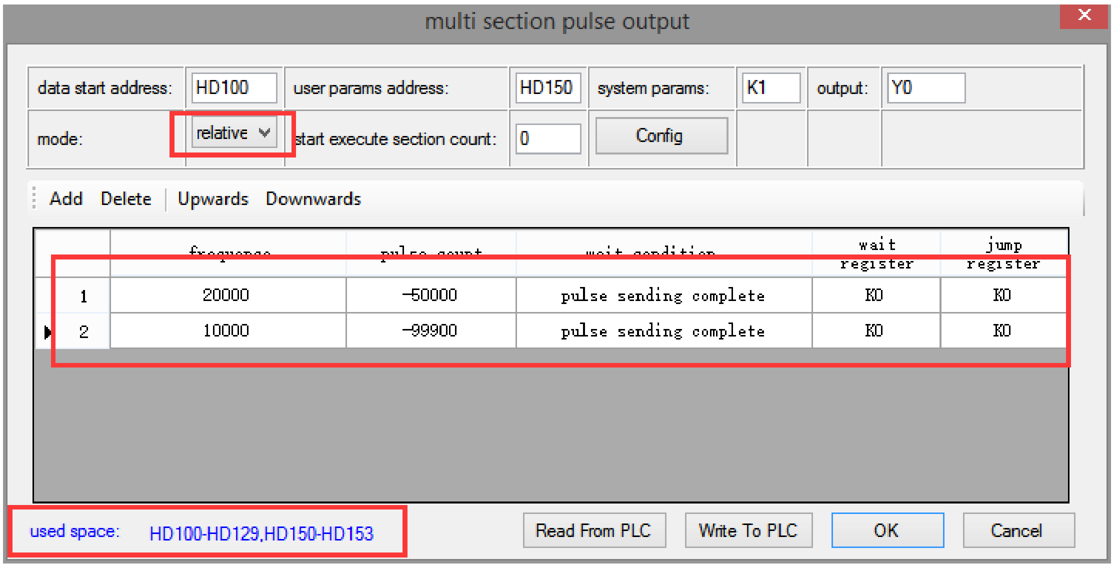
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:



After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of reverse rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

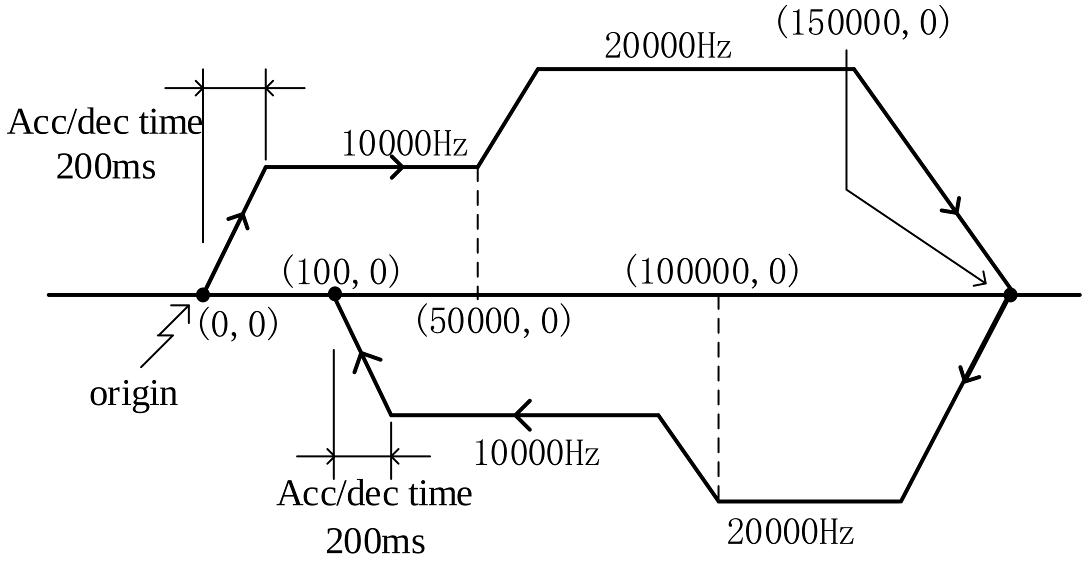


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

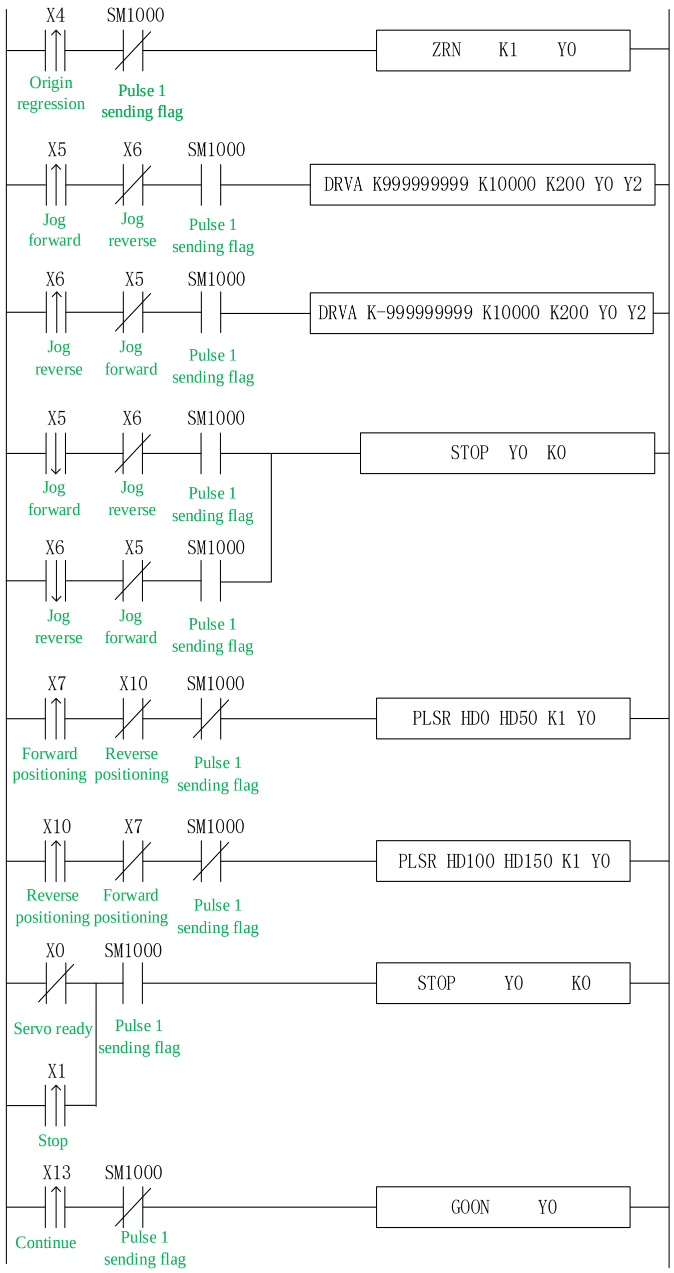
After downloading the program, power off the PLC and then re-energize it.
Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

1-5-6. Forward reverse rotation mulsti-segment sequential control program 【DRVI, DRVA, PLSR, ZRN】

Example 1: According to the following figure, multi-segment absolute positioning mode is adopted.



Firstly, make the ladder chart as follows:



//mechanical origin regression (use group 1 parameters)

//jog forward control, the speed is 10000Hz

//jog reverse control, the speed is -10000Hz

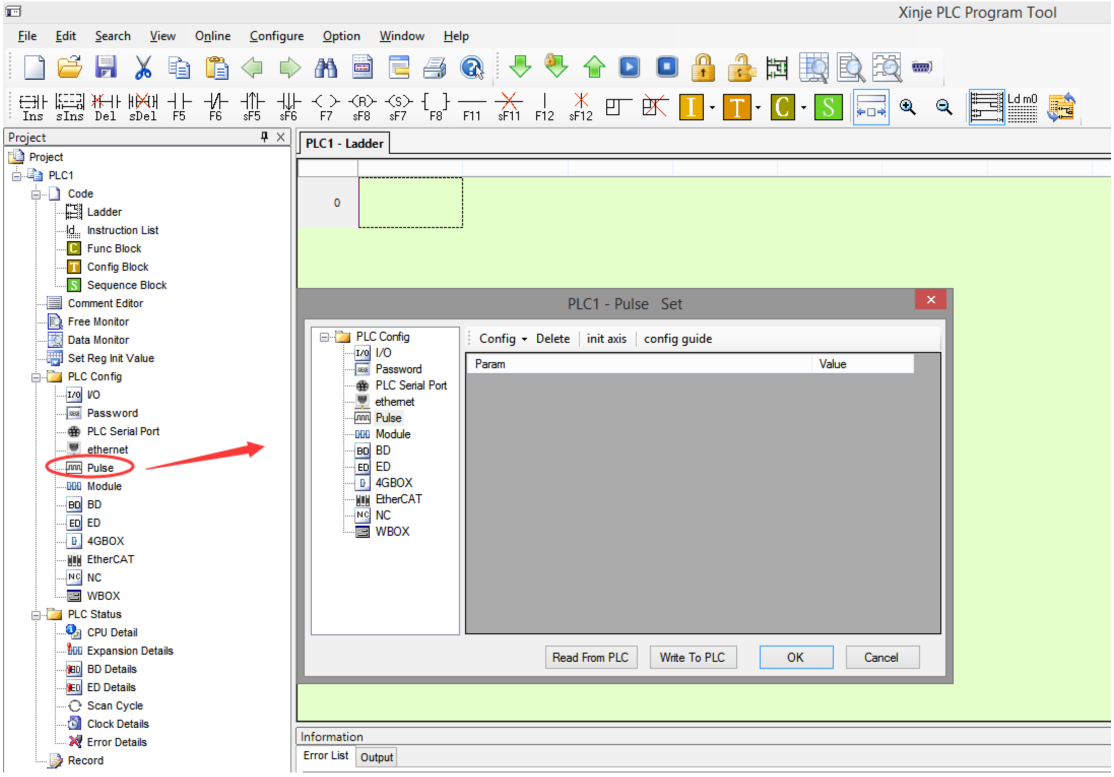
//forward positioning control (use group 1 parameters)

//reverse positioning control (use group 1 parameters)

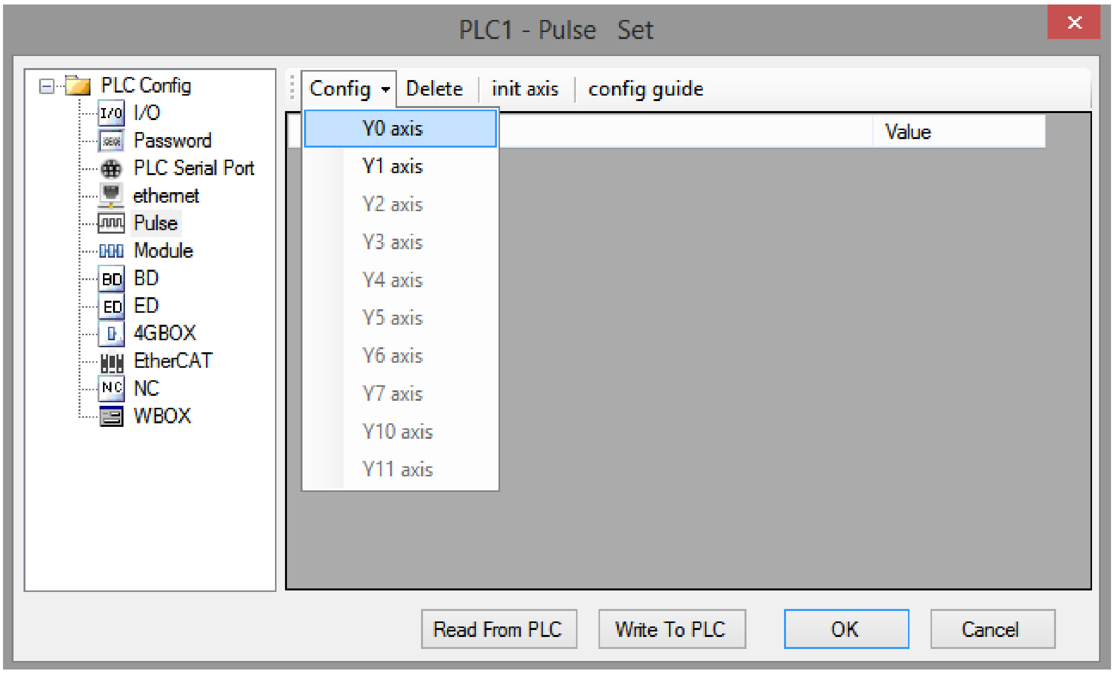
// pulse stop sending (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

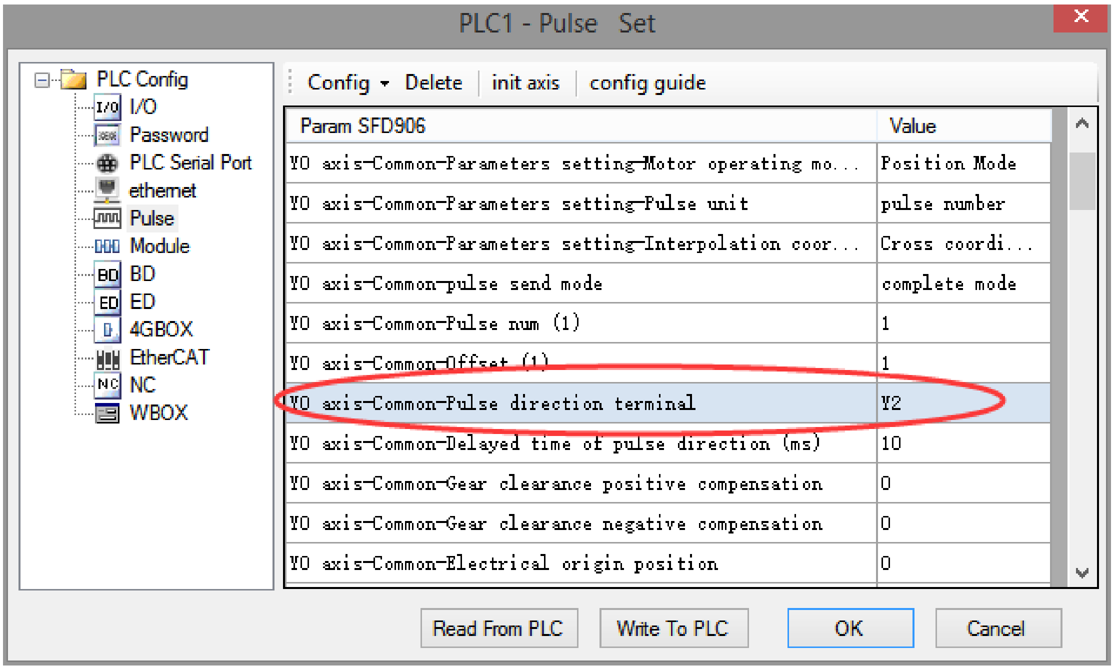
In the sample program, all the system parameters used in the pulse instructions (except DRVA, DRVI) are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

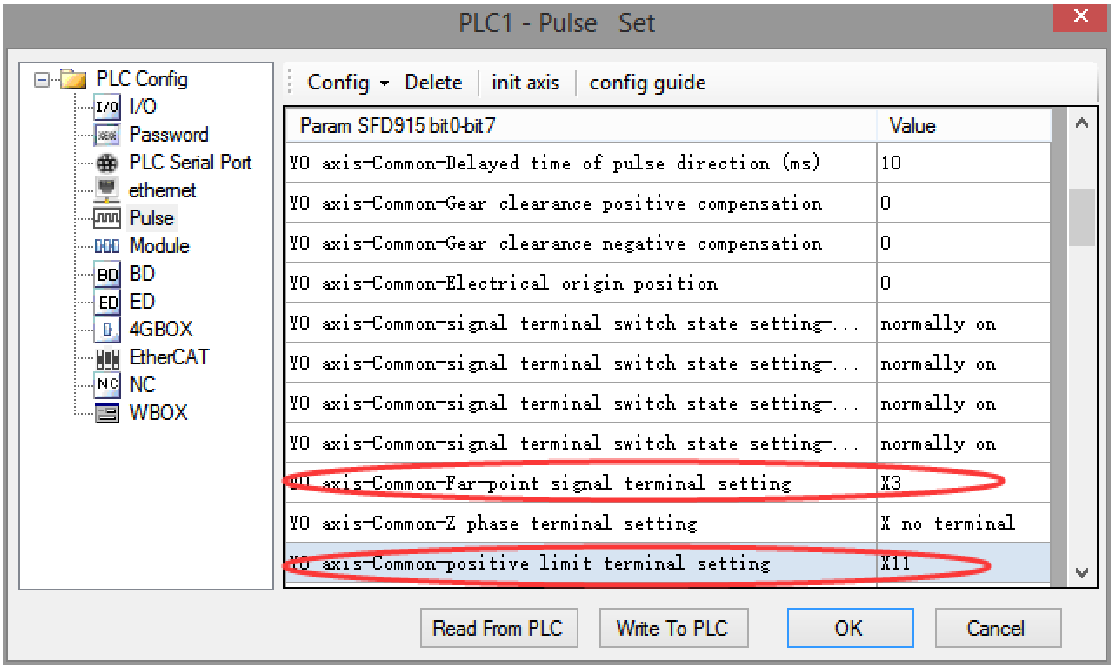


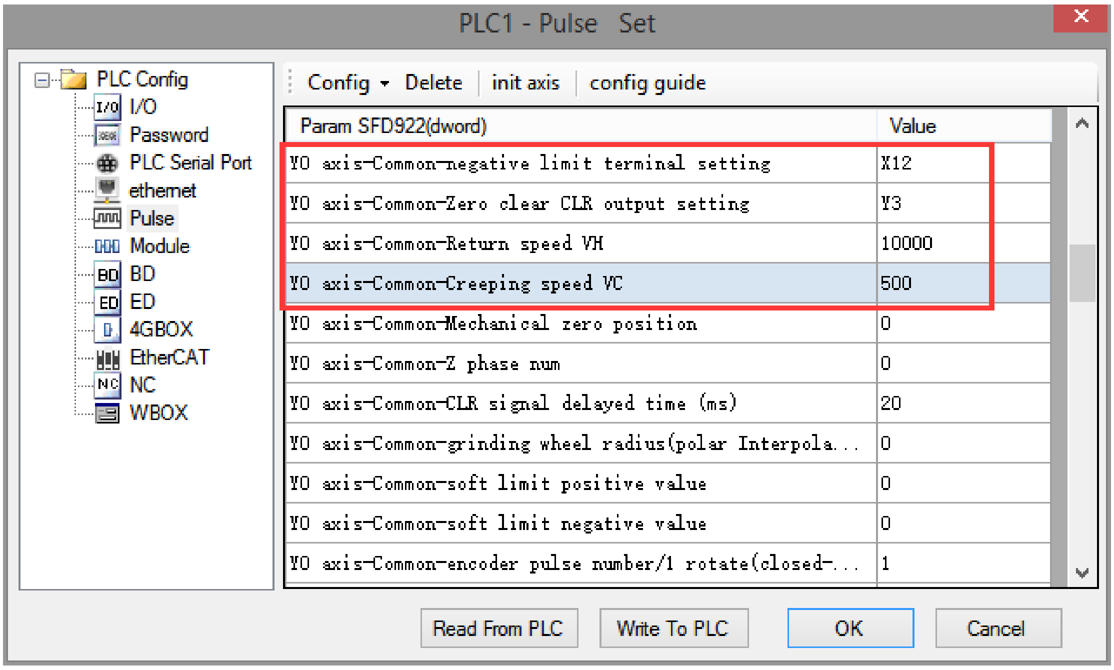
Click config, then select Y0 axis.

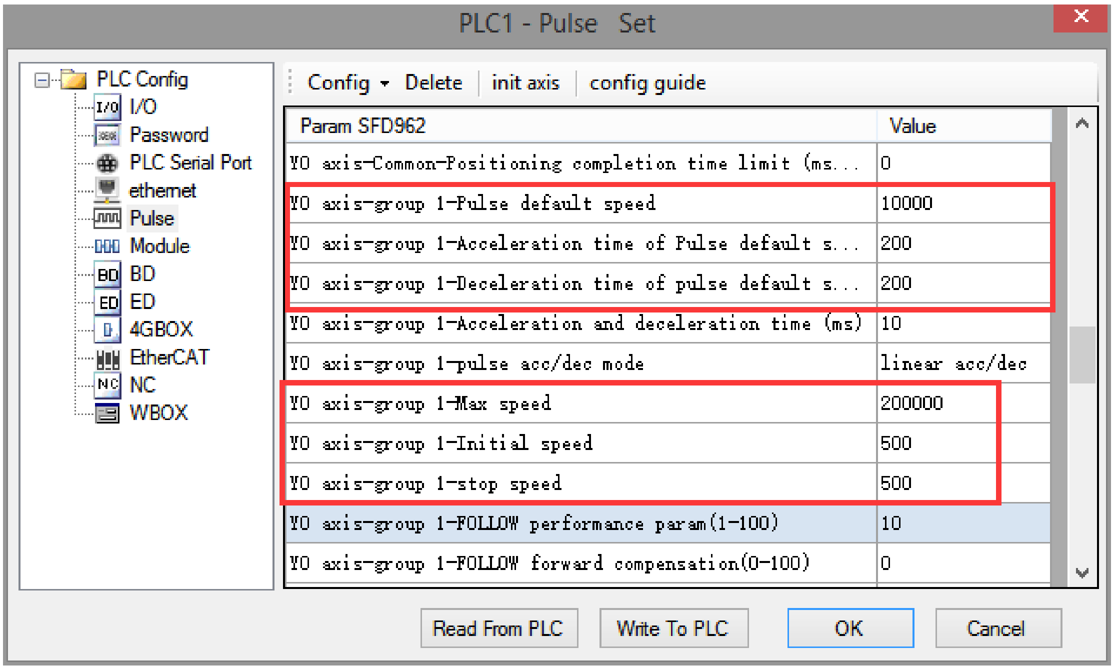


In the parameter configuration table, configure as follows (circled parameters need to be modified):


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD906
Value
Y0 axis-Common-Parameters setting-Motor operating mo...
Position Mode
Y0 axis-Common-Parameters setting-Pulse unit
pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...
Cross coordi...
Y0 axis-Common-pulse send mode
complete mode
Y0 axis-Common-Pulse num (1)
1
Y0 axis-Common-Offset (1)
1
Y0 axis-Common-Pulse direction terminal
Y2
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Read From PLC
Write To PLC
OK
Cancel

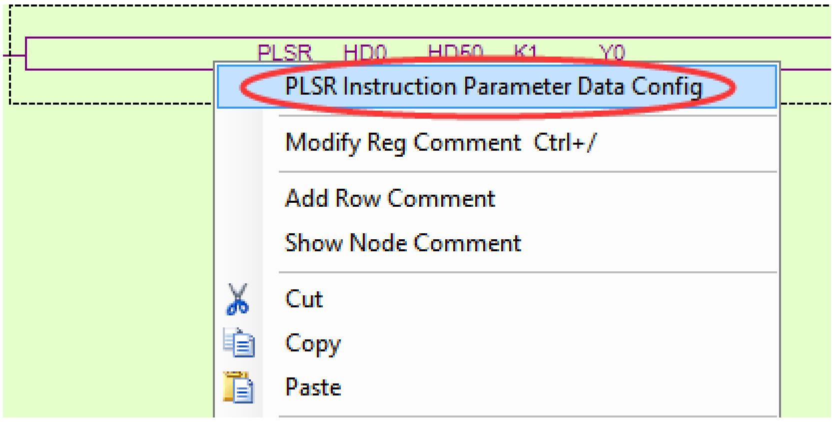

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel



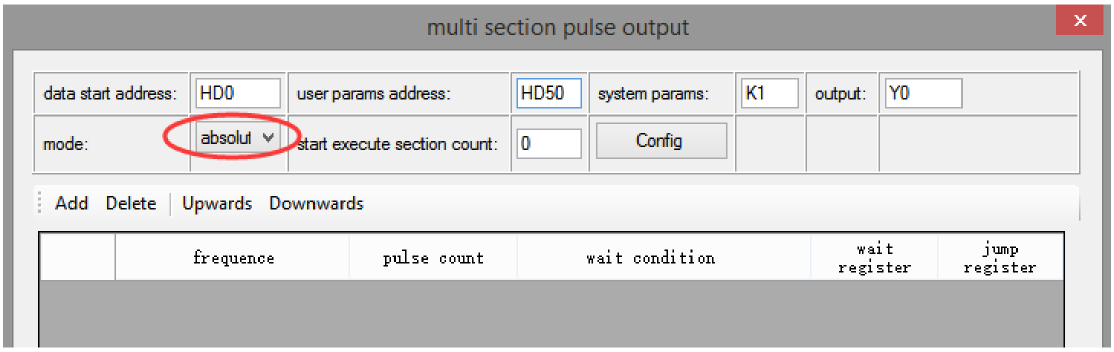


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

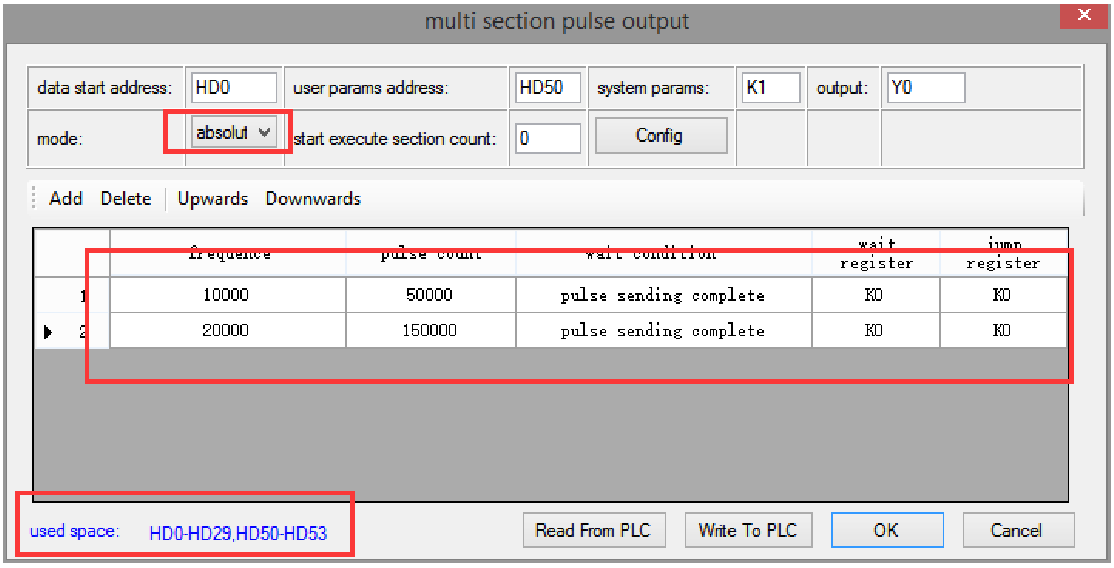
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:

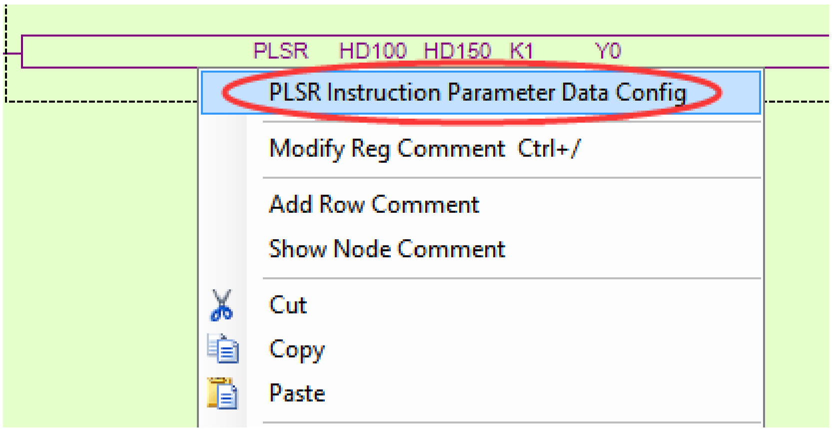


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

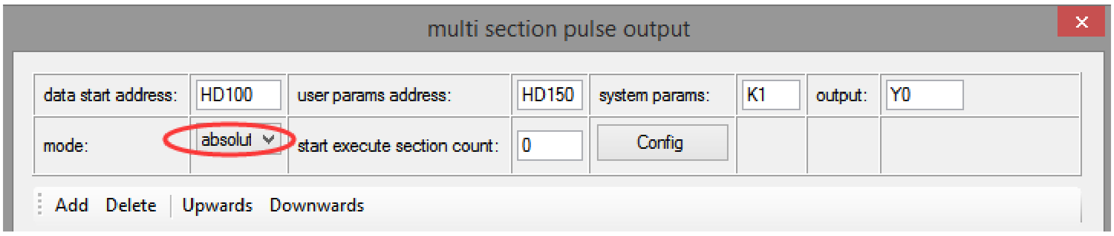


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

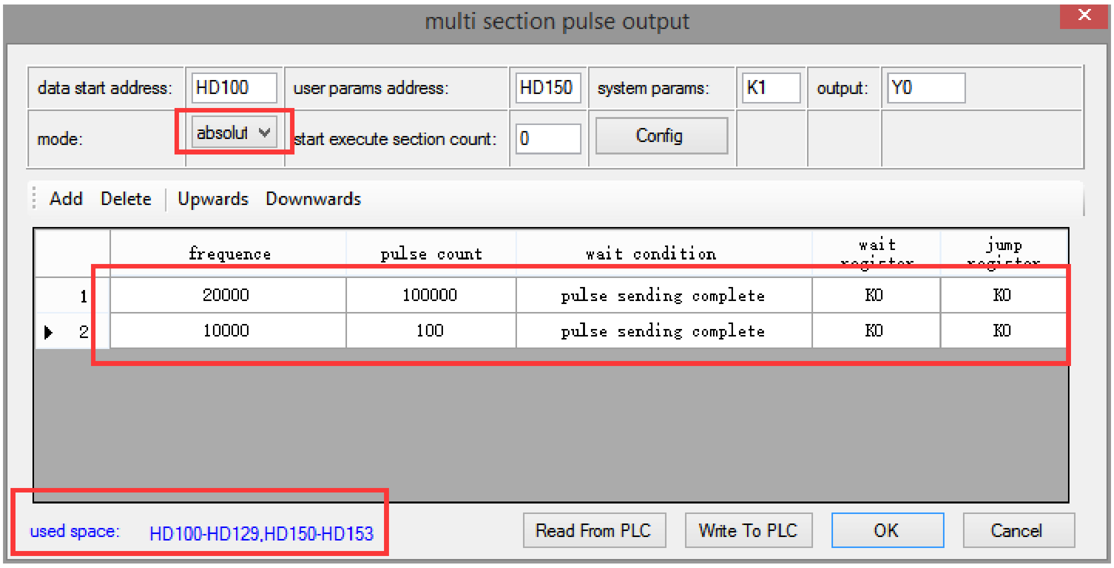
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:



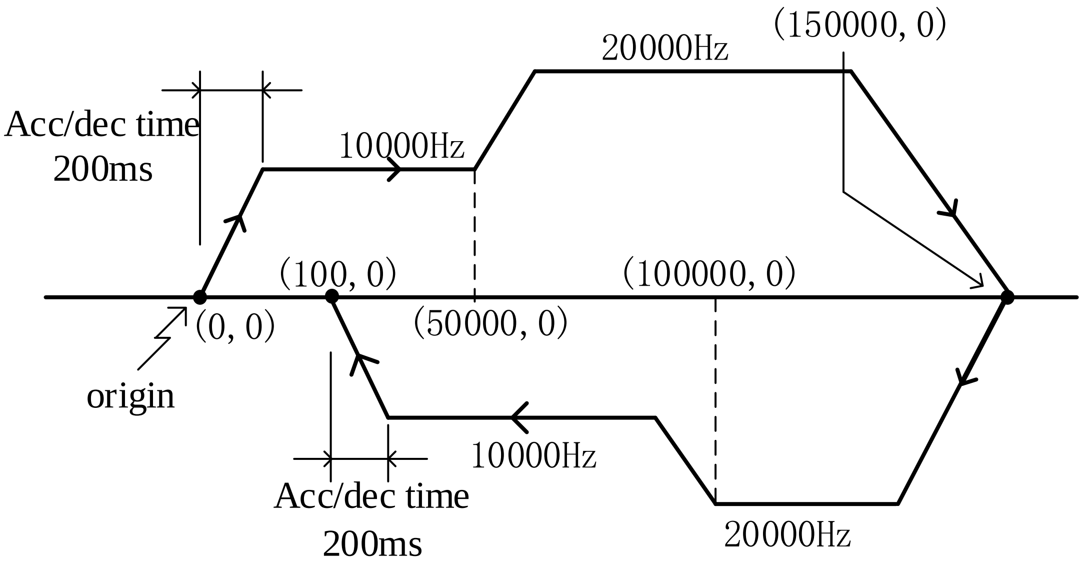
After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:



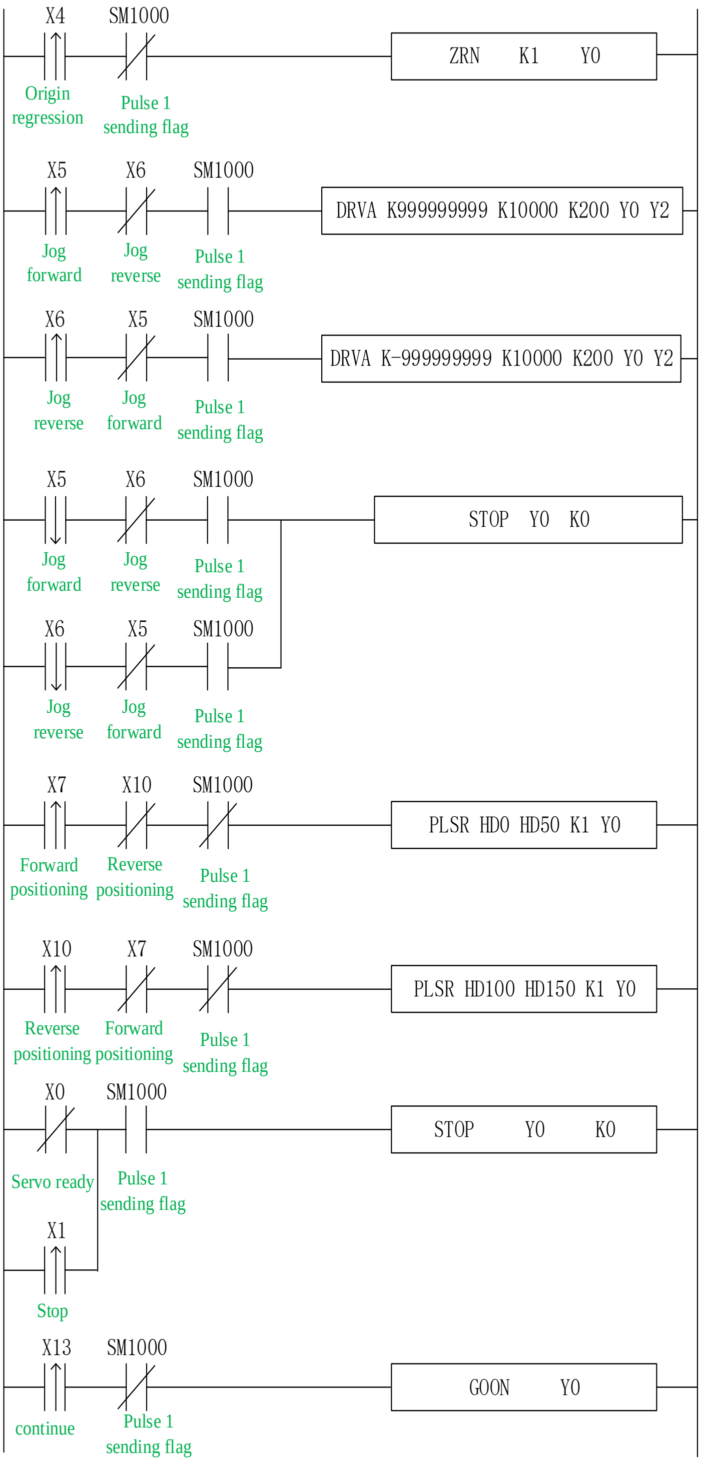
Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

After downloading the program, power off the PLC and then re-energize it.
Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

Example 2: According to the following figure, the relative multi-segment pulse positioning method is used.



Firstly, make the ladder chart as the follows:



//mechanical origin regression (use group 1 parameters)

//jog forward control, the speed is 10000Hz

//jog reverse control, the speed is -10000Hz

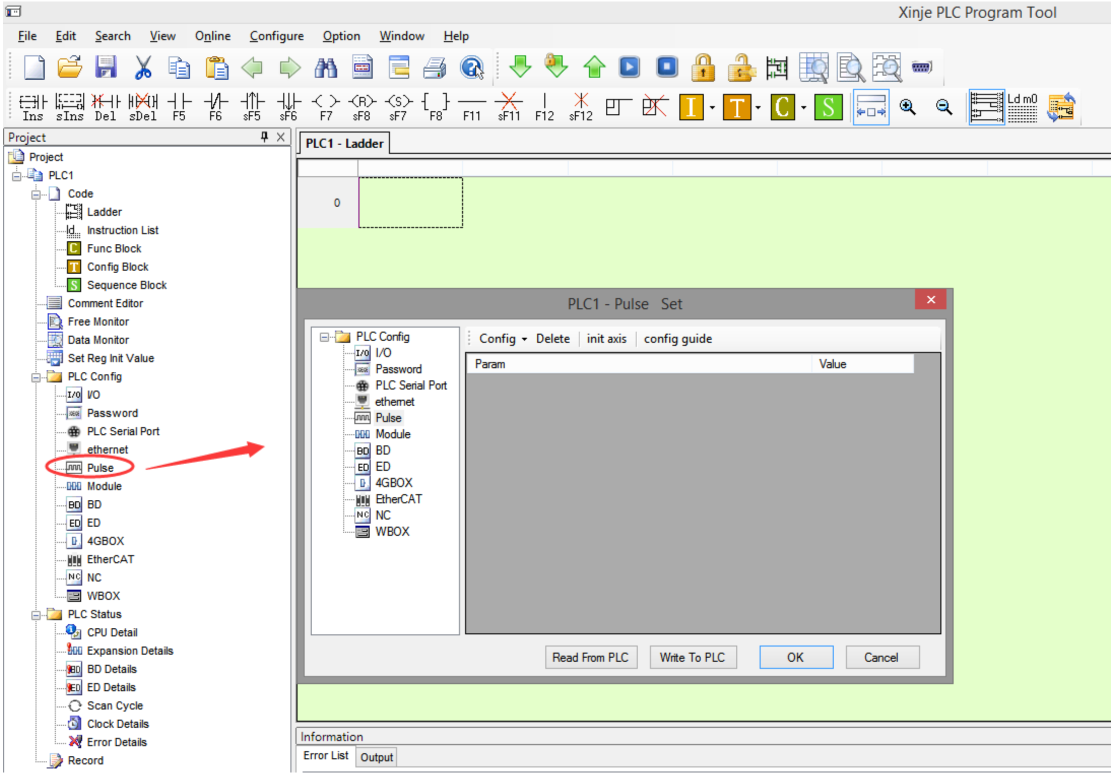
//forward positioning control (use group 1 parameters)

//reverse positioning control (use group 1 parameters)

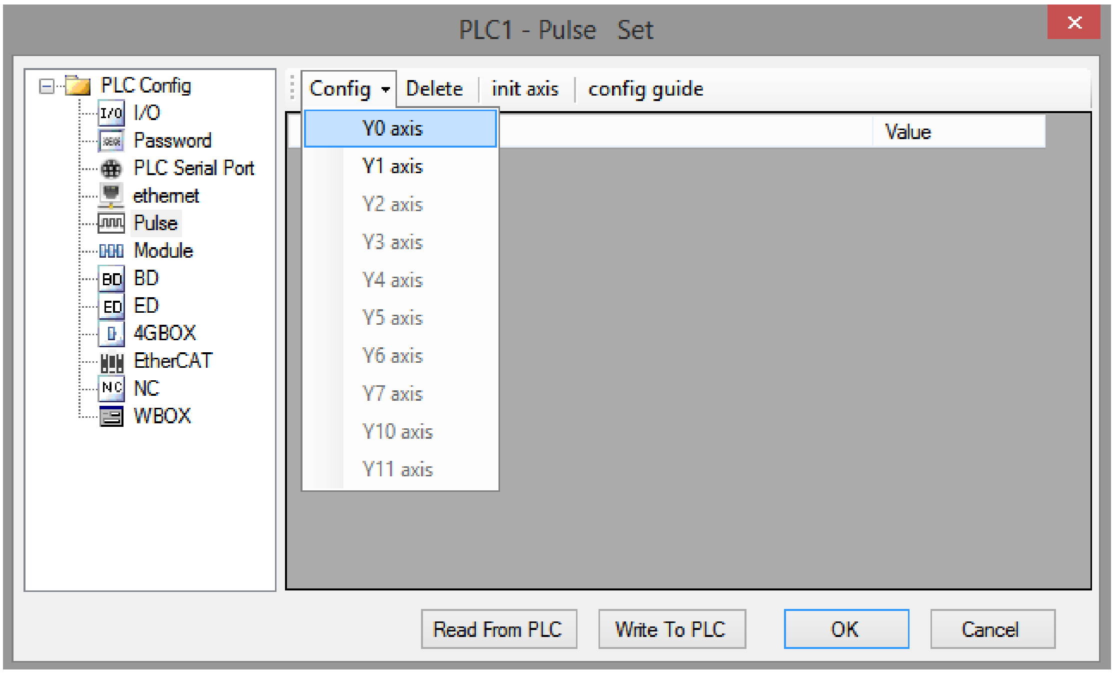
// pulse stop sending (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

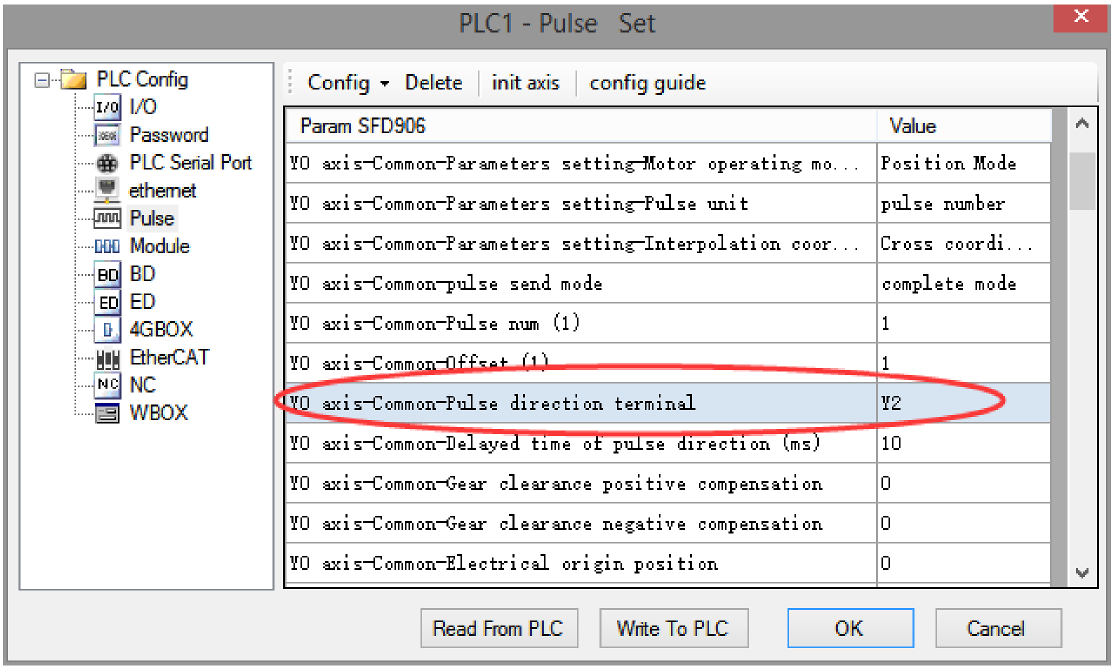
In the sample program, all the system parameters used in the pulse instructions (except DRVA, DRVI) are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

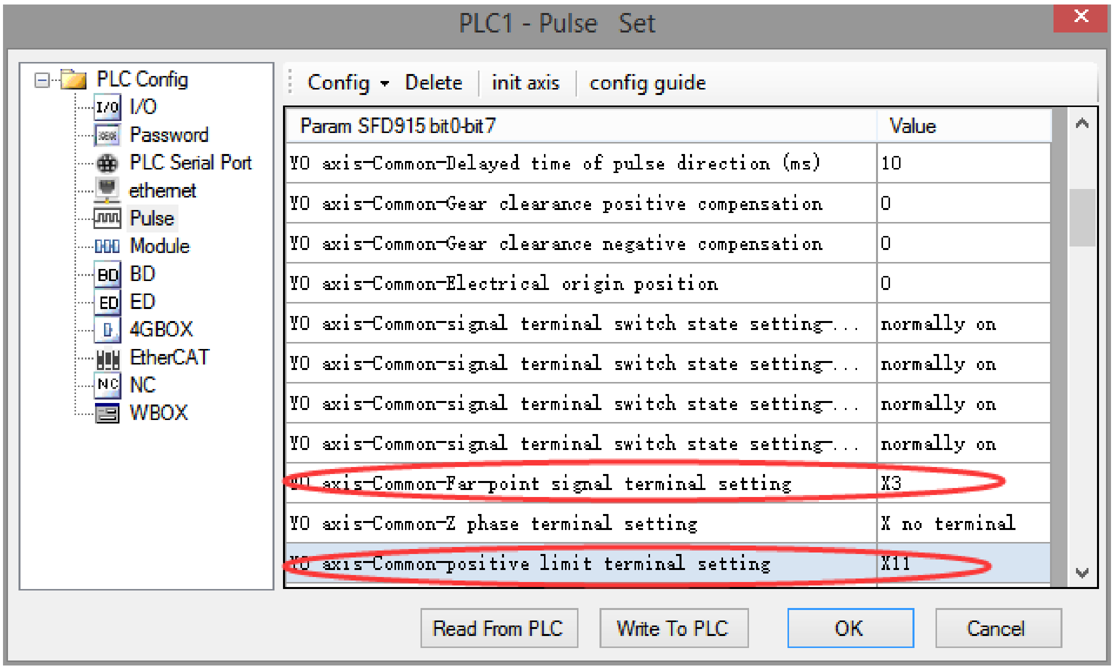


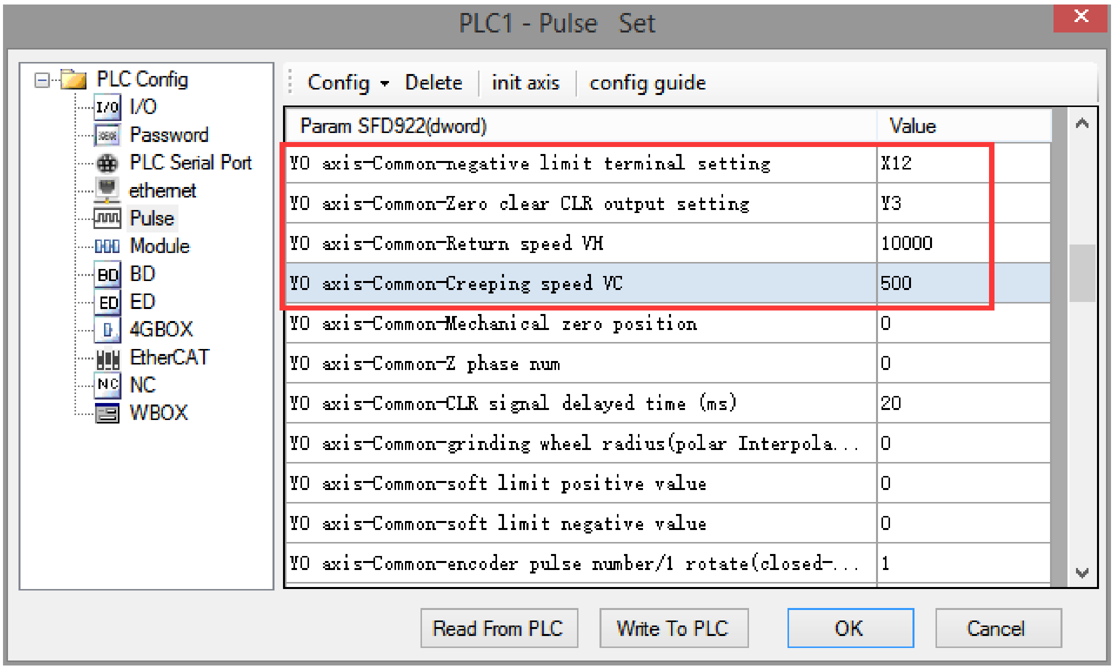
Click config, then select Y0 axis.

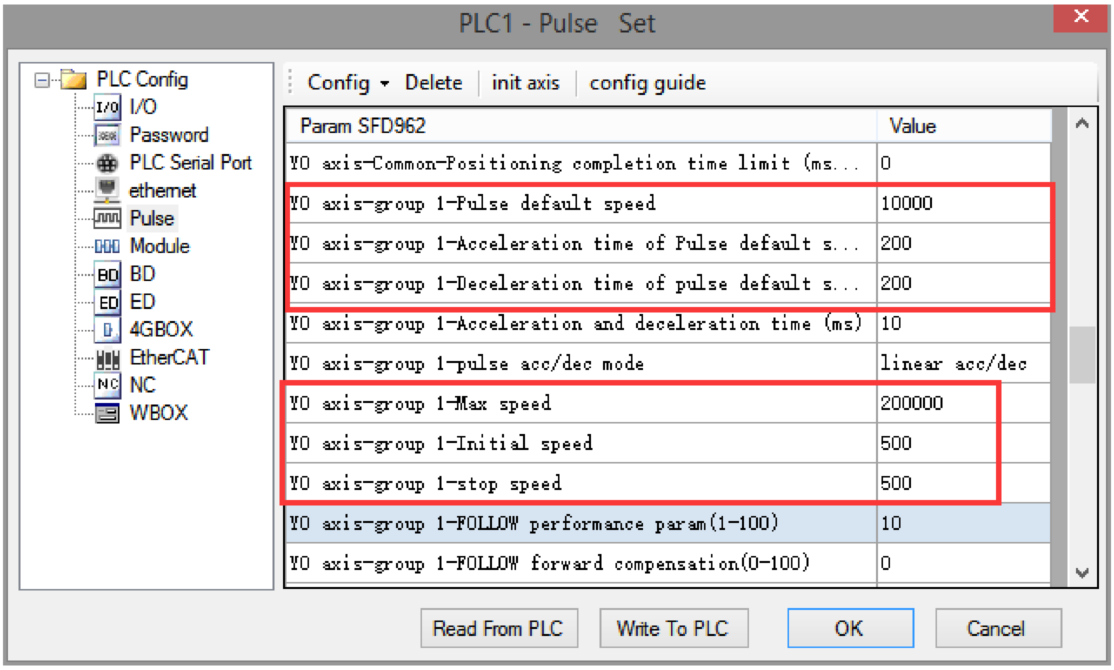


In the parameter configuration table, configure as follows (circled parameters need to be modified):


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD906
Value
Y0 axis-Common-Parameters setting-Motor operating mo...
Position Mode
Y0 axis-Common-Parameters setting-Pulse unit
pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...
Cross coordi...
Y0 axis-Common-pulse send mode
complete mode
Y0 axis-Common-Pulse num (1)
1
Y0 axis-Common-Offset (1)
1
Y0 axis-Common-Pulse direction terminal
Y2
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Read From PLC
Write To PLC
OK
Cancel

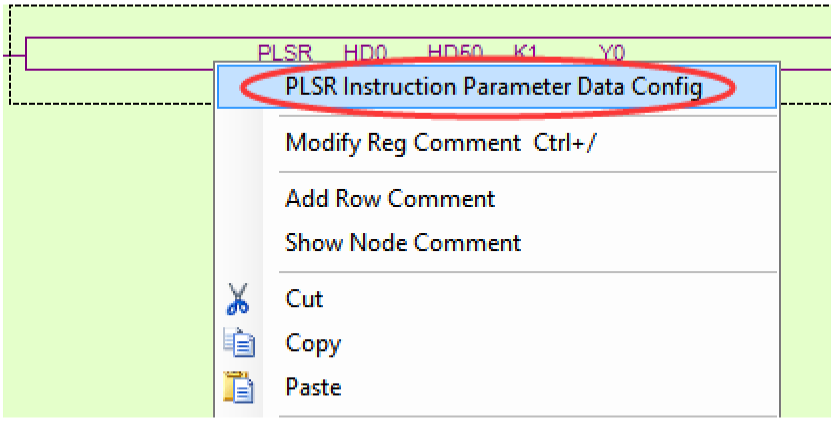

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel



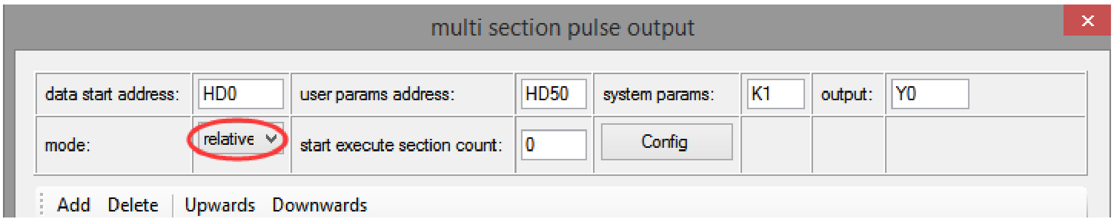


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

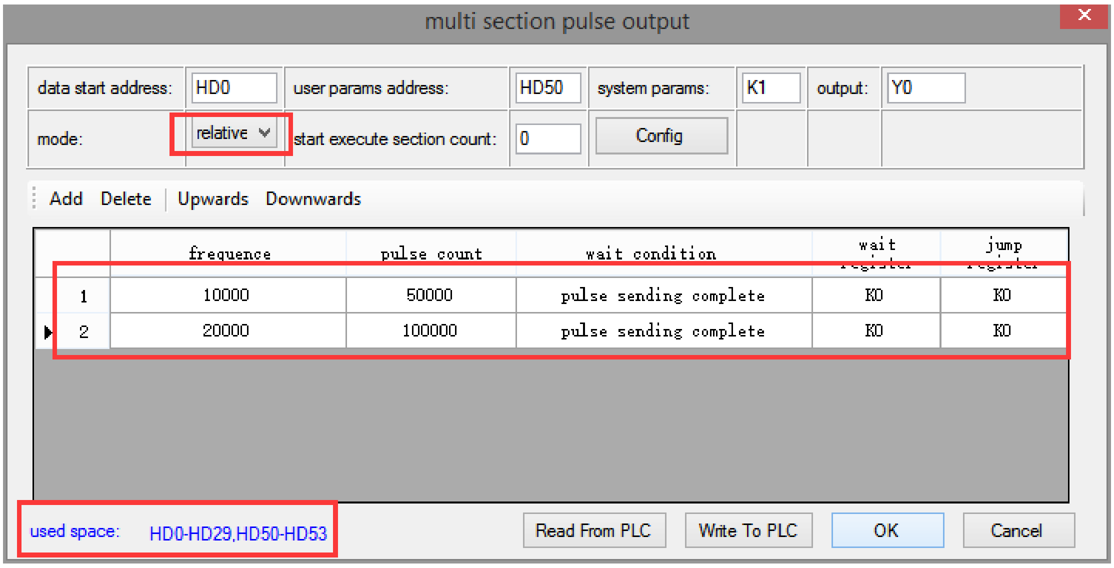
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:

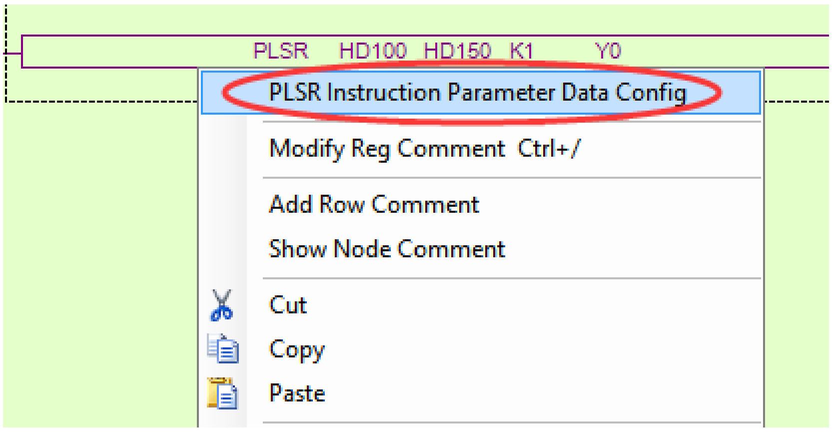


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

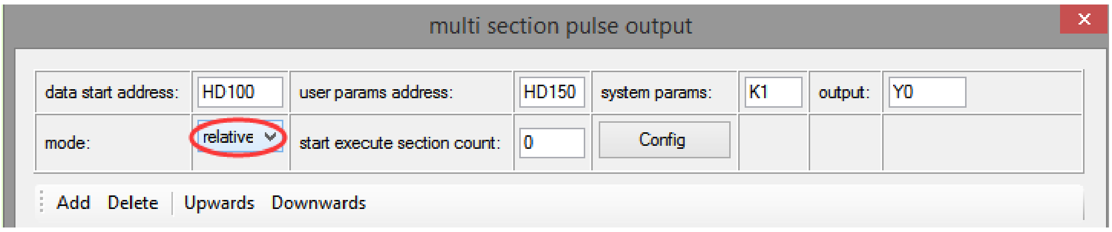


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

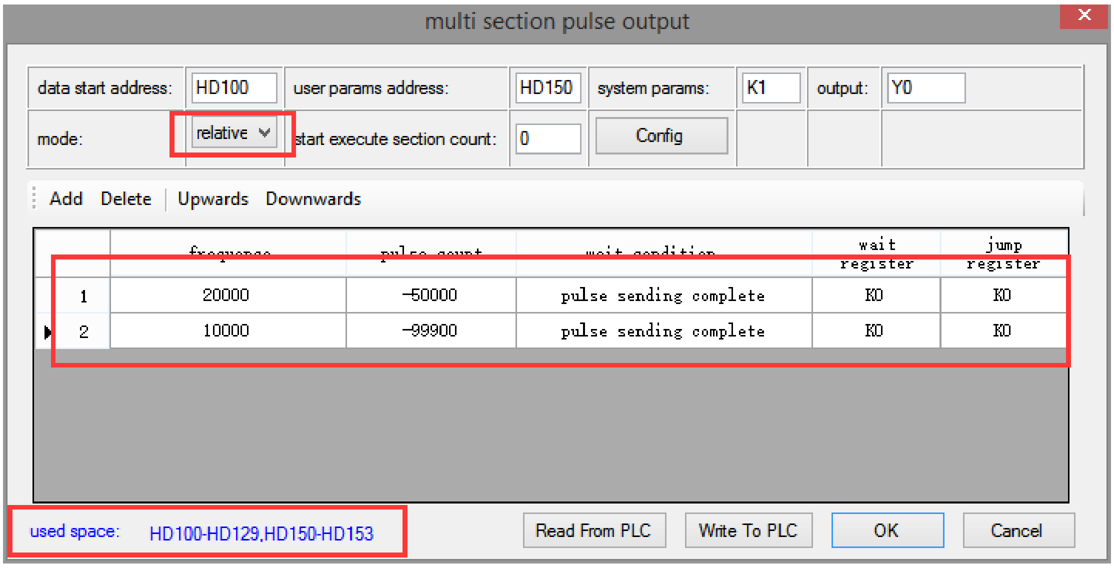
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:



After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of reverse rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:



Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

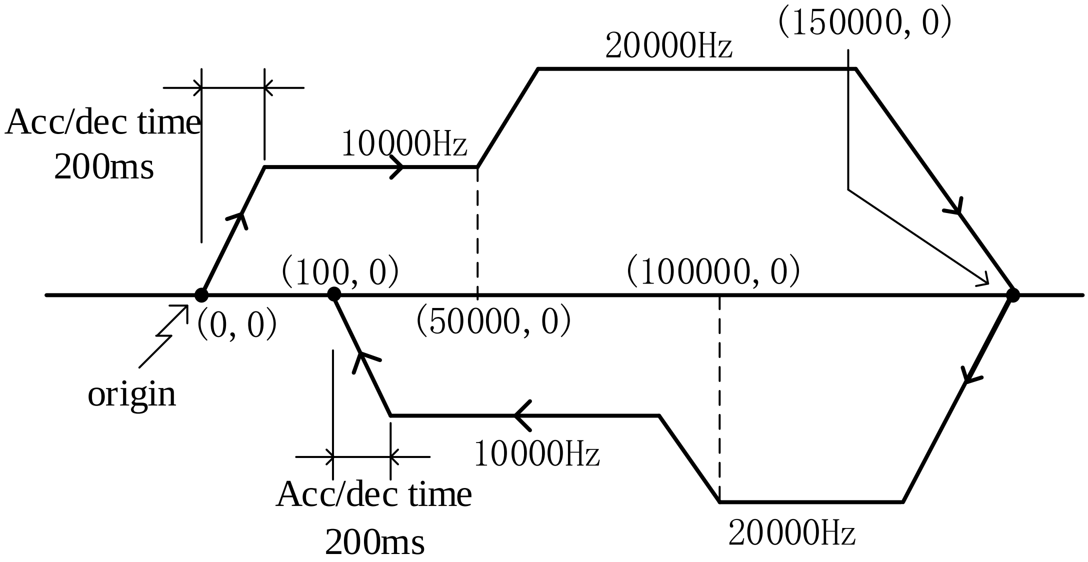
After downloading the program, power off the PLC and then re-energize it.

Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of

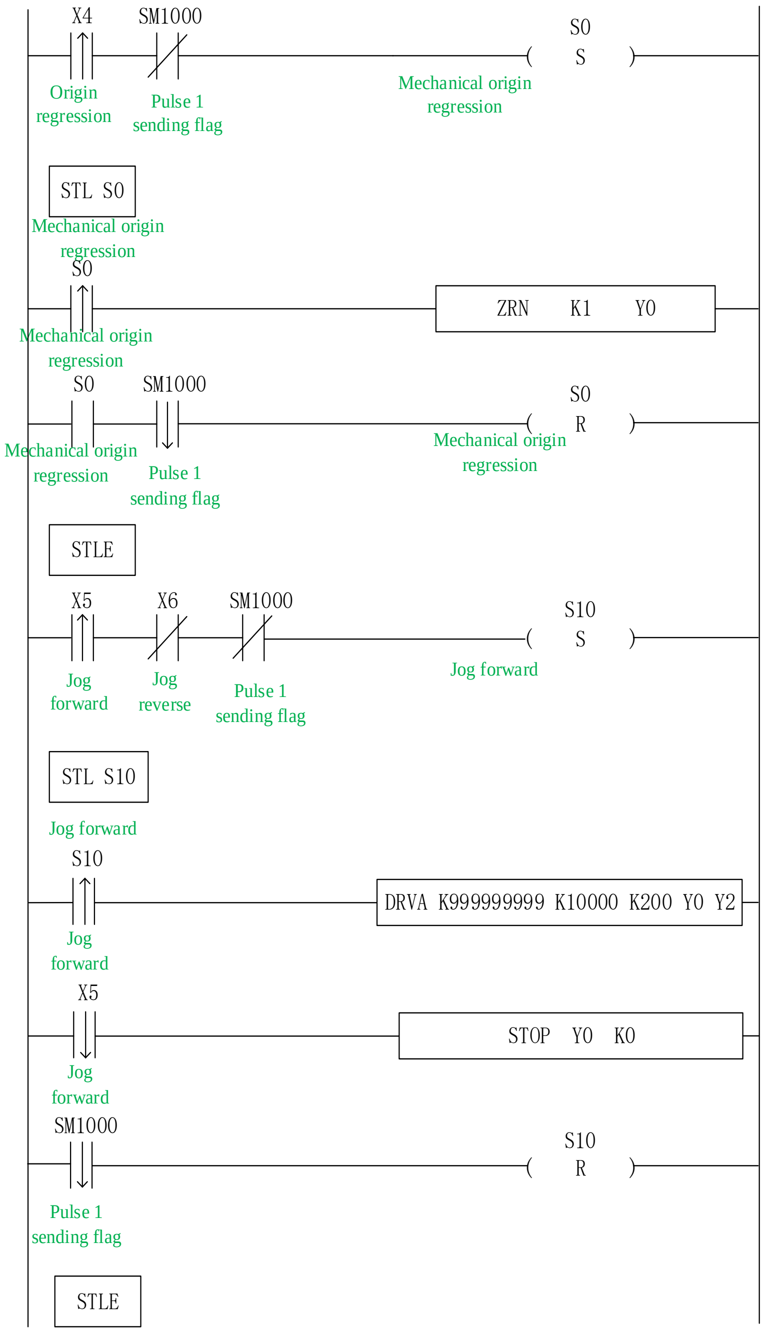
ZRN, PLSF, DRVI and DRVA instructions.

1-5-7. Forward and reverse rotation multi-segment process program 【DRVI, DRVA, PLSR, ZRN】

Example 1: According to the following figure, multi-segment absolute positioning mode is adopted.

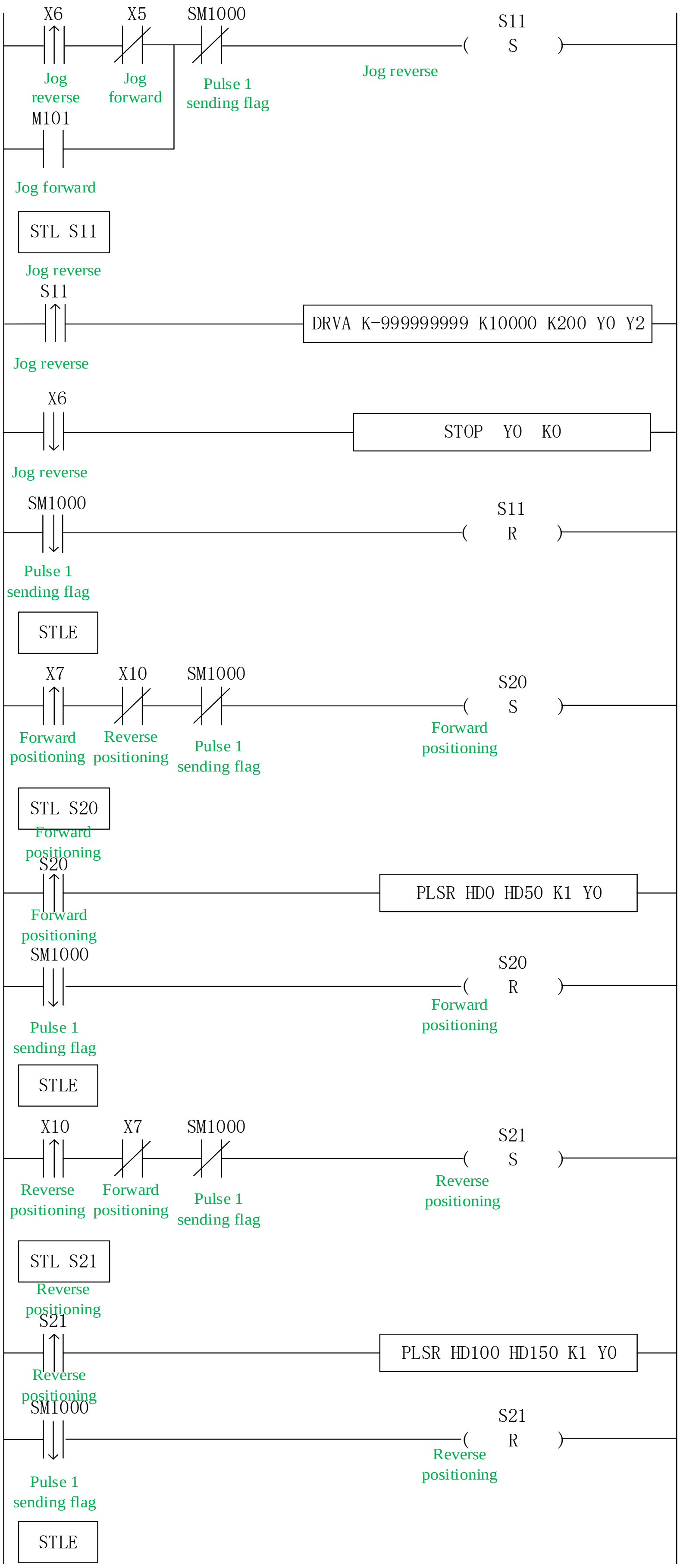


Firstly, make the ladder chart as follows:



//mechanical origin regression (use group 1 parameter)

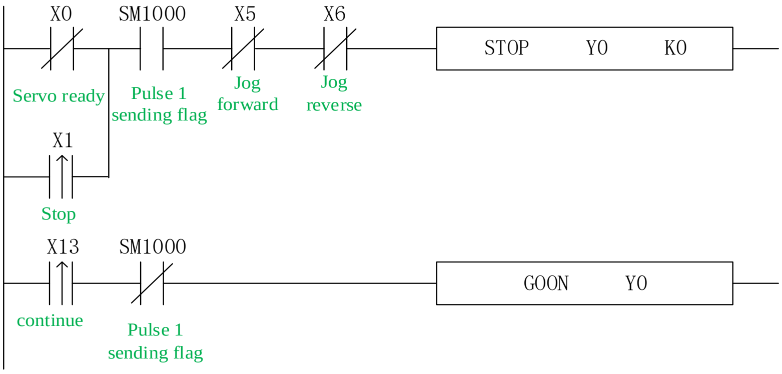
//jog forward control, the speed is 10000Hz



//jog reverse control, the speed is -10000Hz

//forward positioning control (use group 1 parameters)

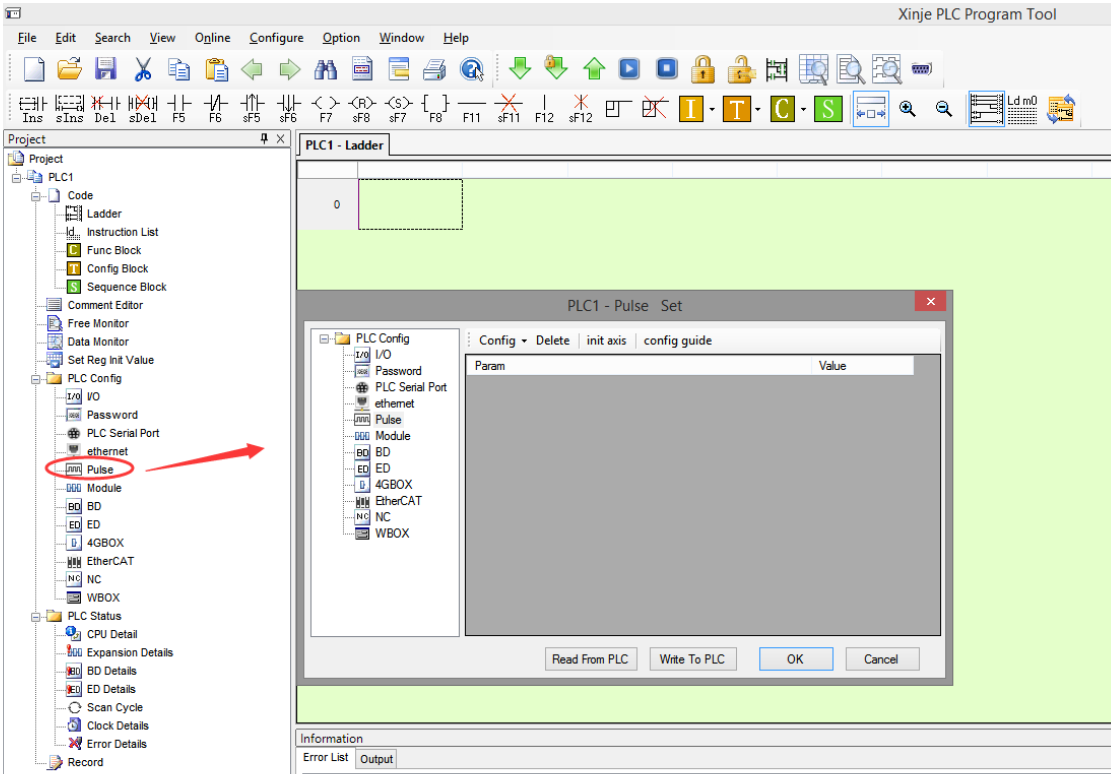
//reverse positioning control (use group 1 parameters)



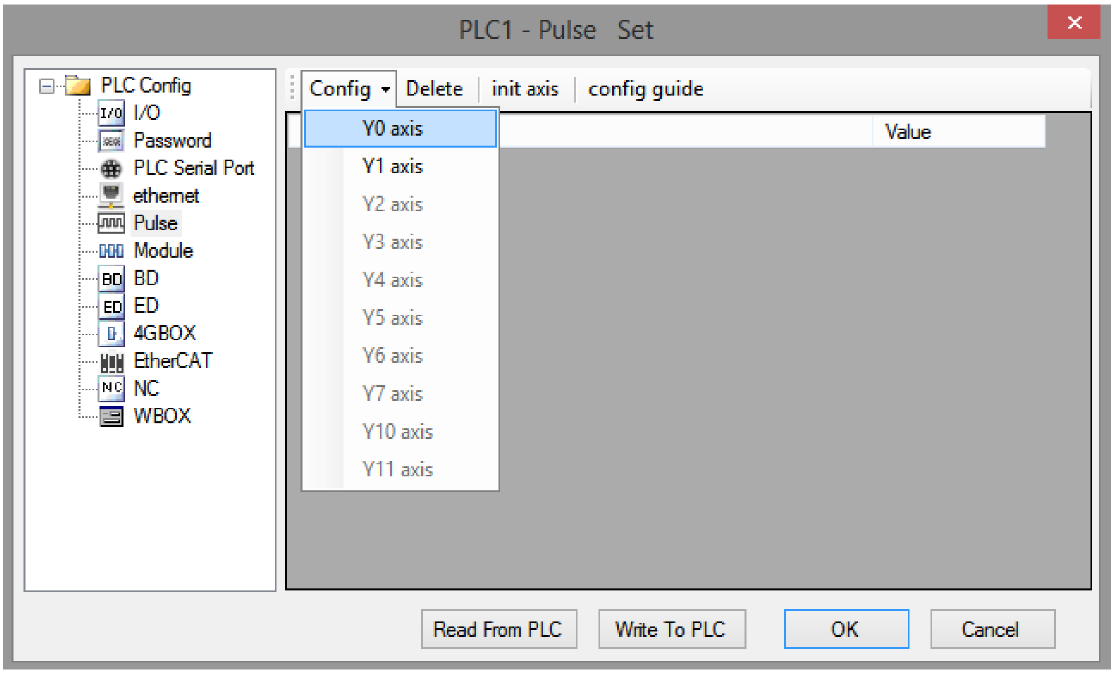
//pulse stop sending (slow stop)

// The execution pulse is not finished and the remaining pulse is continued to be sent out.

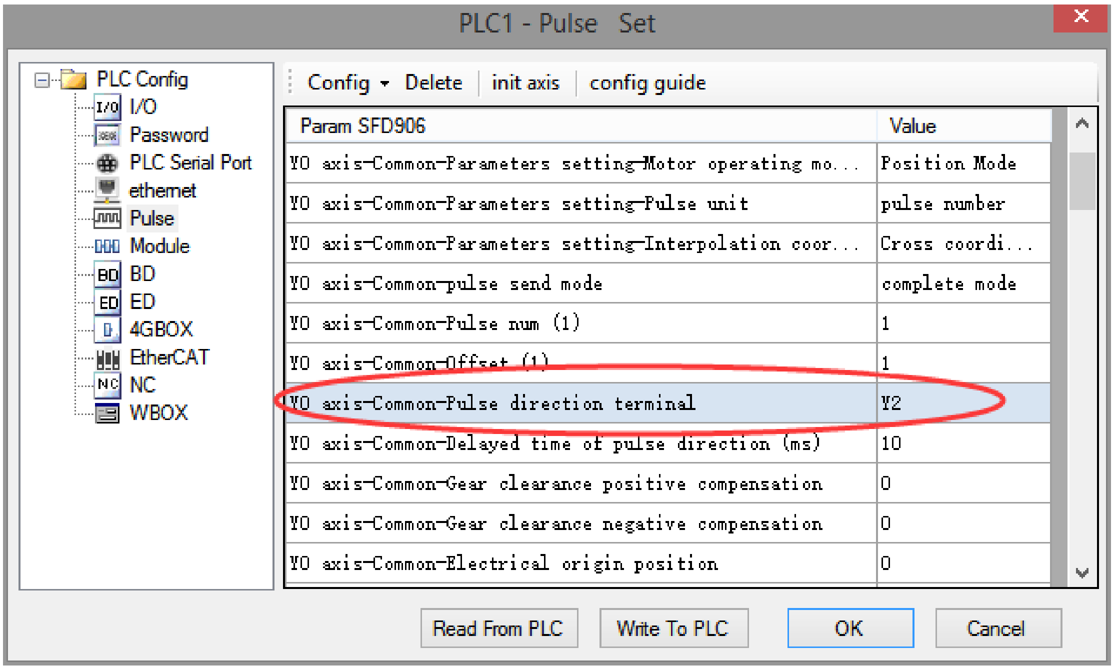
In the sample program, all the system parameters used in the pulse instructions (except DRVA, DRVI) are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

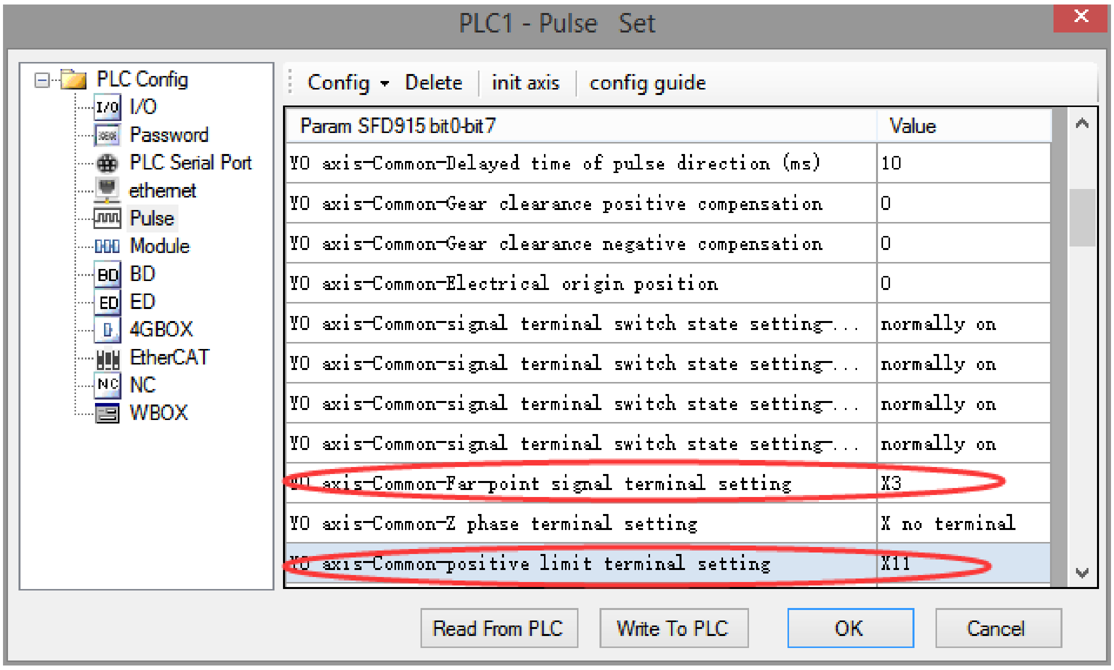


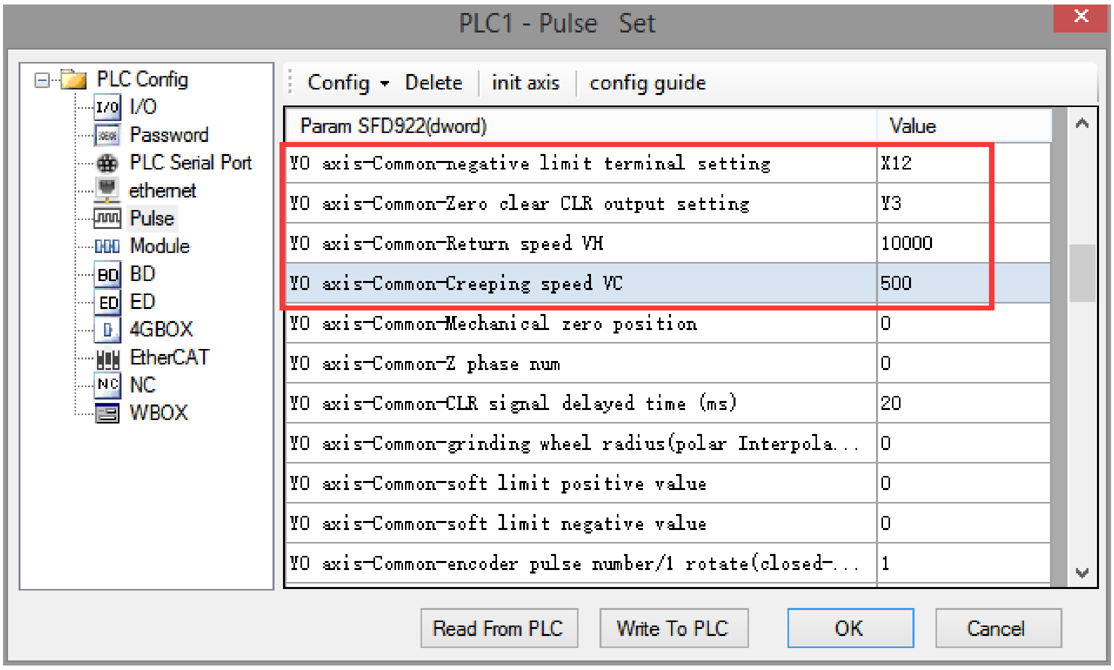
Click config, then select Y0 axis.

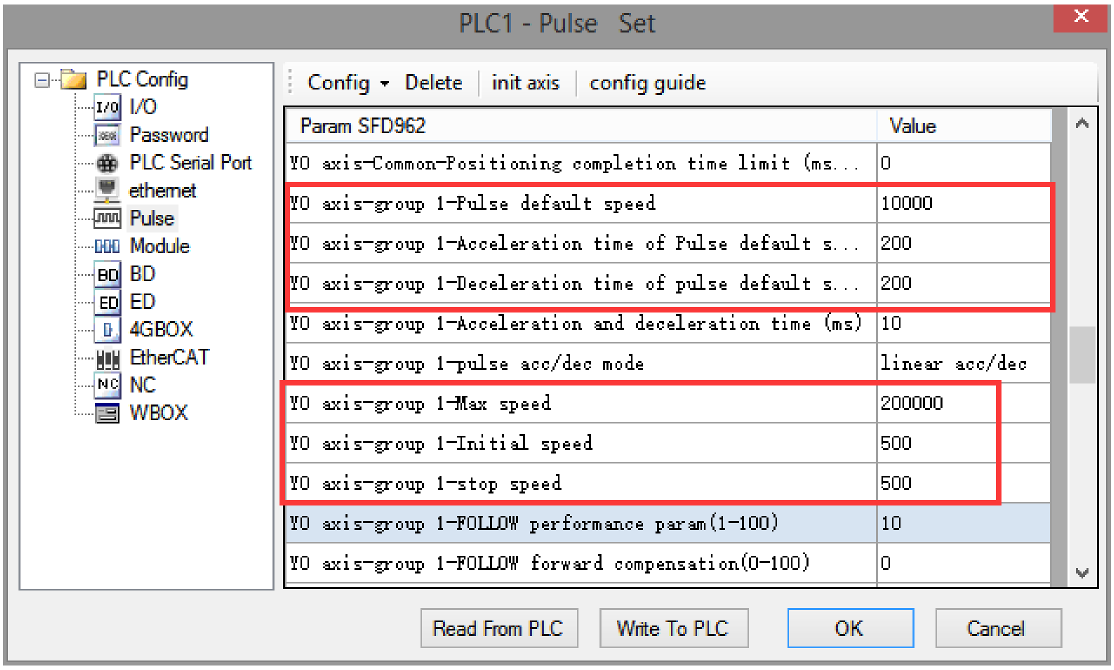


In the parameter configuration table, configure as follows (circled parameters need to be modified):



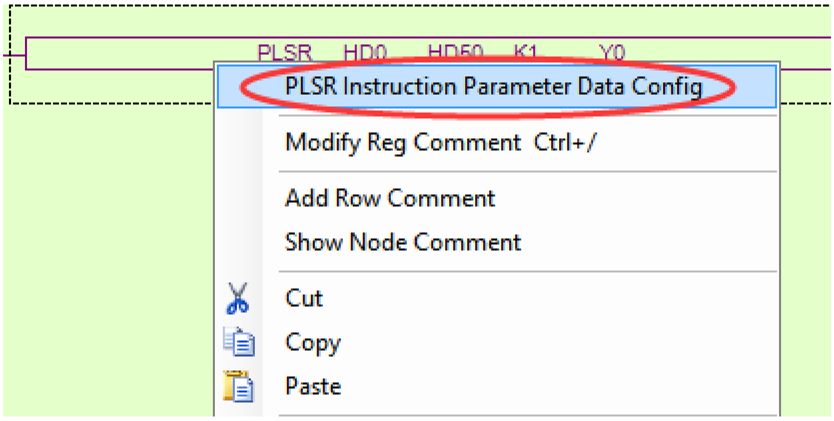

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD922(dword)
Value
Y0 axis-Common-negative limit terminal setting
X12
Y0 axis-Common-Zero clear CLR output setting
Y3
Y0 axis-Common-Return speed VH
10000
Y0 axis-Common-Creeping speed VC
500
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
20
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...
1
Read From PLC
Write To PLC
OK
Cancel

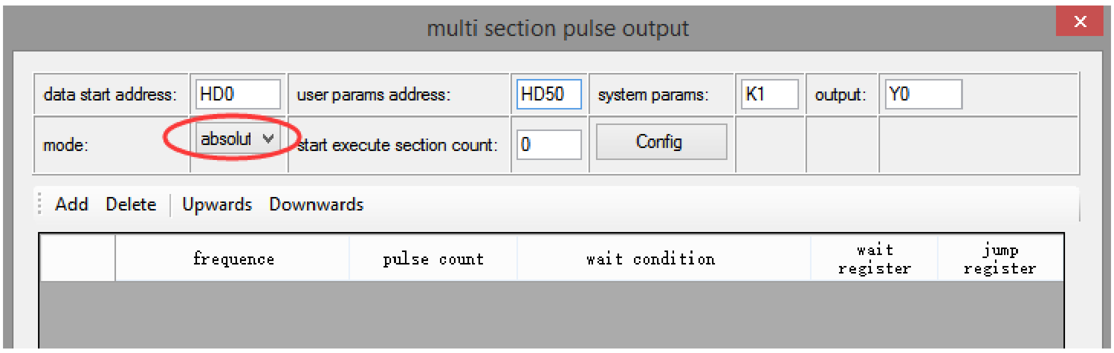


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

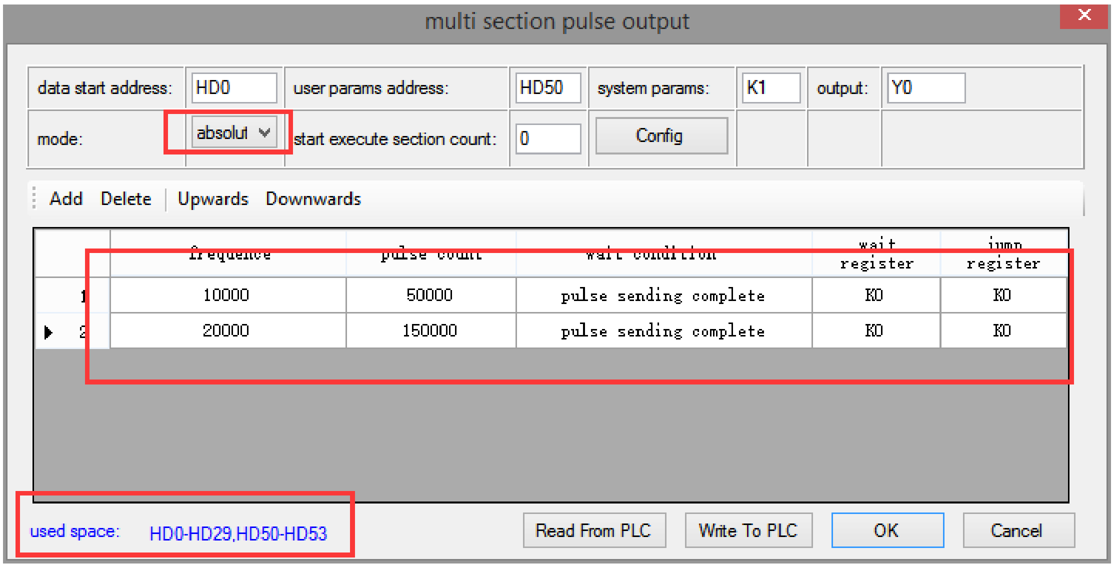
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:

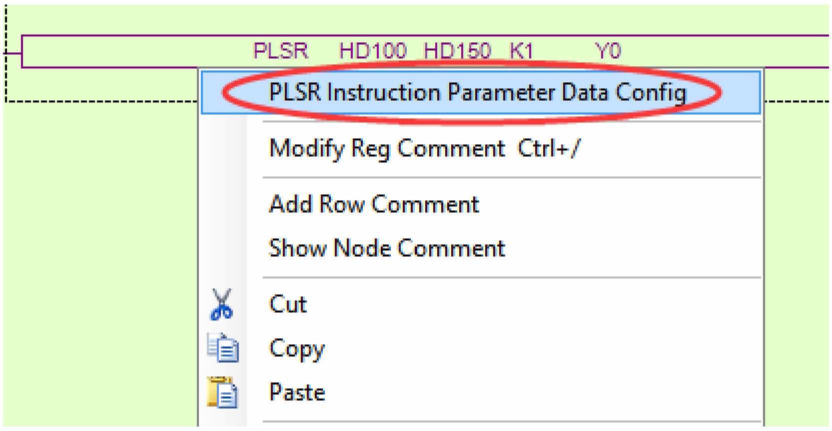


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

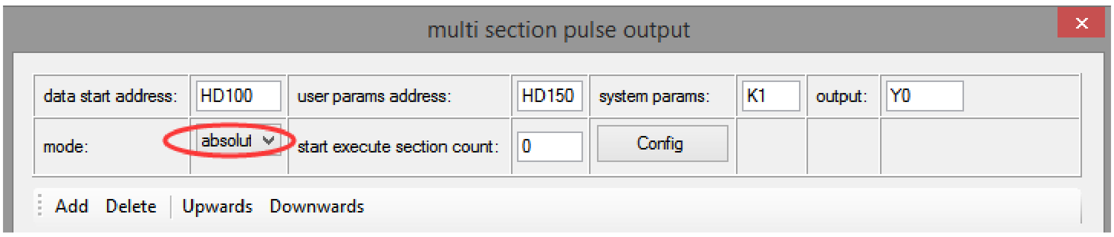


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

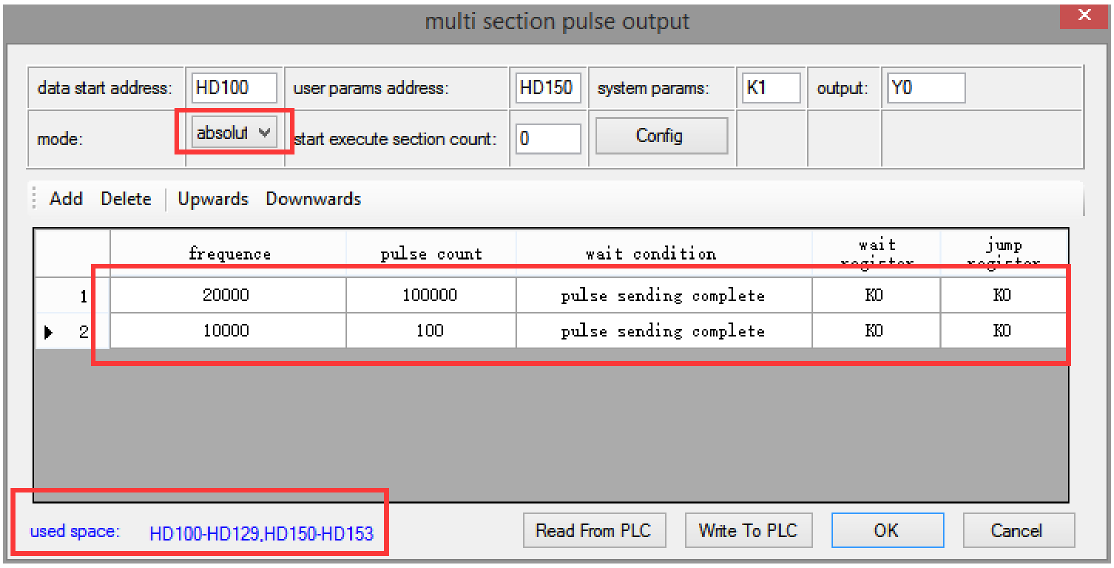
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "absolute" (default is "relative"), as shown in the following figure:



After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

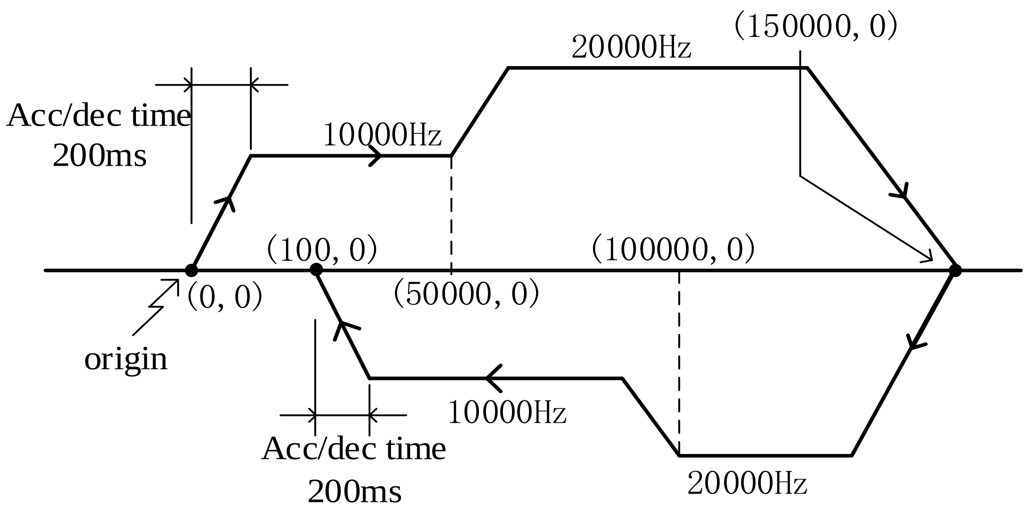


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

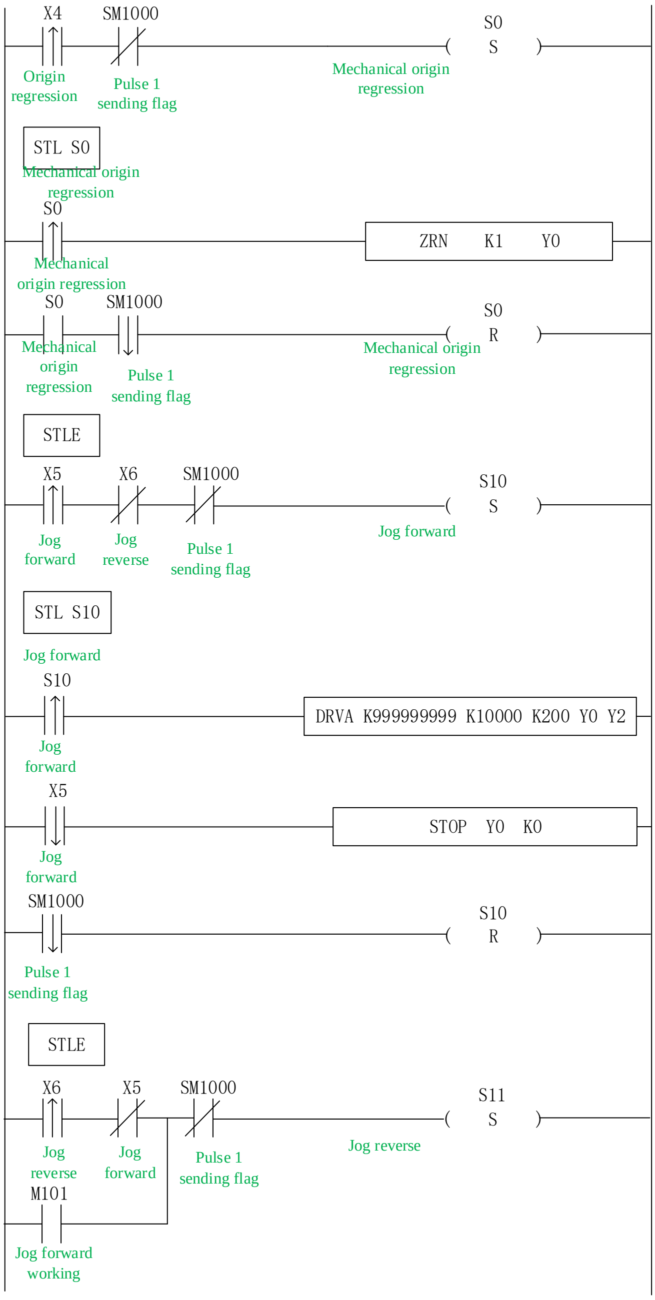
After downloading the program, power off the PLC and then re-energize it.

Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of ZRN, PLSF, DRVI and DRVA instructions.

Example 2: According to the following figure, multi-segment relative positioning method is used.

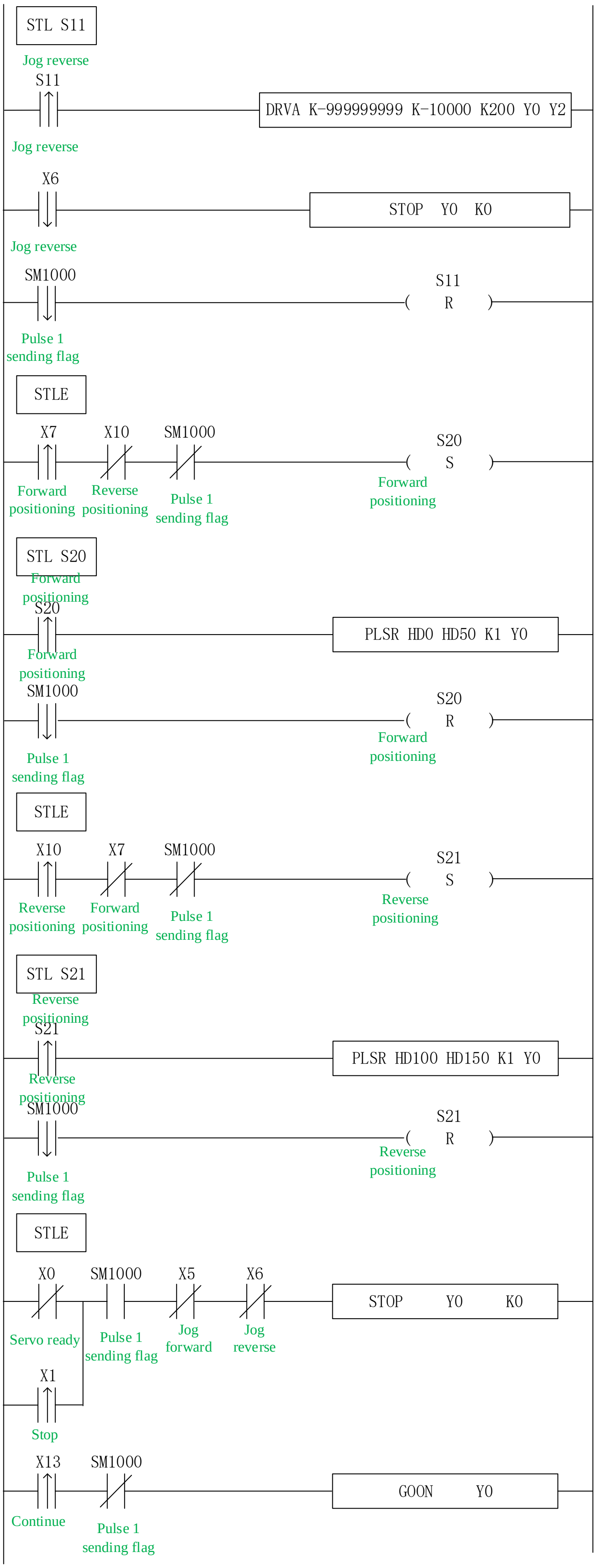


Firstly, make the ladder chart as follows:



//mechanical origin regression (use group 1 parameters)

// Jog forward control, the speed is 10000Hz



//jog reverse control, the speed is -10000Hz

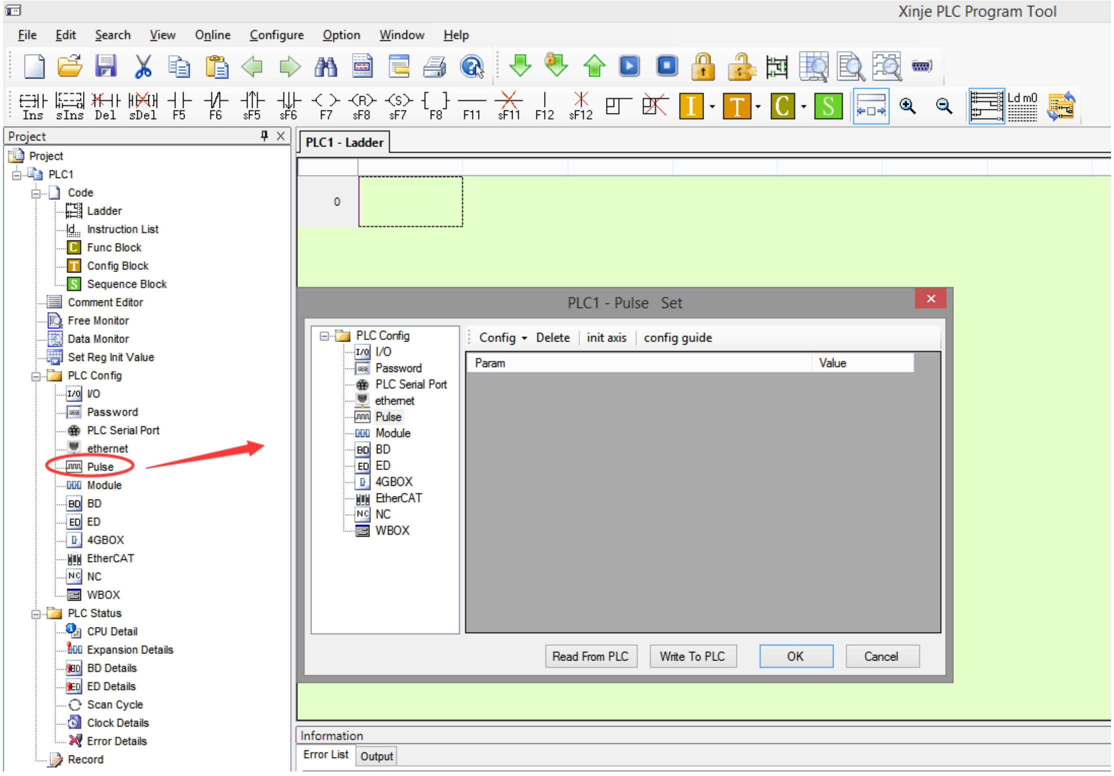
//forward positioning control (use group 1 parameters)

//reverse positioning control (use group 1 parameters)

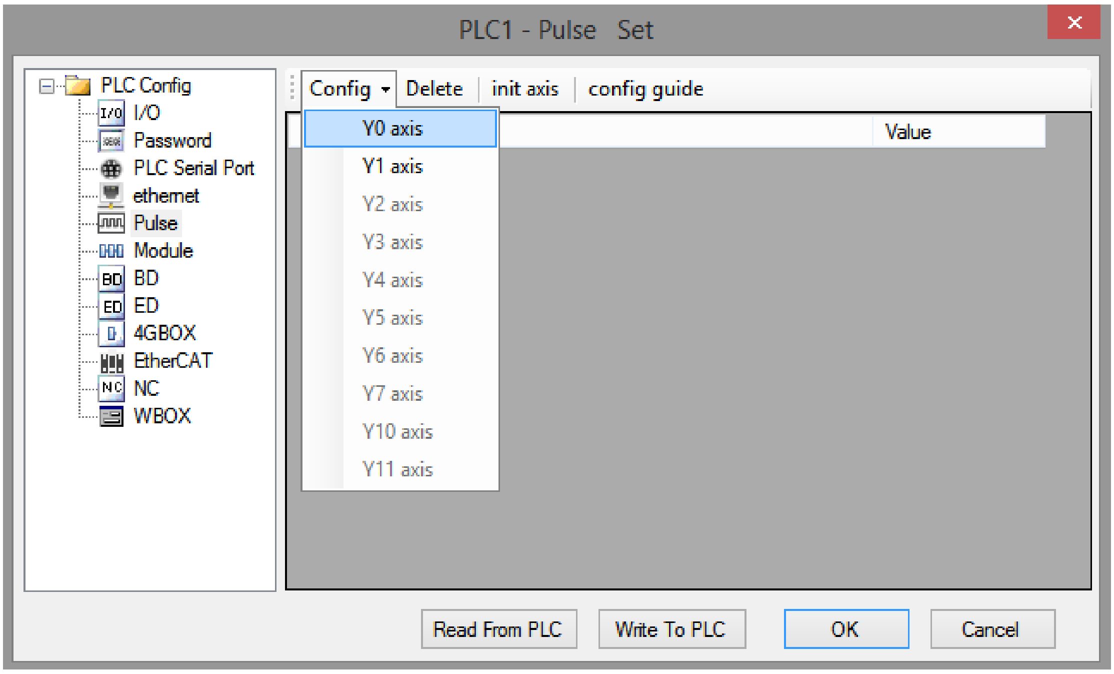
//pulse stop sending (slow stop)

//The execution pulse is not finished and the remaining pulse is continued to be sent out.

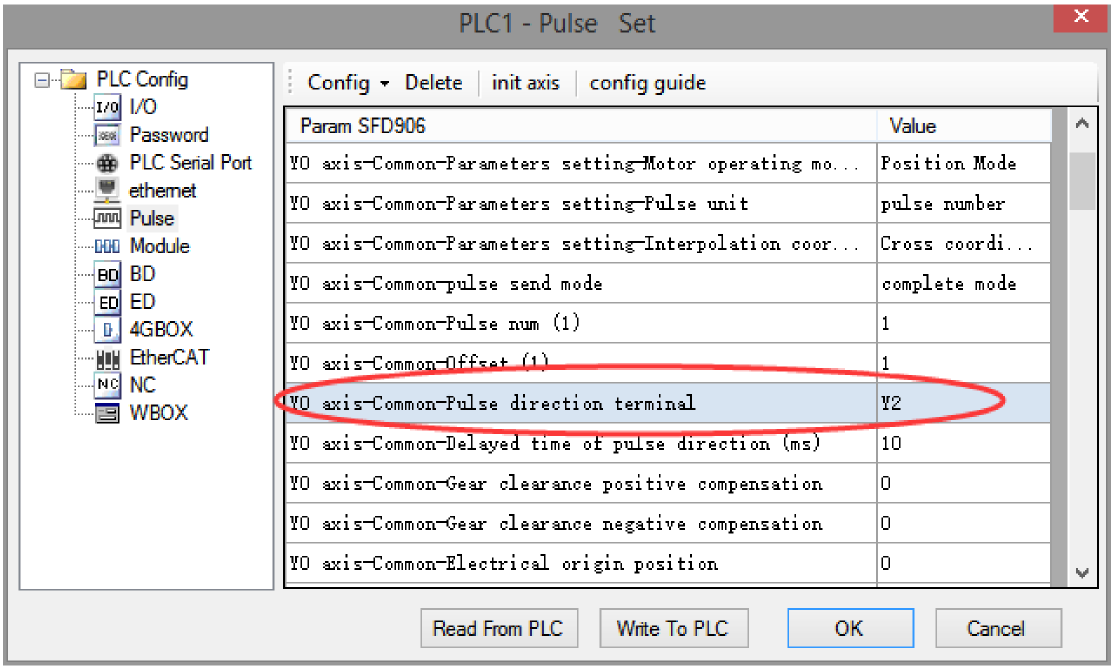
In the sample program, all the system parameters used in the pulse instructions (except DRVA, DRVI) are group 1 parameters. So, we click "pulse configuration parameters" in the PLC programming software, as follows:

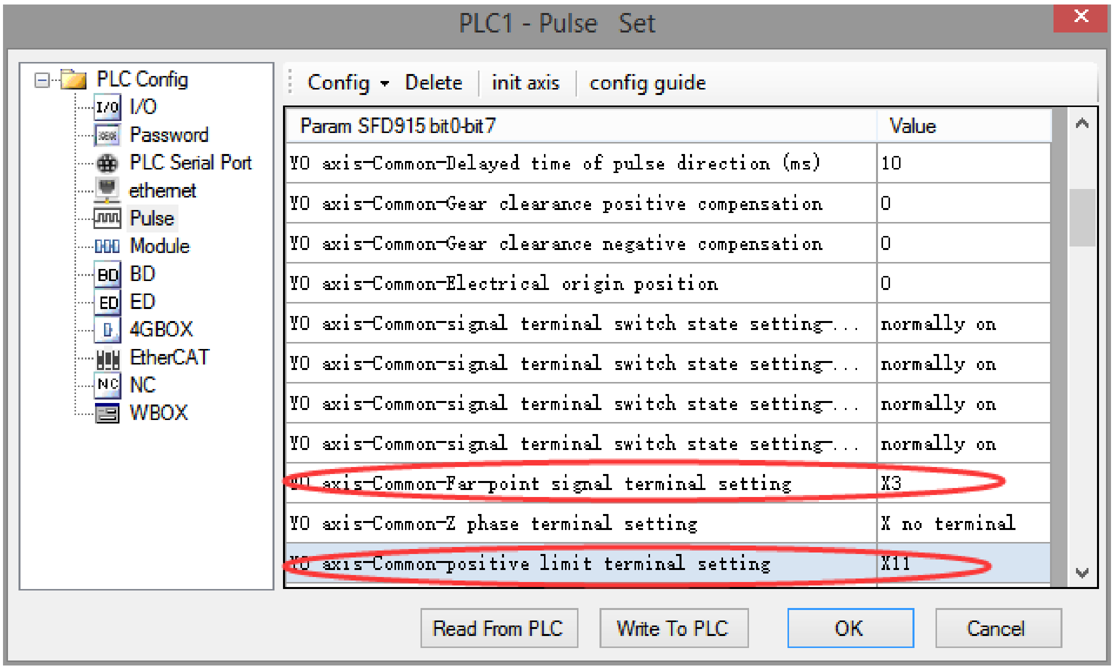


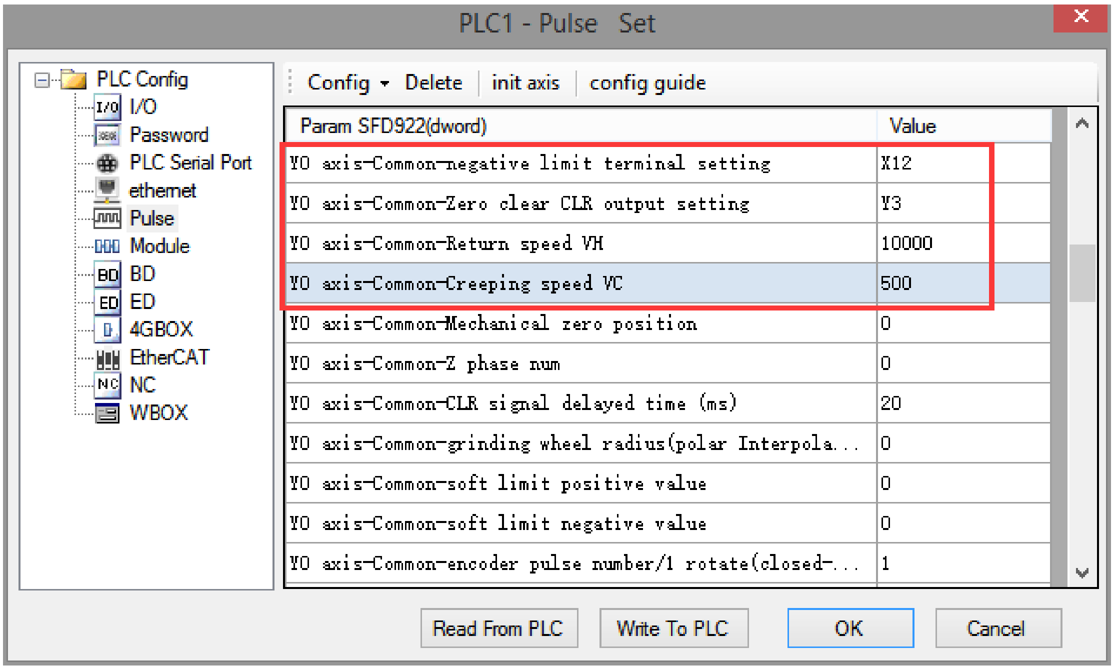
Click config, then select Y0 axis.

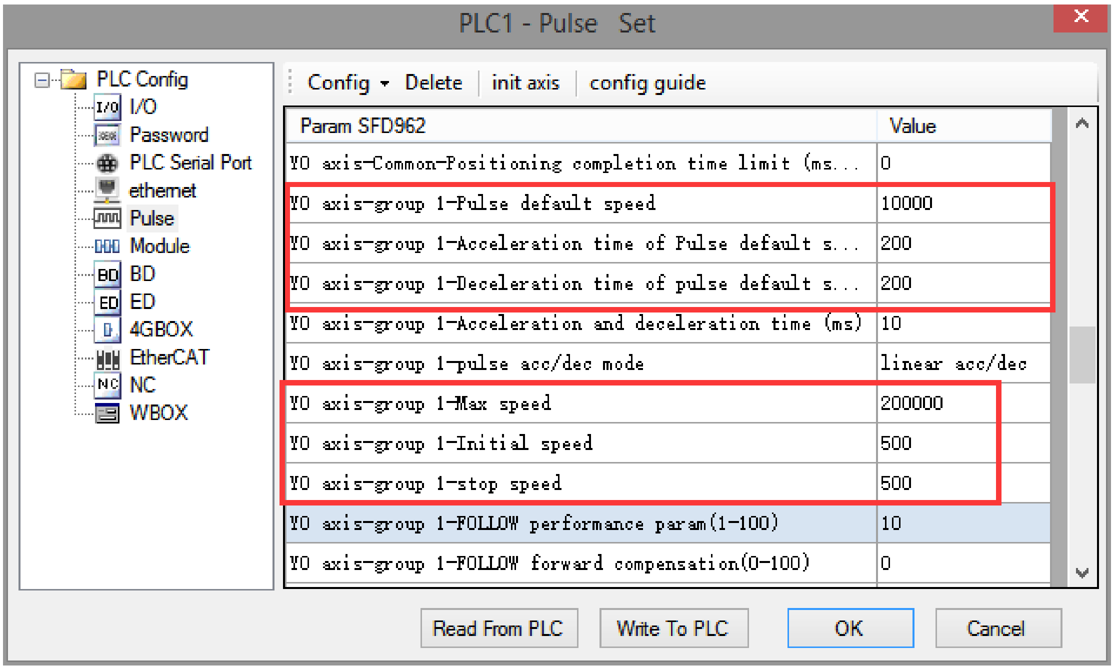


In the parameter configuration table, configure as follows (circled parameters need to be modified):


PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD906
Value
Y0 axis-Common-Parameters setting-Motor operating mo...
Position Mode
Y0 axis-Common-Parameters setting-Pulse unit
pulse number
Y0 axis-Common-Parameters setting-Interpolation coor...
Cross coordi...
Y0 axis-Common-pulse send mode
complete mode
Y0 axis-Common-Pulse num (1)
1
Y0 axis-Common-Offset (1)
1
Y0 axis-Common-Pulse direction terminal
Y2
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Read From PLC
Write To PLC
OK
Cancel

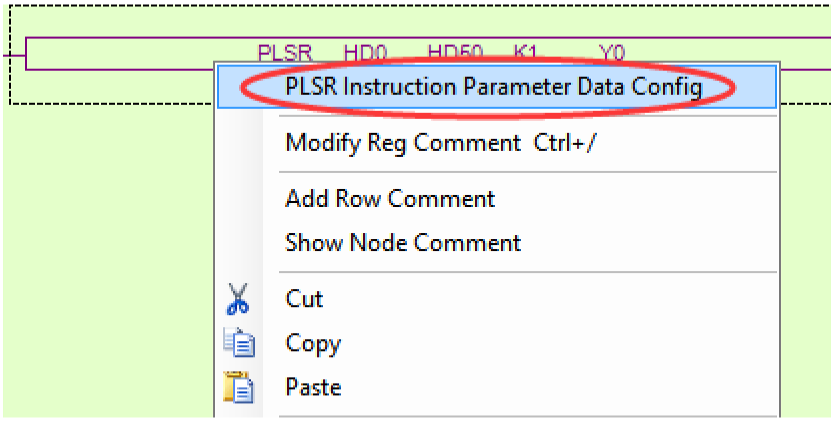

PLC1 - Pulse Set
PLC Config
I/O
Password
PLC Serial Port
ethernet
Pulse
Module
BD
ED
4GBOX
EtherCAT
NC
WBOX
Config
Delete
init axis
config guide
Param SFD915 bit0-bit7
Value
Y0 axis-Common-Delayed time of pulse direction (ms)
10
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X3
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X11
Read From PLC
Write To PLC
OK
Cancel



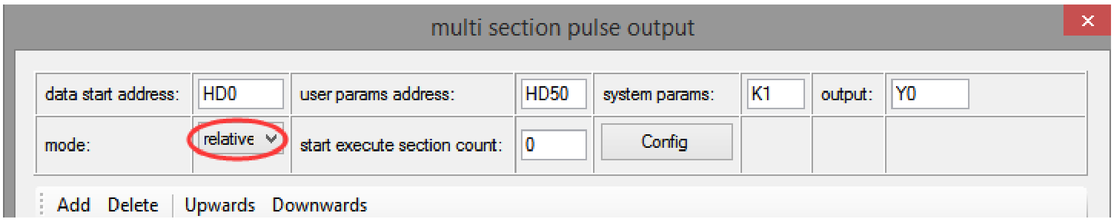


After configuring the parameters of the system parameter block, click the "Write to PLC" button to write the parameters into the PLC. Since the PLSR is used as the multi-segment pulse output instruction, we also need to configure the parameters of the pulse segment (the output frequency and the number of pulses per pulse segment).

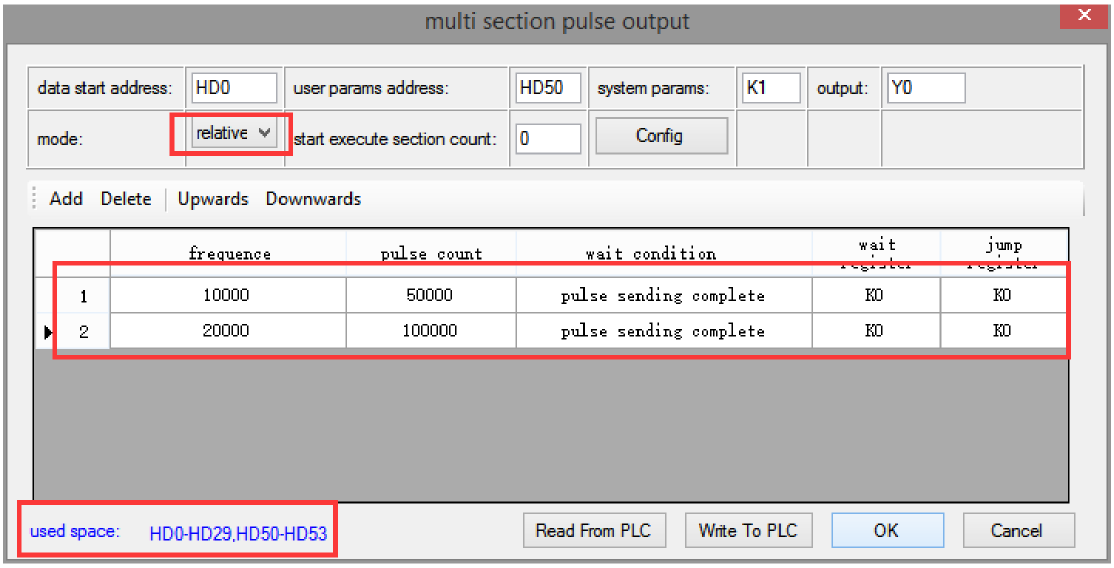
Firstly, right-click on the forward positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:

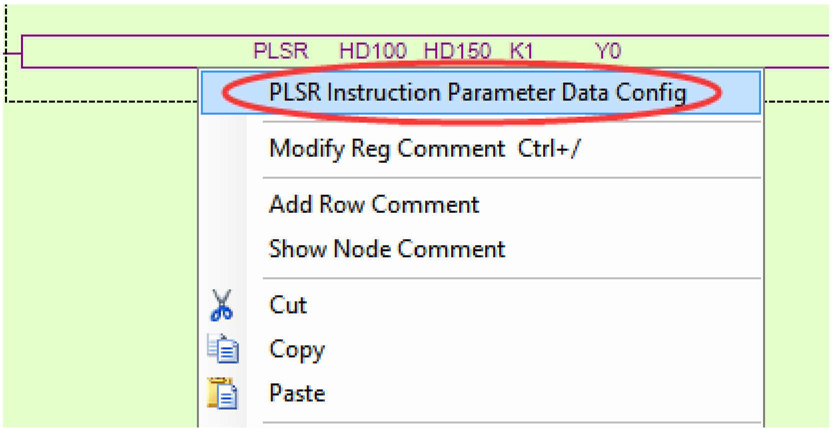


After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of forward rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:

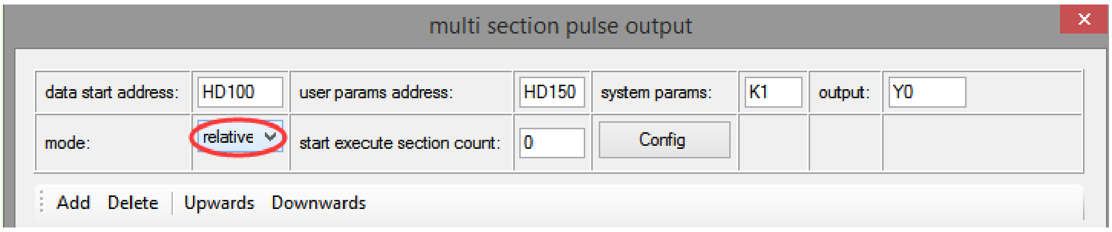


Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD0-HD29, and the range of system parameter block address is HD50-HD53, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

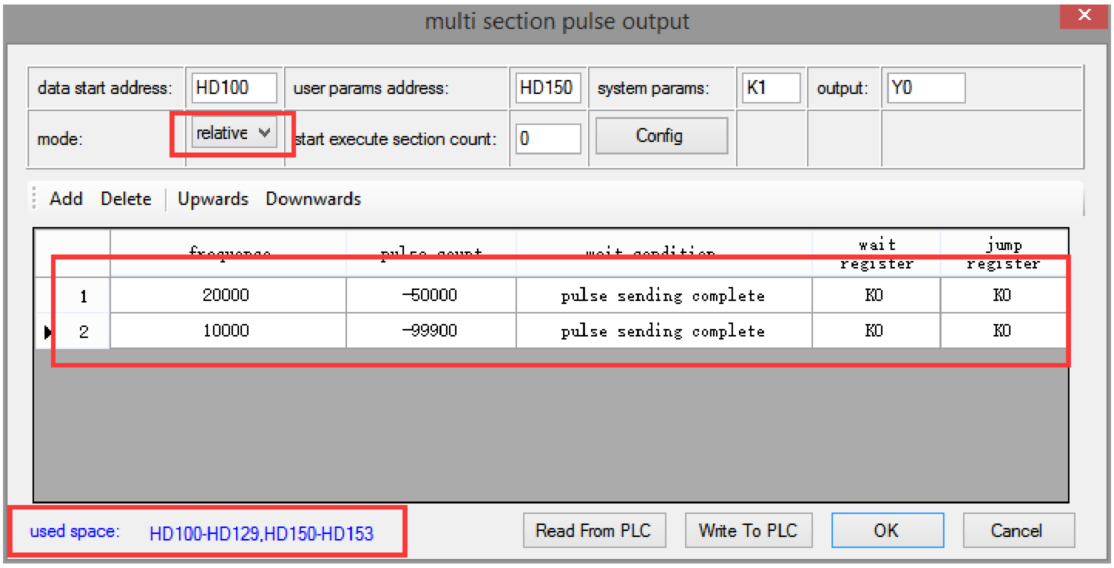
Then, right-click on the reverse positioning command PLSR and pop up the following options. Select the first "PLSR Instruction Parameter data Config":



In the open multi-segment pulse output configuration table, select "mode" as "relative" (default is "relative"), as shown in the following figure:



After choosing the mode, click the "Add" button in the configuration interface to add two continuous pulse parameters of reverse rotation; after configuring, click the "Write to PLC" button to write parameters into the PLC, as shown in the following figure:



Note: Please note the range of real-time occupied registers displayed by "used space". Because the range of starting address of pulse parameter data of PLSR pulse instruction is HD100-HD129, and the range of system parameter block address is HD150-HD153, the range of real-time occupied registers address is not beyond the range. If the range exceeds, the error of pulse output will occur.

After downloading the program, power off the PLC and then re-energize it.

Positive Limit (X11) and Negative Limit (X12) will play an important role in the execution of

ZRN, PLSF, DRVI and DRVA instructions.

1-6. Pulse Output Coil and Register

Pulse output flag bit:

Coil	Function	Notes	
SM1000	Pulse sending flag	1 is pulse sending	PULSE_1
SM1001	Direction flag	1 is positive direction, related direction output is ON	
SM1002	Overflow flag of accumulated pulse number	1 is overflow	
SM1003	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1010	Pulse error flag	ON is error	
SM1020	Pulse sending flag	1 is pulse sending	PULSE_2
SM1021	Direction flag	1 is positive direction, related direction output is ON	
SM1022	Overflow flag of accumulated pulse number	1 is overflow	
SM1023	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1030	Pulse error flag	ON is error	
SM1040	Pulse sending flag	1 is pulse sending	PULSE_3
SM1041	Direction flag	1 is positive direction, related direction output is ON	
SM1042	Overflow flag of accumulated pulse number	1 is overflow	
SM1043	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1050	Pulse error flag	ON is error	
SM1060	Pulse sending flag	1 is pulse sending	PULSE_4
SM1061	Direction flag	1 is positive direction, related direction output is ON	
SM1062	Overflow flag of accumulated pulse number	1 is overflow	
SM1063	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1070	Pulse error flag	ON is error	

SM1080	Pulse sending flag	1 is pulse sending	PULSE_5
SM1081	Direction flag	1 is positive direction, related direction output is ON	
SM1082	Overflow flag of accumulated pulse number	1 is overflow	
SM1083	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1090	Pulse error flag	ON is error	
SM1100	Pulse sending flag	1 is pulse sending	PULSE_6
SM1101	Direction flag	1 is positive direction, related direction output is ON	
SM1102	Overflow flag of accumulated pulse number	1 is overflow	
SM1103	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1110	Pulse error flag	ON is error	
SM1120	Pulse sending flag	1 is pulse sending	PULSE_7
SM1121	Direction flag	1 is positive direction, related direction output is ON	
SM1122	Overflow flag of accumulated pulse number	1 is overflow	
SM1123	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1130	Pulse error flag	ON is error	
SM1140	Pulse sending flag	1 is pulse sending	PULSE_8
SM1141	Direction flag	1 is positive direction, related direction output is ON	
SM1142	Overflow flag of accumulated pulse number	1 is overflow	
SM1143	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1150	Pulse error flag	ON is error	
SM1160	Pulse sending flag	1 is pulse sending	PULSE_9
SM1161	Direction flag	1 is positive direction, related direction output is ON	
SM1162	Overflow flag of accumulated pulse number	1 is overflow	
SM1163	Overflow flag of accumulated pulse	1 is overflow	

	equivalent		
SM1170	Pulse error flag	ON is error	
SM1180	Pulse sending flag	1 is pulse sending	PULSE_10
SM1181	Direction flag	1 is positive direction, related direction output is ON	
SM1182	Overflow flag of accumulated pulse number	1 is overflow	
SM1183	Overflow flag of accumulated pulse equivalent	1 is overflow	
SM1190	Pulse error flag	ON is error	

Pulse output related sepcial registers:

Register	Function	Notes	
SD1000	Present segment (represents segment n)		PULSE_1
SD1001			
SD1002	Present pulse number low 16-bit (the unit is pulse number)		
SD1003	Present pulse number high 16-bit (the unit is pulse number)		
SD1004	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1005	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1006	Present pulse number low 16-bit (the unit is pulse number)		
SD1007	Present pulse number high 16-bit (the unit is pulse number)		
SD1008	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1009	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1010	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error	

		4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC≥VH 13: Origin regression signal error 15:Follow Performance Parameters ≤ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ≤0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1011	Error pulse data block number		
SD1020	Present segment (represents segment n)		
SD1021			
SD1022	Present pulse number low 16-bit (the unit is pulse number)		
SD1023	Present pulse number high 16-bit (the unit is pulse number)		PULSE_2
SD1024	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1025	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1026	Present pulse number low 16-bit (the unit is pulse number)		

SD1027	Present pulse number high 16-bit (the unit is pulse number)		
SD1028	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1029	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1030	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH 13: Origin regression signal error 15:Follow Performance Parameters ⩽ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ⩽0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1031	Error pulse data block number		
SD1040	Present segment (represents segment n)		PULSE_3
SD1041			
SD1042	Present pulse number		

	low 16-bit (the unit is pulse number)		
SD1043	Present pulse number high 16-bit (the unit is pulse number)		
SD1044	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1045	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1046	Present pulse number low 16-bit (the unit is pulse number)		
SD1047	Present pulse number high 16-bit (the unit is pulse number)		
SD1048	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1049	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1050	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC≥VH 13: Origin regression signal error 15:Follow Performance Parameters ≤ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ≤0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed	

		is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1051	Error pulse data block number		
SD1060	Present segment (represents segment n)		PULSE_4
SD1061			
SD1062	Present pulse number low 16-bit (the unit is pulse number)		
SD1063	Present pulse number high 16-bit (the unit is pulse number)		
SD1064	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1065	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1066	Present pulse number low 16-bit (the unit is pulse number)		
SD1067	Present pulse number high 16-bit (the unit is pulse number)		
SD1068	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1069	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1070	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal	

		10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC≥VH 13: Origin regression signal error 15:Follow Performance Parameters ≤ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ≤0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1071	Error pulse data block number		
SD1080	Present segment (represents segment n)		
SD1081			
SD1082	Present pulse number low 16-bit (the unit is pulse number)		
SD1083	Present pulse number high 16-bit (the unit is pulse number)		
SD1084	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1085	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1086	Present pulse number low 16-bit (the unit is pulse number)		
SD1087	Present pulse number high 16-bit (the unit is pulse number)		
SD1088	Present pulse number		PULSE_5

	low 16-bit (the unit is pulse equivalent)		
SD1089	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1090	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH 13: Origin regression signal error 15:Follow Performance Parameters ⩽ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ⩽0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1091	Error pulse data block number		
SD1100	Present segment (represents segment n)		PULSE_6
SD1101			
SD1102	Present pulse number low 16-bit (the unit is pulse number)		
SD1103	Present pulse number high 16-bit (the unit is		

	pulse number)		
SD1104	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1105	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1106	Present pulse number low 16-bit (the unit is pulse number)		
SD1107	Present pulse number high 16-bit (the unit is pulse number)		
SD1108	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1109	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1110	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH 13: Origin regression signal error 15:Follow Performance Parameters ⩽ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ⩽0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error	

		24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1111	Error pulse data block number		
SD1120	Present segment (represents segment n)		PULSE_7
SD1121			
SD1122	Present pulse number low 16-bit (the unit is pulse number)		
SD1123	Present pulse number high 16-bit (the unit is pulse number)		
SD1124	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1125	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1126	Present pulse number low 16-bit (the unit is pulse number)		
SD1127	Present pulse number high 16-bit (the unit is pulse number)		
SD1128	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1129	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1130	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH	

		13: Origin regression signal error 15:Follow Performance Parameters ≤ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ≤0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1131	Error pulse data block number		
SD1140	Present segment (represents segment n)		
SD1141			
SD1142	Present pulse number low 16-bit (the unit is pulse number)		
SD1143	Present pulse number high 16-bit (the unit is pulse number)		
SD1144	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1145	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1146	Present pulse number low 16-bit (the unit is pulse number)		
SD1147	Present pulse number high 16-bit (the unit is pulse number)		
SD1148	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1149	Present pulse number high 16-bit (the unit is pulse equivalent)		PULSE_8

SD1150	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH 13: Origin regression signal error 15:Follow Performance Parameters ⩽ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ⩽0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1151	Error pulse data block number		
SD1160	Present segment (represents segment n)		PULSE_9
SD1161			
SD1162	Present pulse number low 16-bit (the unit is pulse number)		
SD1163	Present pulse number high 16-bit (the unit is pulse number)		
SD1164	Present pulse number low 16-bit (the unit is pulse equivalent)		

SD1165	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1166	Present pulse number low 16-bit (the unit is pulse number)		
SD1167	Present pulse number high 16-bit (the unit is pulse number)		
SD1168	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1169	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1170	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH 13: Origin regression signal error 15:Follow Performance Parameters ⩽ 0 or >100 16:Follow Feedforward Compensation <0 or>100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ⩽0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1171	Error pulse data block		

	number		
SD1180	Present segment (represents segment n)		PULSE-_10
SD1181			
SD1182	Present pulse number low 16-bit (the unit is pulse number)		
SD1183	Present pulse number high 16-bit (the unit is pulse number)		
SD1184	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1185	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1186	Present pulse number low 16-bit (the unit is pulse number)		
SD1187	Present pulse number high 16-bit (the unit is pulse number)		
SD1188	Present pulse number low 16-bit (the unit is pulse equivalent)		
SD1189	Present pulse number high 16-bit (the unit is pulse equivalent)		
SD1190	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per rotation and the movement per rotation is 0 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC⩾VH 13: Origin regression signal error 15:Follow Performance Parameters ⩽ 0 or >100 16:Follow Feedforward Compensation <0 or>100	

		17:Follow Multiplication Coefficient and Division Coefficient Ratio ≤0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1191	Error pulse data block number		

High speed pulse special data register HSD (power off memory)

Register	Function	Note	
HSD0	Low 16 bits of cumulative pulse (the unit is pulse number)		
HSD1	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD2	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD3	High 16 bits of cumulative pulse (the unit is pulse equivalent)		PULSE_1
HSD4	Low 16 bits of cumulative pulse (the unit is pulse number)		
HSD5	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD6	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD7	High 16 bits of cumulative pulse (the unit is pulse equivalent)		PULSE_2
HSD8	Low 16 bits of cumulative pulse (the unit is pulse number)		
HSD9	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD10	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD11	High 16 bits of cumulative pulse (the unit is pulse equivalent)		PULSE_3

HSD12	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_4
HSD13	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD14	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD15	High 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD16	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_5
HSD17	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD18	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD19	High 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD20	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_6
HSD21	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD22	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD23	High 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD24	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_7
HSD25	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD26	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD27	High 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD28	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_8
HSD29	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD30	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD31	High 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD32	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_9

HSD33	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD34	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD35	High 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD36	Low 16 bits of cumulative pulse (the unit is pulse number)		PULSE_10
HSD37	High 16 bits of cumulative pulse (the unit is pulse number)		
HSD38	Low 16 bits of cumulative pulse (the unit is pulse equivalent)		
HSD39	High 16 bits of cumulative pulse (the unit is pulse equivalent)		

2 Motion control

2-1．Motion control instruction list

The following motion control instructions are suitable for XDM, XDME, XLME series PLC.

Instruction	Function	Chapter
DRV	Quick positioning	2-4-1
DRVR	Quick positioning, polar coordinate mode (temporarily unavailable)	2-4-2
LIN line	Linear interpolation	2-4-3
LIN line VM	Linear interpolation, maximum speed can be specified separately	2-4-3
LIN line VBEM	Linear interpolation, can specify the starting speed, terminal speed and maximum speed separately	2-4-3
CW clockwise	Clockwise circular interpolation	2-4-4
CW closewise VM	Clockwise circular interpolation, maximum speed can be specified separately	2-4-4
CW closewise VBEM	Clockwise circular interpolation，can specify the starting speed, terminal speed and maximum speed separately	2-4-4
CCW anticlockwise	Anticlockwise circular interpolation	2-4-5
CCW anticlockwise VM	Anticlockwise circular interpolation, maximum speed can be specified separately	2-4-5
CCW anticlockwise VBEM	Anticlockwise circular interpolation, can specify the starting speed, terminal speed and maximum speed separately	2-4-5
CW_R closewise	Clockwise circular interpolation (Specified radius)	2-4-6
CW_R closewise VM	Clockwise circular interpolation(Specified radius), maximum speed can be specified separately	2-4-6
CW_R closewise VBEM	Clockwise circular interpolation(Specified radius)，can specify the starting speed, terminal speed and maximum speed separately	2-4-6
CCW_R anticlockwise	Anticlockwise circular interpolation(Specified radius)	2-4-7
CCW_R anticlockwise VM	Anticlockwise circular interpolation(Specified radius), maximum speed can be specified separately	2-4-7
CCW_R anticlockwise VBEM	Anticlockwise circular interpolation(Specified radius)，can specify the starting speed, terminal speed and maximum speed separately	2-4-7

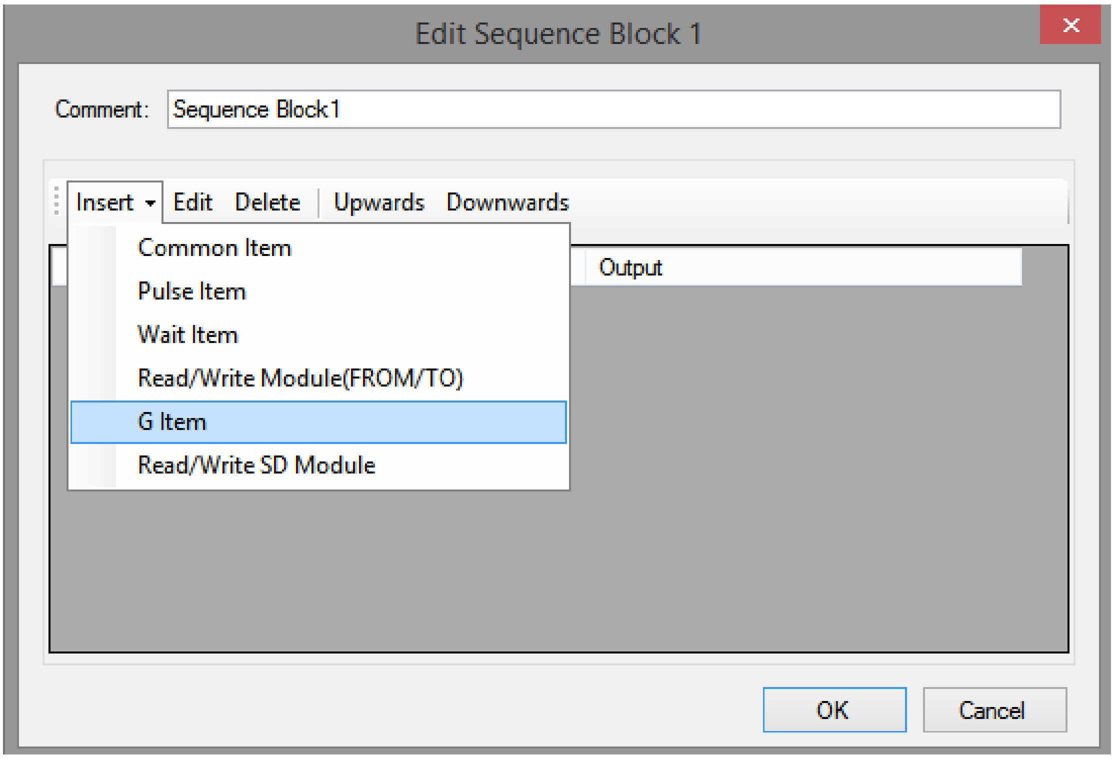
ARC three points	Three points arc	2-4-8
ARC three point VM	Three points arc, maximum speed can be specified separately	2-4-8
ARC three point VBEM	Three points arc, can specify the starting speed, terminal speed and maximum speed separately	2-4-8
FOLLOW	Single phase follow	2-4-9
FOLLOW_AB	AB phase follow	2-4-9

Note: All interpolation instructions have no stop when jumping, there is inflection point.

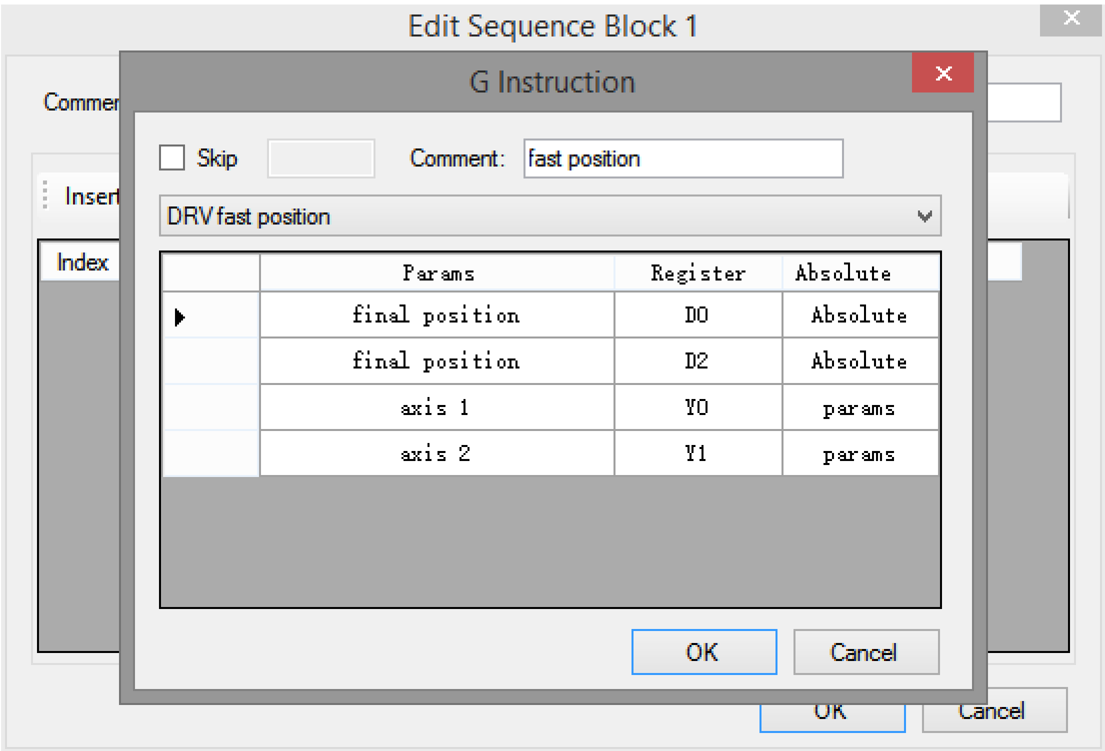
2-2. Writing method of motion control instruction

Except FOLLOW, other motion control instructions must be written in the BLOCK. The specific methods are as follows:

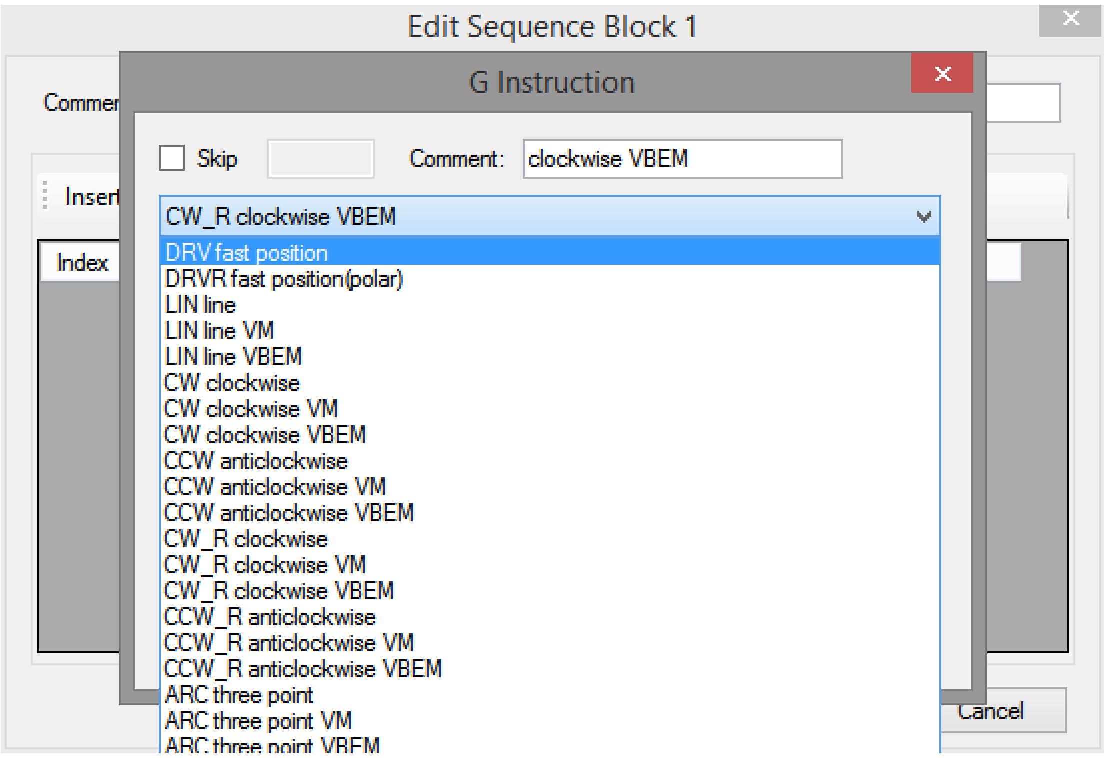
1. insert a sequence block S in the ladder chart, then insert G instruction.



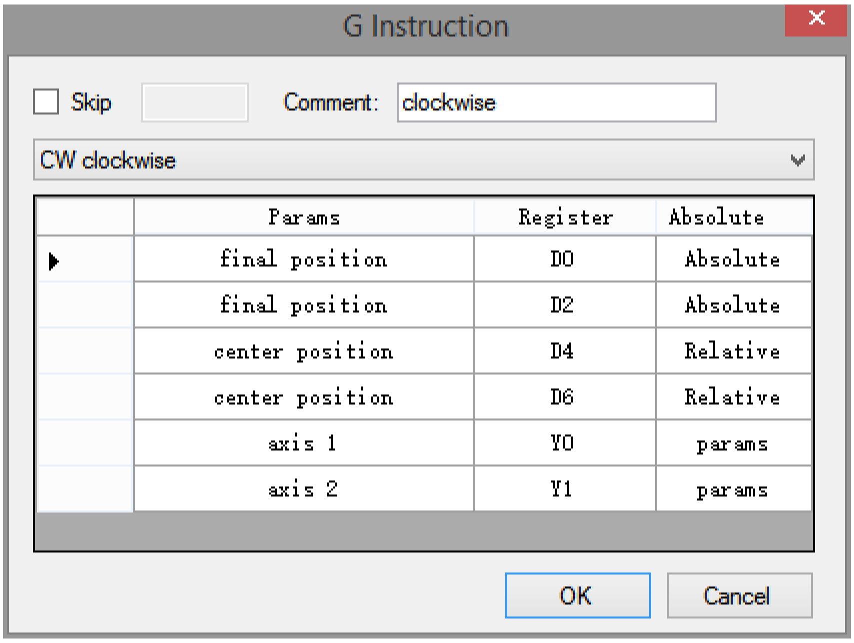
2. it will show the following window



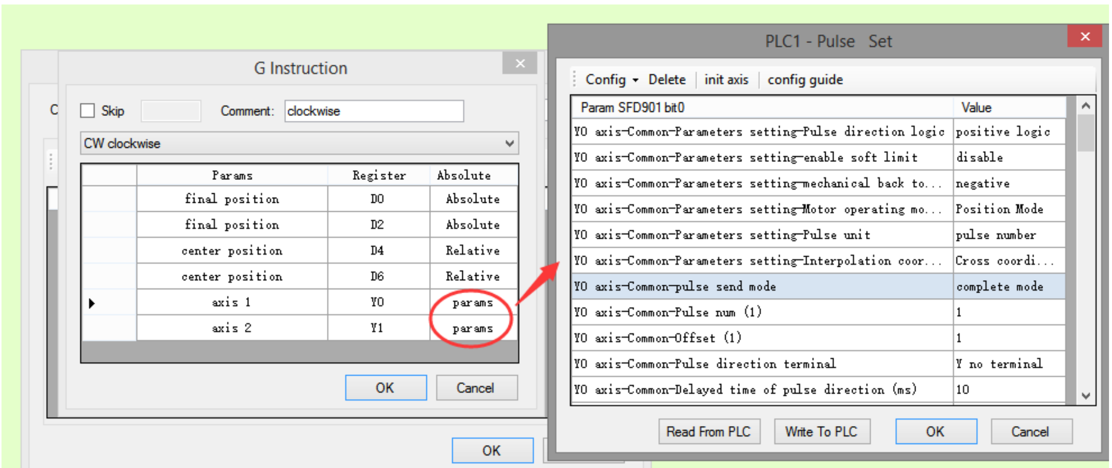
3. click the dropdown menu, select the motion control instruction to



4. click the motion control instruction CW clockwise, it will show the instruction configuration window:



In the register list, double click the value can change the register address and axis output terminal. In the absolute list, double click the value can set the mode (relative/absolute).
Double click the parameters can set the direction, speed, acc/dec time of the two axes, please see the follows:

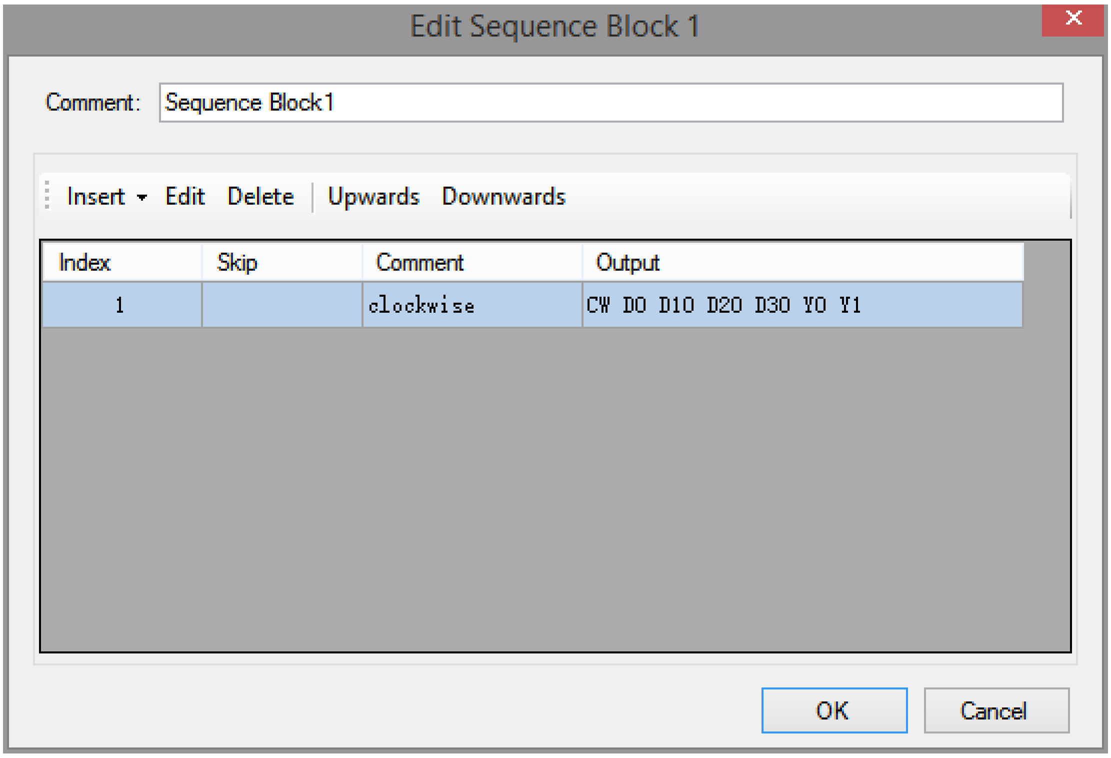


Note:

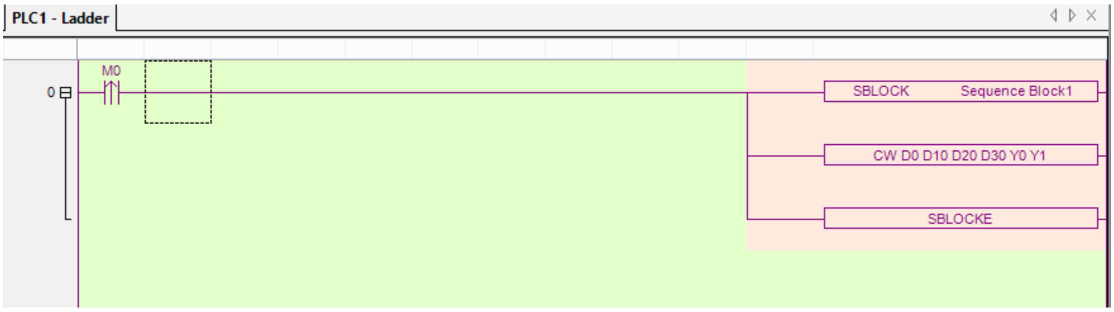
(1) Different instructions require different system parameter blocks. See chapter 2-3-2 and instructions for details.

(2) See chapter 1-2-1 for system parameters.

5. Configuration is completed, click OK, and you can see the general situation of the generated instructions in the SBLOCK:



6. A complete motion control instruction is completed by generating the motion control instructions in the ladder diagram and inputting the driving conditions.



7. Execute BLOCK once every time M0 rises.

8. Multiple motion control instructions can be inserted into BLOCK. Lines and arcs can be used to fulfill different interpolation requirements.

2-3. Pulse output terminal distribution and parameters

This section will introduce the distribution of the output port of each PLC pulse in XD series and the configuration of the parameters of each axis pulse.

2-3-1. Pulse output port distribution

In all transistor output terminals of XDM series PLC, the operation axes of axle 1 and axle 2 can be arbitrarily specified, and the corresponding direction terminals can also be arbitrarily specified.

XDM-24T4

Output	Y0~Y3	Y4~Y11
Function	Pulse output	Direction output

XDM-32T4, XLME-32T4

Output	Y0~Y3	Y4~Y15
Function	Pulse output	Direction output

XDM-60T4, XDM-60T4L

Output	Y0~Y3	Y4~Y27
Function	Pulse output	Direction output

XDM-60T10, XDME-60T10

Output	Y0~Y11	Y12~Y27
Function	Pulse output	Direction output

Note: Pulse output terminals that are not used can also be used as directional terminals.

2-3-2. Pulse output terminal parameters

In order to execute the motion control command, it is necessary to configure the pulse control parameters of axis 1 and axis 2. However, only part of the pulse parameters are used in the motion control command, and part of these parameters are common parameters of two axes (i.e. the parameters configurated in axis 1 are valid). As shown in the following figure:

Common parameter	Pulse direction logic	Independent parameter	Axis 1 and 2 need to be set
	Enable soft limit	Common parameter	Only need to set axis 1
	Pulse unit	Common parameter	Only need to set axis 1
	Pulse number	Independent	Axis 1 and 2 need to be set

		parameter	
	Offset	Independent parameter	Axis 1 and 2 need to be set
	Pulse direction terminal	Independent parameter	Axis 1 and 2 need to be set
	Signal terminal switch state setting---positive limit	Independent parameter	Axis 1 and 2 need to be set
	Signal terminal switch state setting---negative limit	Independent parameter	Axis 1 and 2 need to be set
	Positive limit terminal setting	Independent parameter	Axis 1 and 2 need to be set
	Negative limit terminal setting	Independent parameter	Axis 1 and 2 need to be set
	Soft limit positive value	Independent parameter	Axis 1 and 2 need to be set
	Soft limit negative value	Independent parameter	Axis 1 and 2 need to be set
Group 2 parameters	Pulse default speed	Common parameter	Only need to set axis 1
	Acceleration time of pulse default speed	Common parameter	Only need to set axis 1
	Deceleration time of pulse default speed	Common parameter	Only need to set axis 1
	Max speed	Common parameter	Only need to set axis 1
	Initial speed	Common parameter	Only need to set axis 1
	Stop speed	Common parameter	Only need to set axis 1

Note: The above table is applicable to all motion control instructions except DRV and DRVR.

DRV and DRVR instructions used parameters:

Common parameters	Pulse direction logic	Independent parameter	Axis 1 and 2 need to be set
	Enable soft limit	Common parameter	Only need to set axis 1
	Pulse unit	Common parameter	Only need to set axis 1
	Pulse number	Independent parameter	Axis 1 and 2 need to be set
	Offset	Independent parameter	Axis 1 and 2 need to be set
	Pulse direction terminal	Independent	Axis 1 and 2 need to be set

		parameter	
	Signal terminal switch state setting---positive limit	Independent parameter	Axis 1 and 2 need to be set
	Signal terminal switch state setting---negative limit	Independent parameter	Axis 1 and 2 need to be set
	Positive limit terminal setting	Independent parameter	Axis 1 and 2 need to be set
	Negative limit terminal setting	Independent parameter	Axis 1 and 2 need to be set
	Soft limit positive value	Independent parameter	Axis 1 and 2 need to be set
	Soft limit negative value	Independent parameter	Axis 1 and 2 need to be set
Group 1 parameters	Pulse default speed	Common parameter	Axis 1 and 2 need to be set
	Acceleration time of pulse default speed	Common parameter	Axis 1 and 2 need to be set
	Deceleration time of pulse default speed	Common parameter	Axis 1 and 2 need to be set
	Max speed	Common parameter	Axis 1 and 2 need to be set
	Initial speed	Common parameter	Axis 1 and 2 need to be set
	Stop speed	Common parameter	Axis 1 and 2 need to be set

Note: For a detailed description of the pulse parameters, please refer to the relevant content of Chapter 1.

2-4. Motion control instruction

2-4-1. Quick positioning [DRV]

1. instruction overview

Quick positioning instructions. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Quick positioning [DRV]			
16-bit instruction	-	32-bit instruction	DRV
Execute condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. operand

Operand	Function	Type
S0	The target position of axis 1	Double words, 32-bit
S1	The target position of axis 2	Double words, 32-bit
D0	Pulse output terminal of axis 1	Bit
D1	Pulse output terminal of axis 2	Bit

3. suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

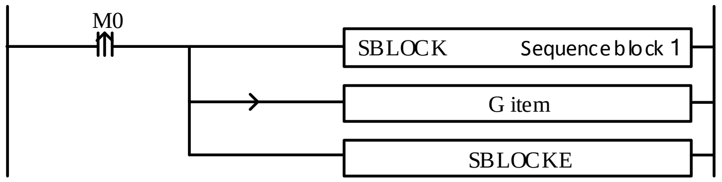
4. Parameter setting

Relative parameters	Settings	Note
Final position	Free to specify register address	Must set
Relative/ absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Axis 1 pulse output port	Free to specify pulse output terminal	Must set
Axis 2 pulse output	Free to specify pulse output terminal	Must set

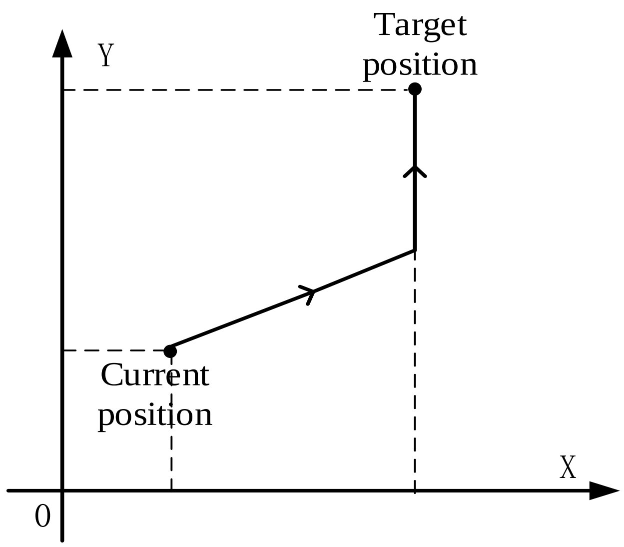
port		
Axis 1 direction port	Arbitrarily specify idle output points, set in system parameters	Must set
Axis 2 direction port	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	Setting in System Parameters of Axis 1	Must set
Pulse default speed	Specify in group 1 parameters of the system parameters of each axis	Must set
Acceleration time	Specify in group 1 parameters of the system parameters of each axis	No need to set
Deceleration time	Specify in group 1 parameters of the system parameters of each axis	No need to set

Function and action

《Instruction format》



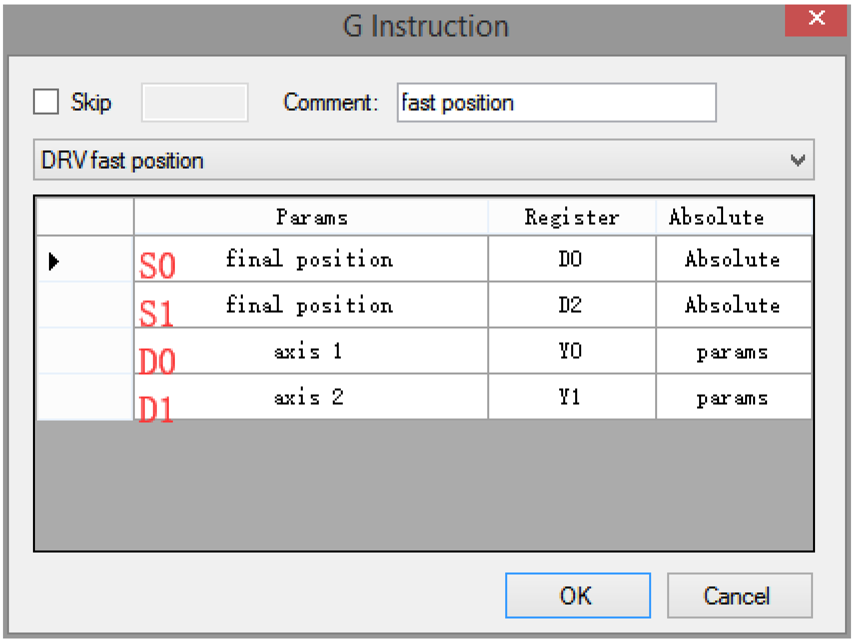
When the quick positioning DRV command is executed, the two axes will move rapidly from the current position to the target position at the default pulse speed set by their respective axes (when one axis is finished first, the other axis will continue to move at the default pulse speed, and then finish positioning after reaching the target position). As shown in the following figure:



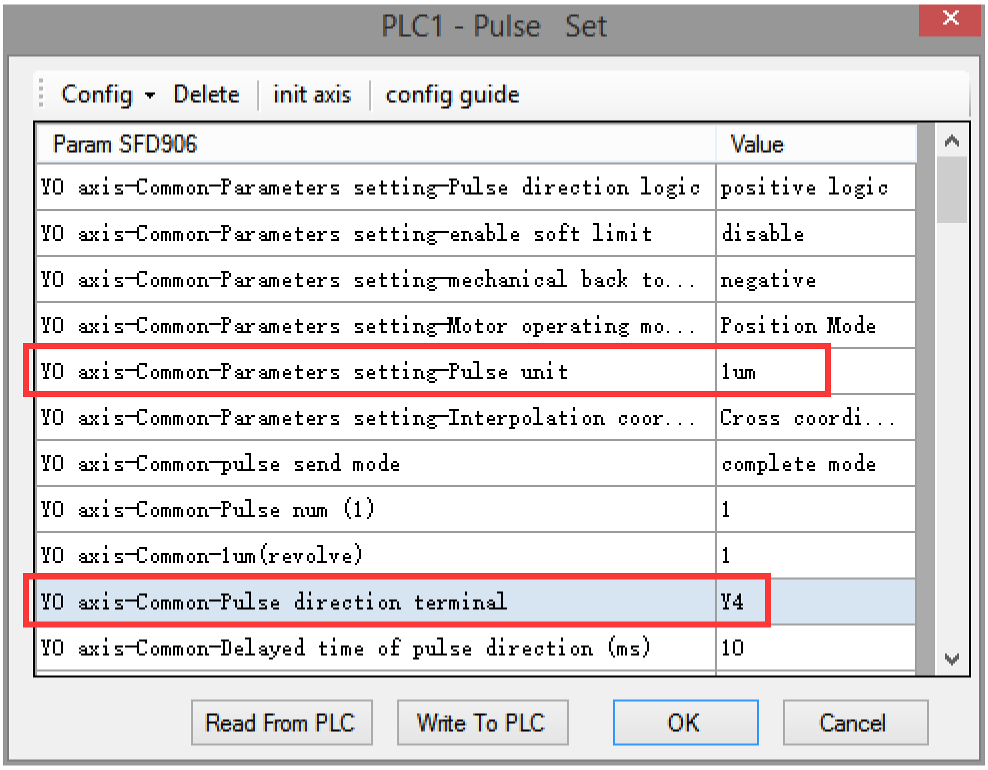
DRV quick positioning

Parameter configuration

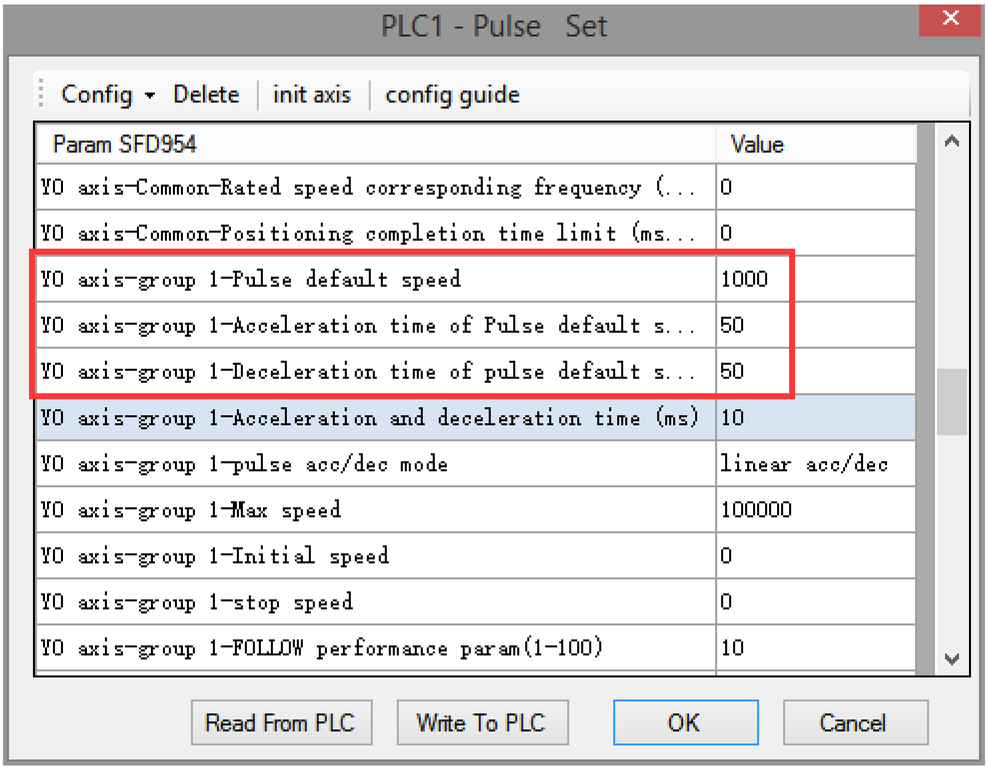
Double click G item, it will pop up the DRV configuration panel:



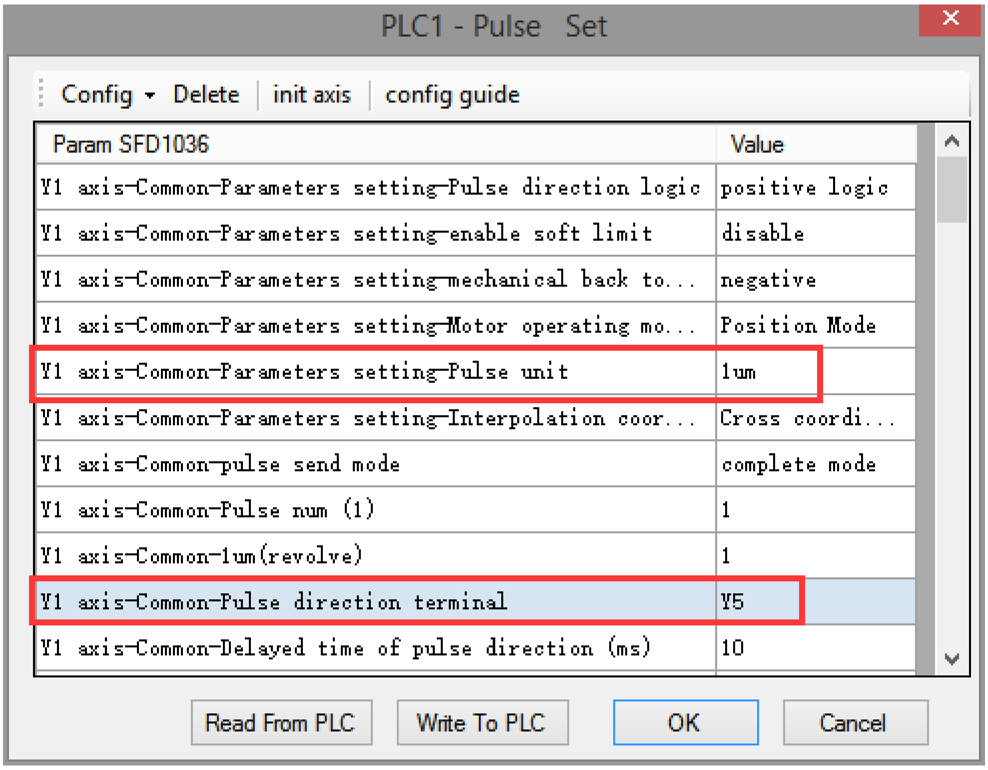
Command configuration



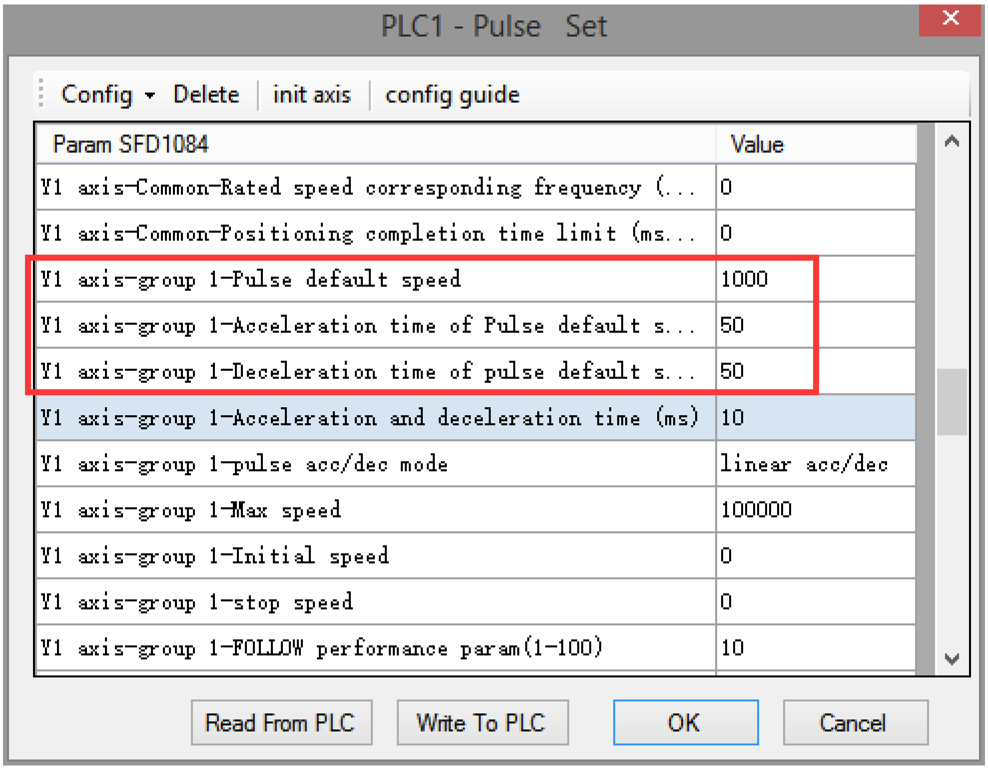
Y0 axis system parameters (1)



Y0 axis system parameters (2)



Y1 axis system parameters (1)



Y1 axis system parameters (2)

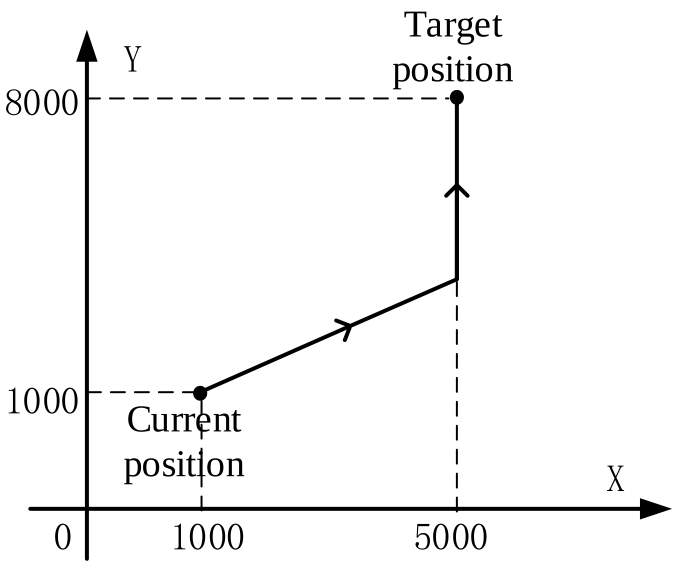
- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is set ON for the forward pulse and set OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Position movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 500, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute DRV instructions and move to the target position with 1000 Hz, 50ms acceleration/deceleration time, if:

  (1) If the final position is absolute mode, the target position is (5000,2000);

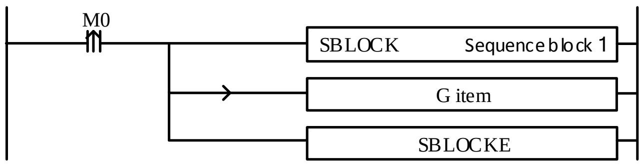
  (2) When the final position is in the relative mode, the target position is (5500,3000).
- When the DRV instruction is running, the pulse flag bit corresponding to the output port Y of the DRV instruction will be set on.

Note: DRV instructions are fixed using group 1 parameters!

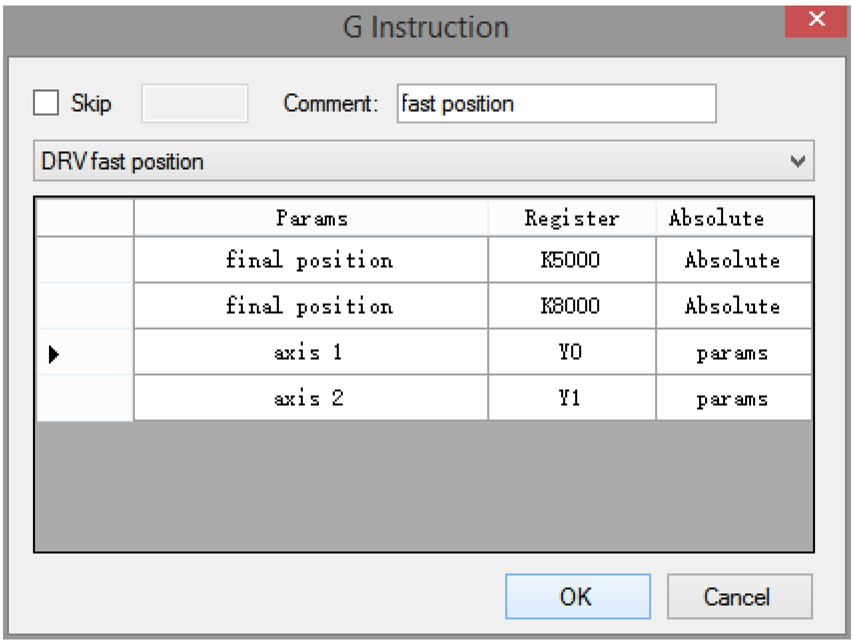
As shown in the figure below, the current position coordinates of the worktable are (1000,1000) and the target coordinates are (5000,8000). The two axes are Y0 and Y1, respectively. The default pulse speeds are all 5000. The acceleration and deceleration slopes are changed by 1000Hz for 30ms, and the pulse direction terminals are Y4 and Y5. Note: The above numerical units are pulse numbers.



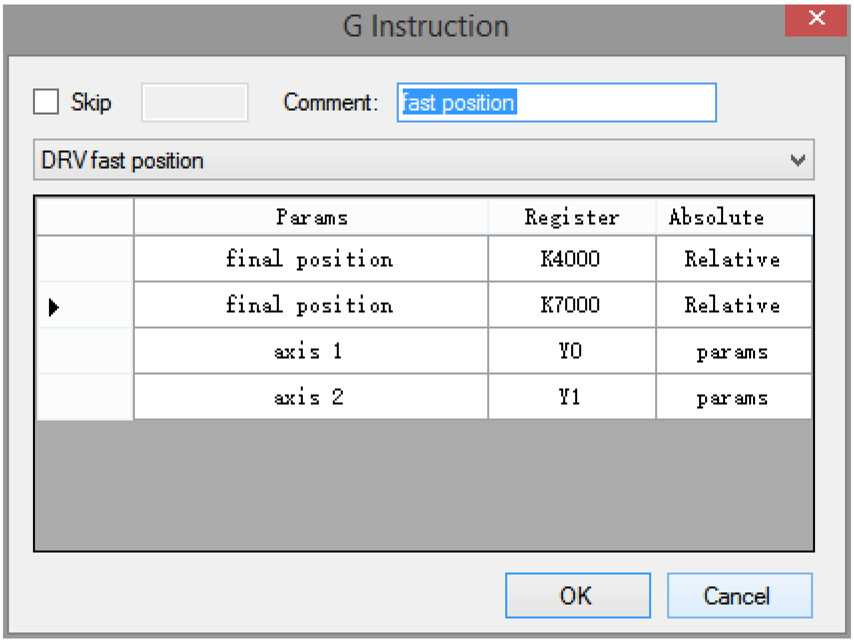
Ladder chart:



G item configurations:

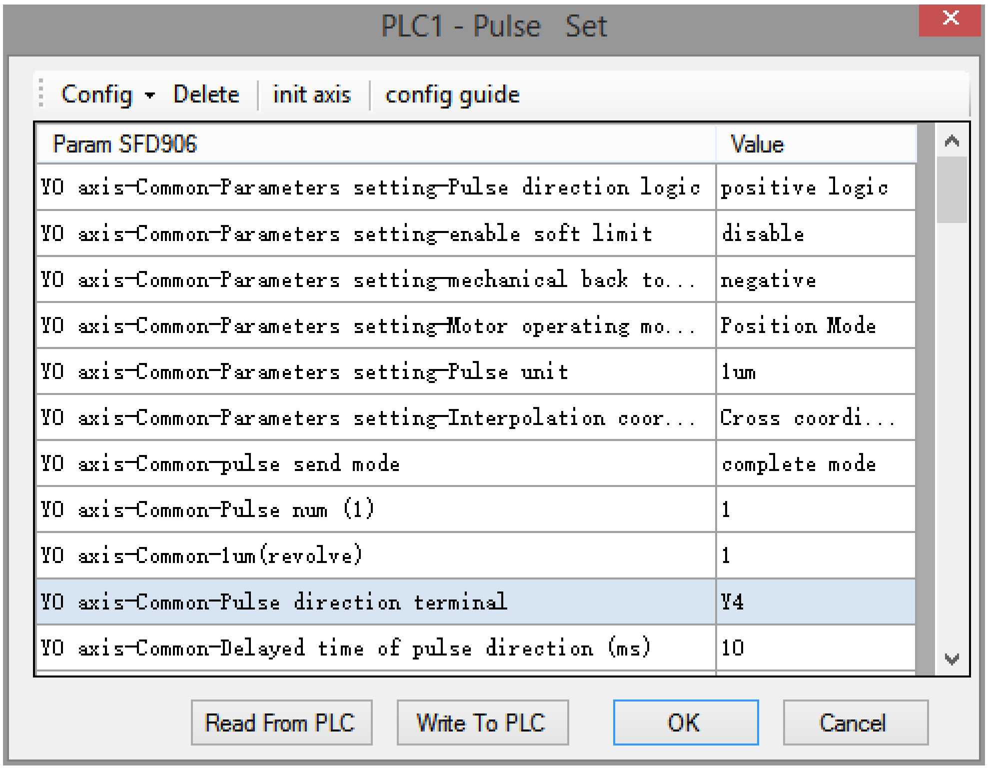


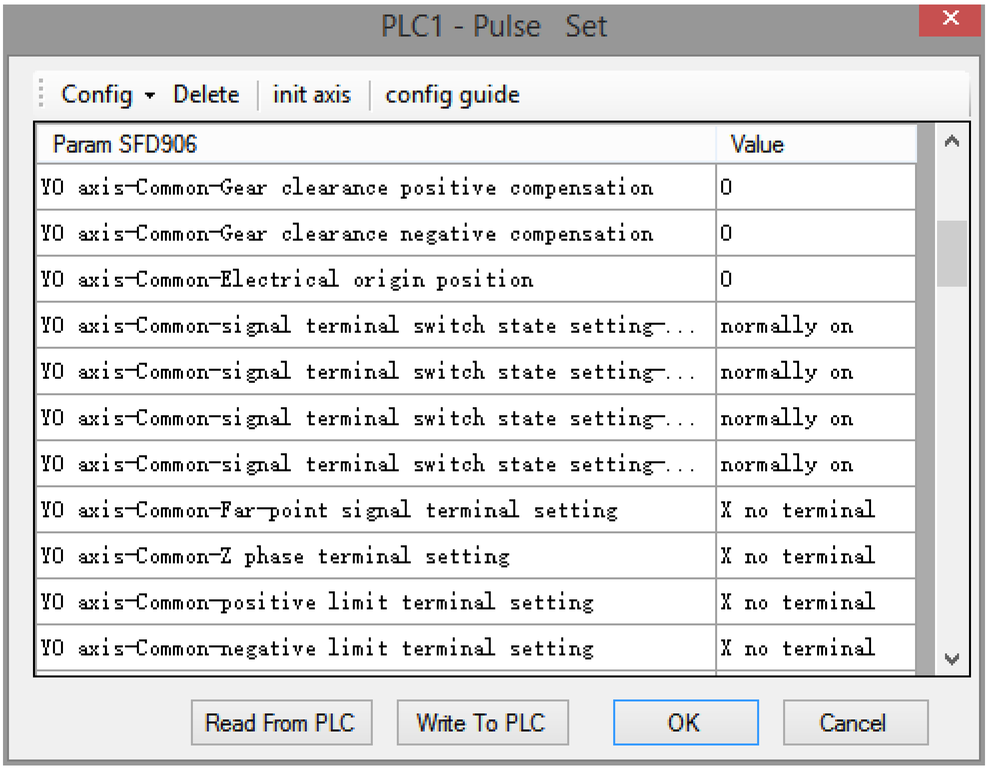
Absolute mode



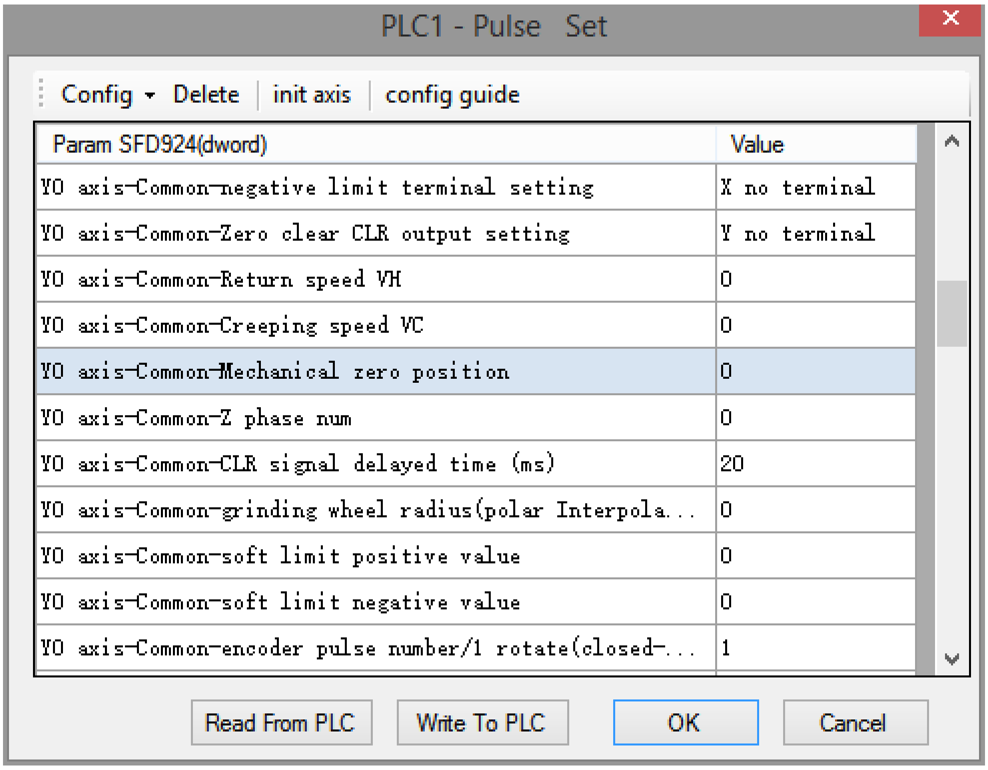
Relative mode

Axis 1(Y0) parameters:

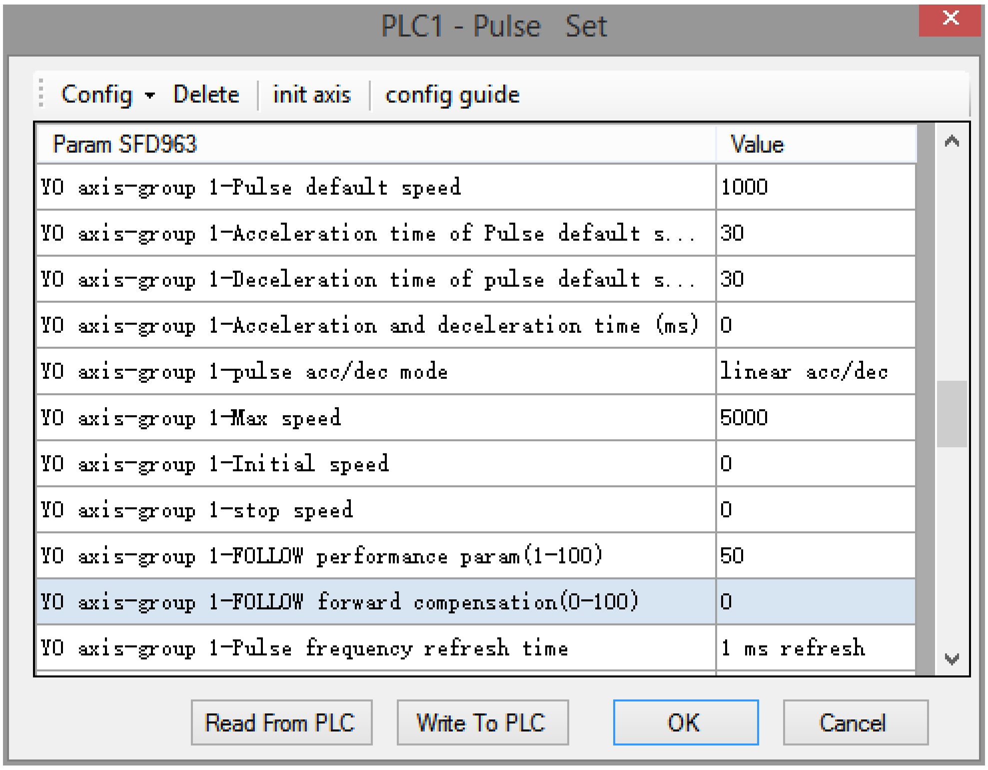




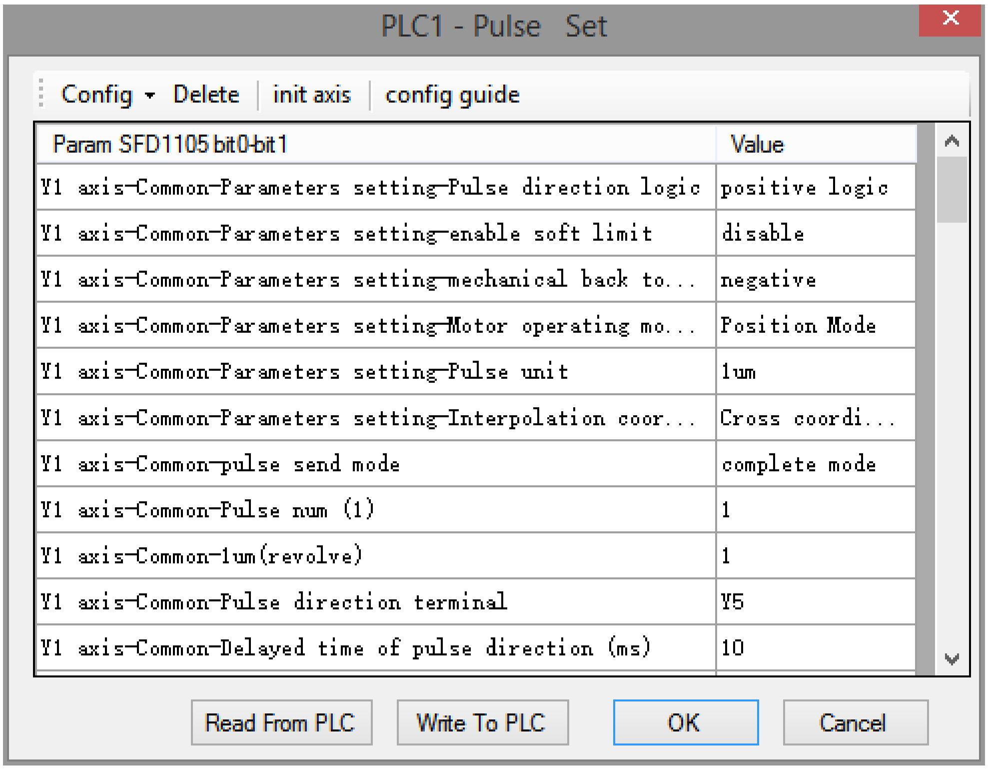
Param SFD906	Value
Y0 axis-Common-Gear clearance positive compensation	0
Y0 axis-Common-Gear clearance negative compensation	0
Y0 axis-Common-Electrical origin position	0
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-signal terminal switch state setting-...	normally on
Y0 axis-Common-Far-point signal terminal setting	X no terminal
Y0 axis-Common-Z phase terminal setting	X no terminal
Y0 axis-Common-positive limit terminal setting	X no terminal
Y0 axis-Common-negative limit terminal setting	X no terminal

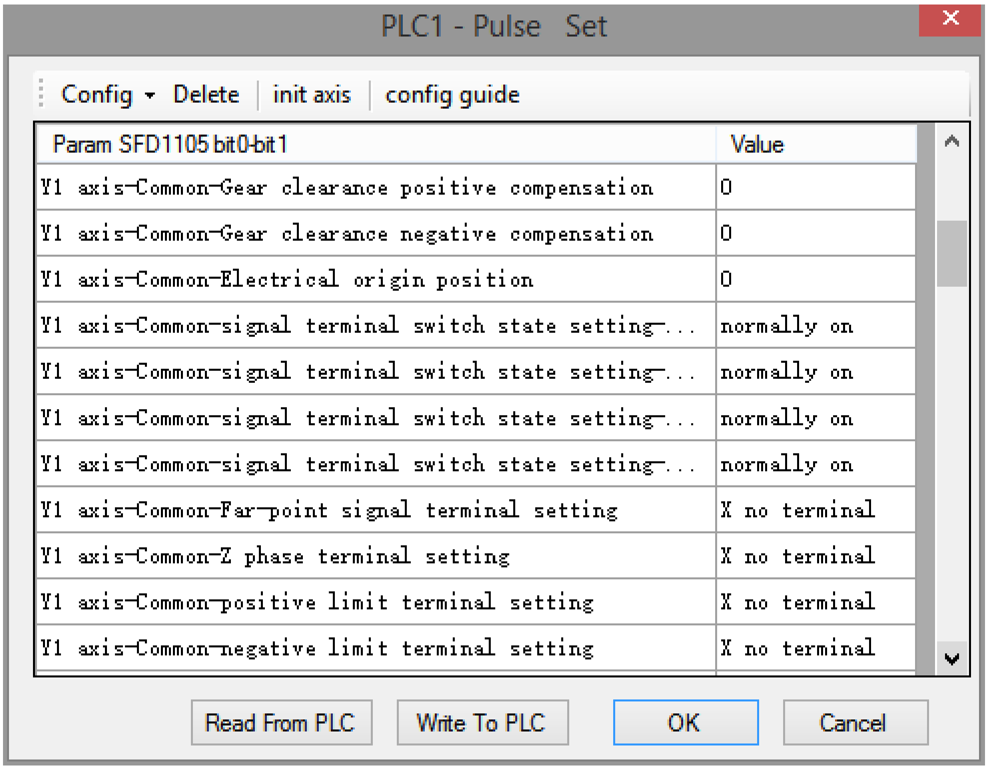


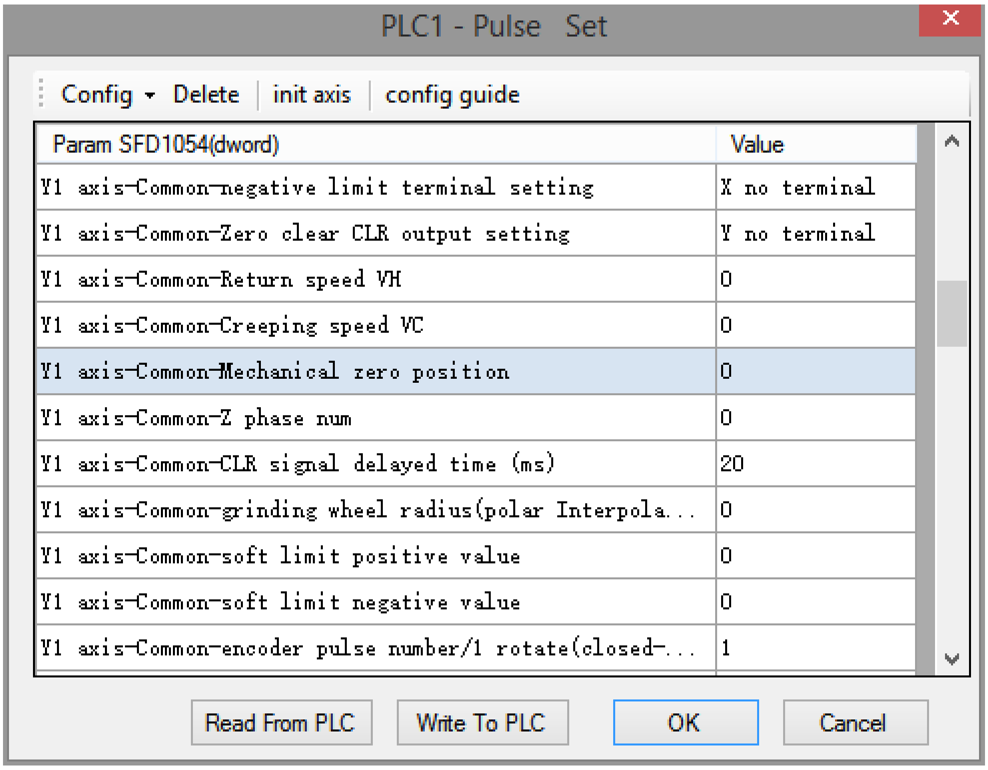
Param SFD924(dword)	Value
Y0 axis-Common-negative limit terminal setting	X no terminal
Y0 axis-Common-Zero clear CLR output setting	Y no terminal
Y0 axis-Common-Return speed VH	0
Y0 axis-Common-Creeping speed VC	0
Y0 axis-Common-Mechanical zero position	0
Y0 axis-Common-Z phase num	0
Y0 axis-Common-CLR signal delayed time (ms)	20
Y0 axis-Common-grinding wheel radius(polar Interpola...	0
Y0 axis-Common-soft limit positive value	0
Y0 axis-Common-soft limit negative value	0
Y0 axis-Common-encoder pulse number/1 rotate(closed-...	1

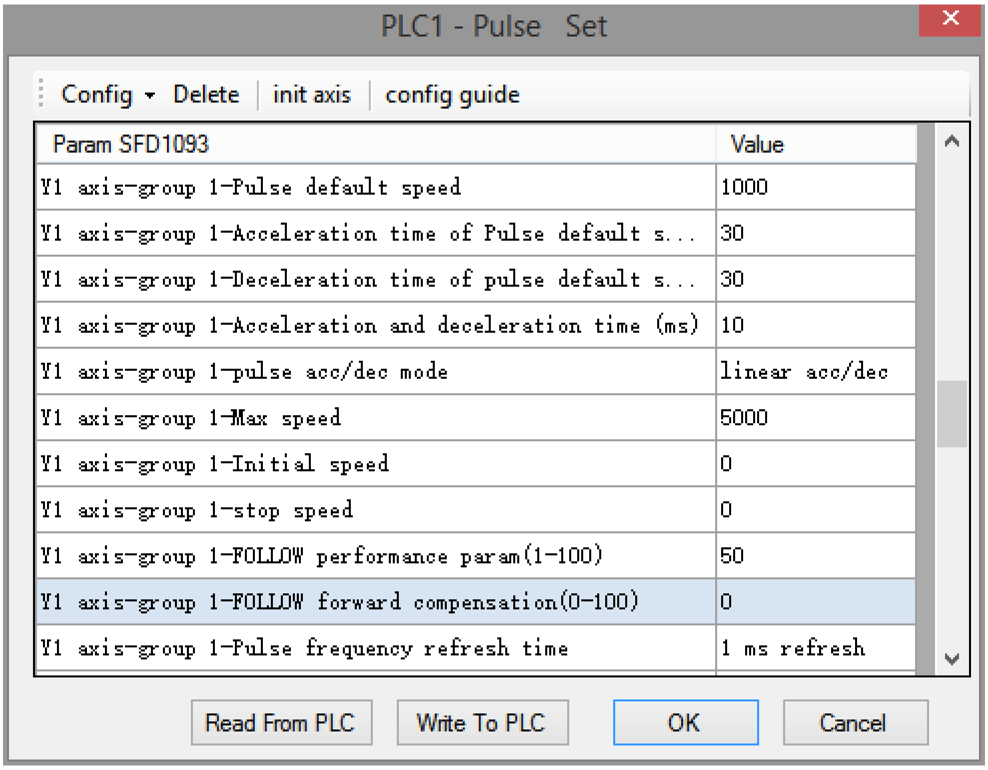


Axis 2 (Y1) parameters:




PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD1105 bit0-bit1
Value
Y1 axis-Common-Gear clearance positive compensation
0
Y1 axis-Common-Gear clearance negative compensation
0
Y1 axis-Common-Electrical origin position
0
Y1 axis-Common-signal terminal switch state setting-...
normally on
Y1 axis-Common-signal terminal switch state setting-...
normally on
Y1 axis-Common-signal terminal switch state setting-...
normally on
Y1 axis-Common-signal terminal switch state setting-...
normally on
Y1 axis-Common-Far-point signal terminal setting
X no terminal
Y1 axis-Common-Z phase terminal setting
X no terminal
Y1 axis-Common-positive limit terminal setting
X no terminal
Y1 axis-Common-negative limit terminal setting
X no terminal
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD1054(dword)
Value
Y1 axis-Common-negative limit terminal setting
X no terminal
Y1 axis-Common-Zero clear CLR output setting
Y no terminal
Y1 axis-Common-Return speed VH
0
Y1 axis-Common-Creeping speed VC
0
Y1 axis-Common-Mechanical zero position
0
Y1 axis-Common-Z phase num
0
Y1 axis-Common-CLR signal delayed time (ms)
20
Y1 axis-Common-grinding wheel radius(polar Interpola...
0
Y1 axis-Common-soft limit positive value
0
Y1 axis-Common-soft limit negative value
0
Y1 axis-Common-encoder pulse number/1 rotate(closed-...
1
Read From PLC
Write To PLC
OK
Cancel



2-4-2. Quick positioning (polar coordinates) [DRVR]

1. Instruction overview

Quick positioning (polar coordinates) instructions. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Quick positioning [DRVR]			
16-bit instruction	-	32-bit instruction	DRVR
Execute condition	Rise/fall edge of the coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis X target position	Double words, 32-bit
S1	Axis Y target position	Double words, 32-bit
D0	Pulse output port of axis X	Bit
D1	Pulse output port of axis Y	Bit

3. suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

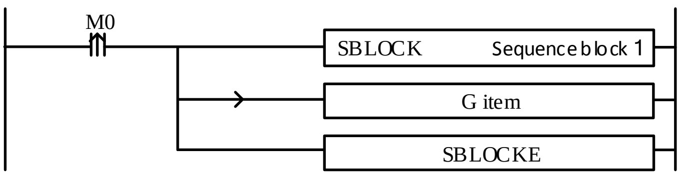
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

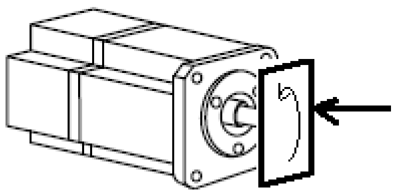
Related parameters	Setting	Note
Final position	Free to specify register address	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	Set in axis 1 system parameters	Must set
Default speed	Set in axis 1 group 1 parameters	Must set
Acceleration time	Set in axis 1 group 1 parameters	No need to set
Deceleration time	Set in axis 1 group 1 parameters	No need to set

Function and action

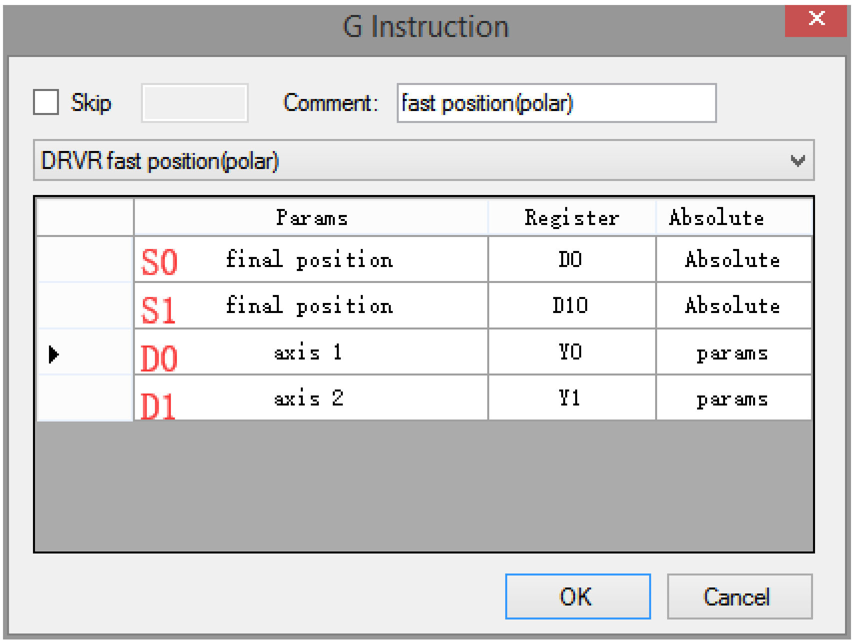
《instruction format》



Fast positioning (polar coordinates) instruction refers to the rotation axis of one axis, which rotates the workpiece on the rotating axis, and the forward and backward feed axis which is perpendicular to the rotating axis. When the rotating axis drives the workpiece to rotate, the feed axis processes the trajectory of the rotating workpiece through forward and backward processing. The trajectory of motion can include straight line and arc, and can be used in processing and grinding equipment.



Double click G item, it will pop up DRVR fast position(polar) instruction configuration panel, as shown below:



2-4-3. Linear interpolation [LIN]

There are three modes of linear interpolation, the following will introduce one by one.

Mode 1: LIN line

1. Instruction overview

Linear interpolation instruction, operate according to the set default speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Linear interpolation [LIN]			
16-bit instruction	-	32-bit instruction	LIN
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

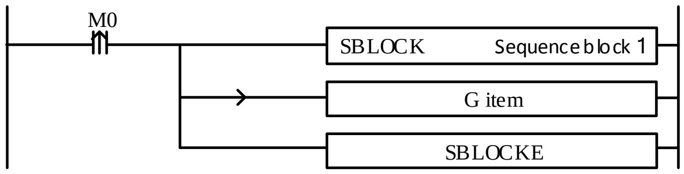
4. Parameter setting

Related parameters	Setting	Note
Final position	Free to specify register address	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Pulse output port of	Arbitrary specify pulse output point	Must set

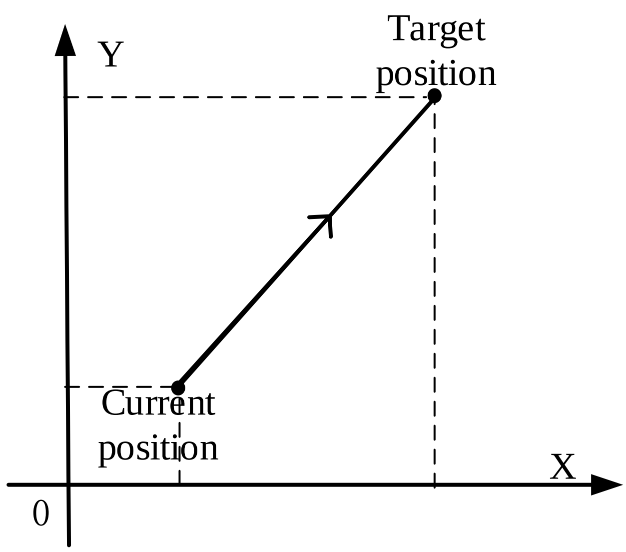
axis 1		
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	Set in axis 1 system parameters	Must set
Default speed	The synthetic speed of two axes, set in axis 1 group 2 parameters	Must set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

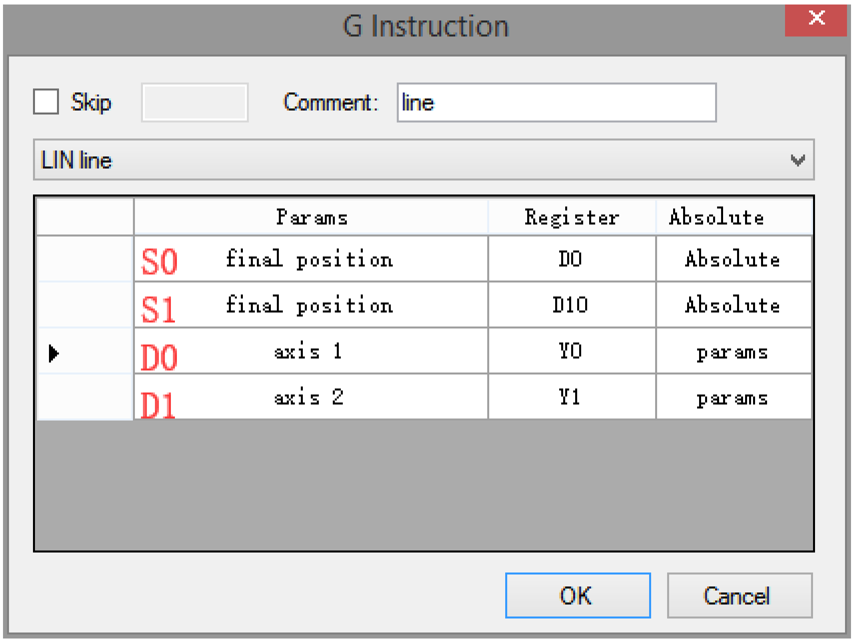


When the LIN instruction of linear interpolation (mode 1) is executed, the two axes will move rapidly from the current position to the target position at the highest synthetic speed of the two axes (the default speed set in axis 1 group 2 parameters). As shown in the following figure:

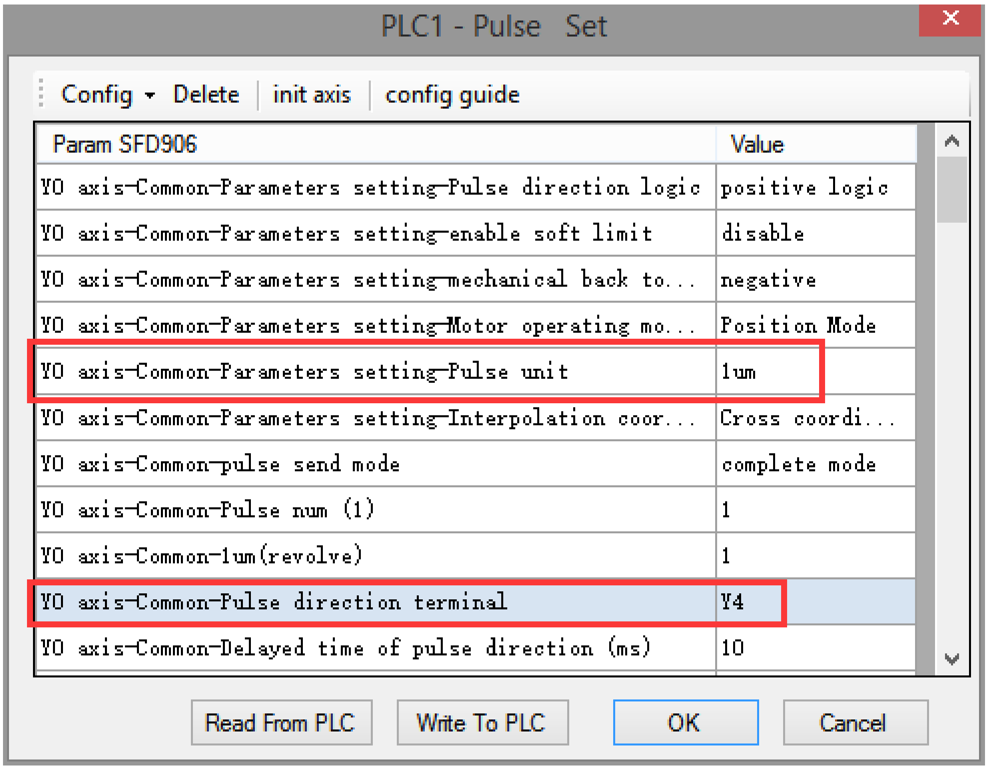


LIN linear interpolation

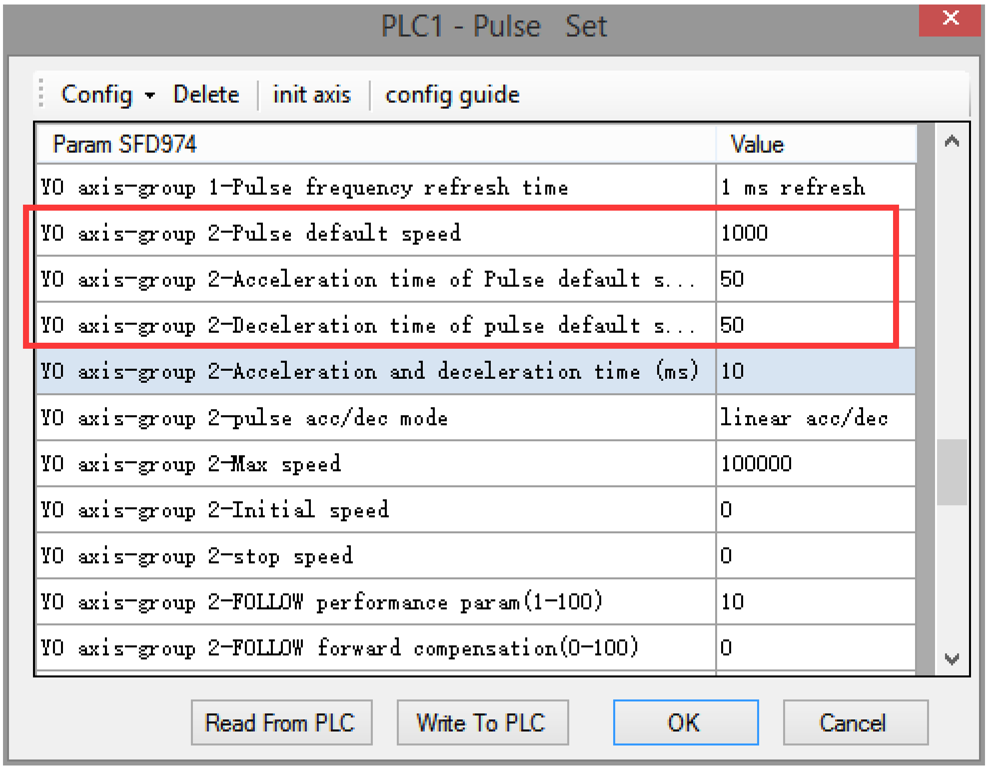
The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:



Instruction configuration



Axis Y0 system parameters (1)



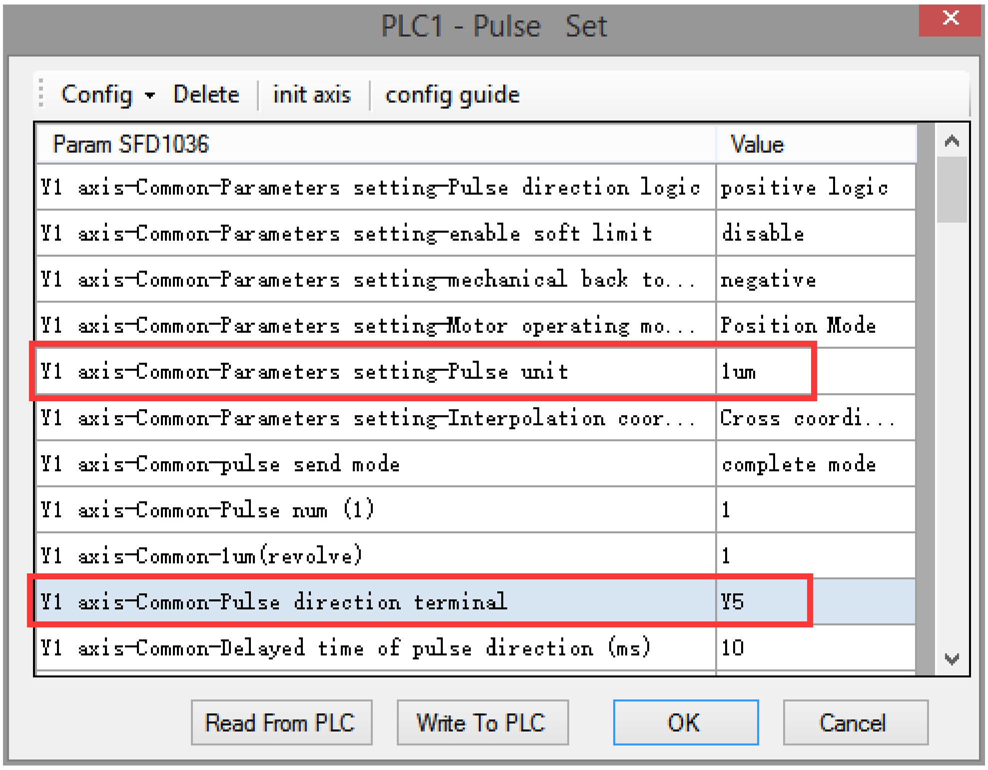
PLC1 - Pulse Set

Config Delete init axis config guide

Param SFD974	Value
Y0 axis-group 1-Pulse frequency refresh time	1 ms refresh
Y0 axis-group 2-Pulse default speed	1000
Y0 axis-group 2-Acceleration time of Pulse default s...	50
Y0 axis-group 2-Deceleration time of pulse default s...	50
Y0 axis-group 2-Acceleration and deceleration time (ms)	10
Y0 axis-group 2-pulse acc/dec mode	linear acc/dec
Y0 axis-group 2-Max speed	100000
Y0 axis-group 2-Initial speed	0
Y0 axis-group 2-stop speed	0
Y0 axis-group 2-FOLLOW performance param(1-100)	10
Y0 axis-group 2-FOLLOW forward compensation(0-100)	0

Read From PLC Write To PLC OK Cancel

Axis Y0 system parameters (2)



PLC1 - Pulse Set

Config Delete init axis config guide

Param SFD1036	Value
Y1 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y1 axis-Common-Parameters setting-enable soft limit	disable
Y1 axis-Common-Parameters setting-mechanical back to...	negative
Y1 axis-Common-Parameters setting-Motor operating mo...	Position Mode
Y1 axis-Common-Parameters setting-Pulse unit	1um
Y1 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y1 axis-Common-pulse send mode	complete mode
Y1 axis-Common-Pulse num (1)	1
Y1 axis-Common-1um(revolve)	1
Y1 axis-Common-Pulse direction terminal	Y5
Y1 axis-Common-Delayed time of pulse direction (ms)	10

Read From PLC Write To PLC OK Cancel

Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3

for other optional ports.

- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 500, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute LIN command and move to the target position at the default speed of 1000Hz:

(1) If the final position is absolute mode, the target position is (5000,2000);
(2) When the final position is in the relative mode, the target position is (5500,3000).

- When the LIN instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 2: LIN line VM

1. Instruction overview

Linear interpolation instruction, operate according to the set maximum synthetic speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Linear interpolation [LIN]			
16-bit instruction	-	32-bit instruction	LIN
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	maximum synthetic speed of axis 1 and 2	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

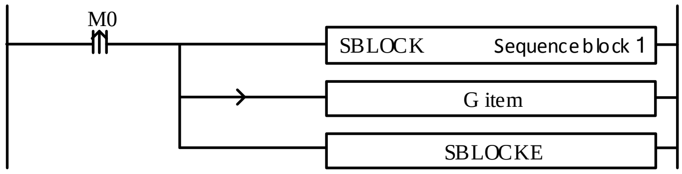
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

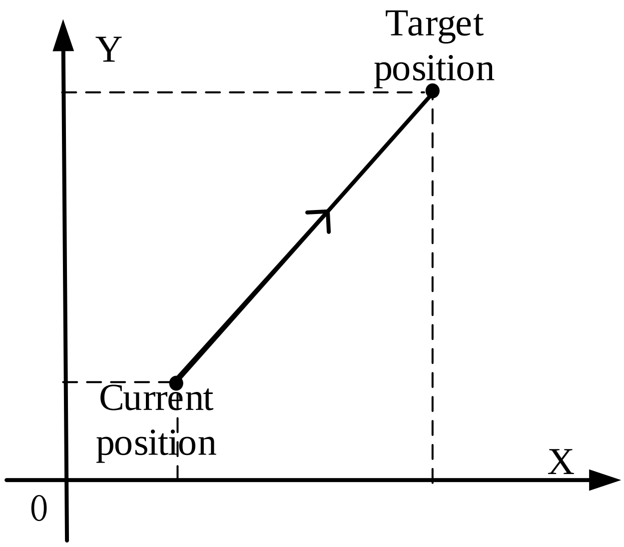
Related parameters	Setting	Note
Final position	Free to specify register address	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Max speed	Specify the maximum smooth running speed of the two-axis combination, and specify any address.	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

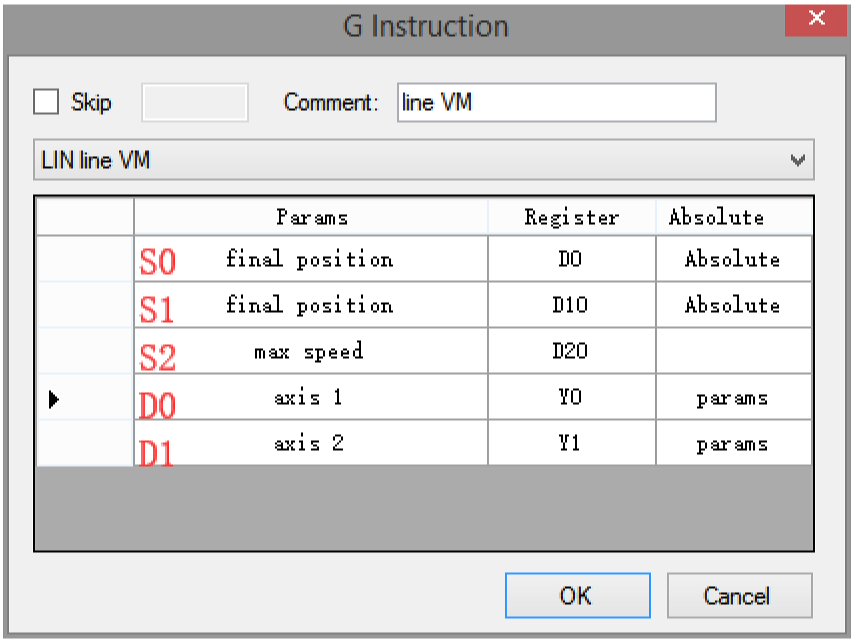


When the LIN instruction of linear interpolation (mode 2) is executed, the two axes will move rapidly from the current position to the target position at the set max synthetic speed. As shown in the following figure:

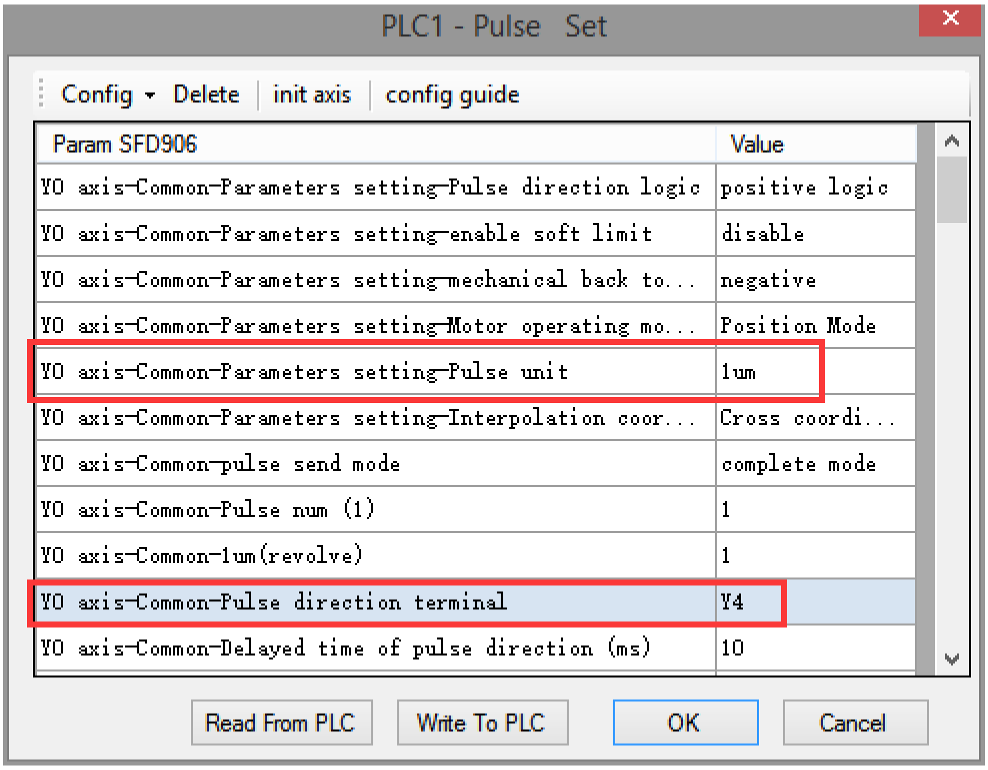


LIN linear interpolation

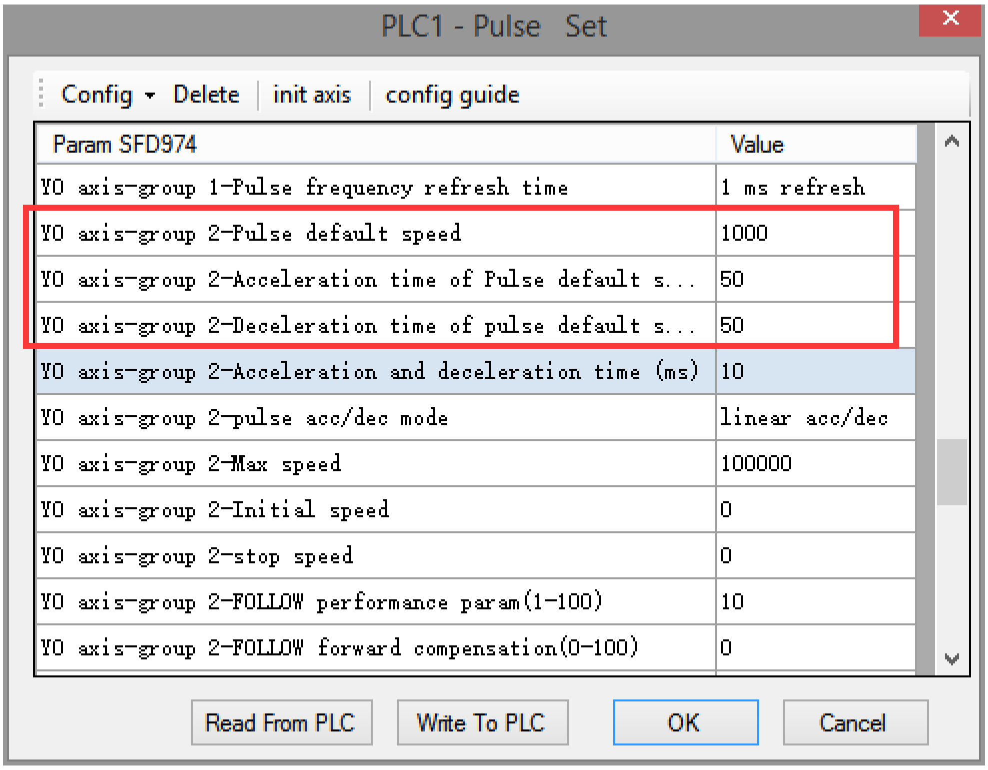
The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:



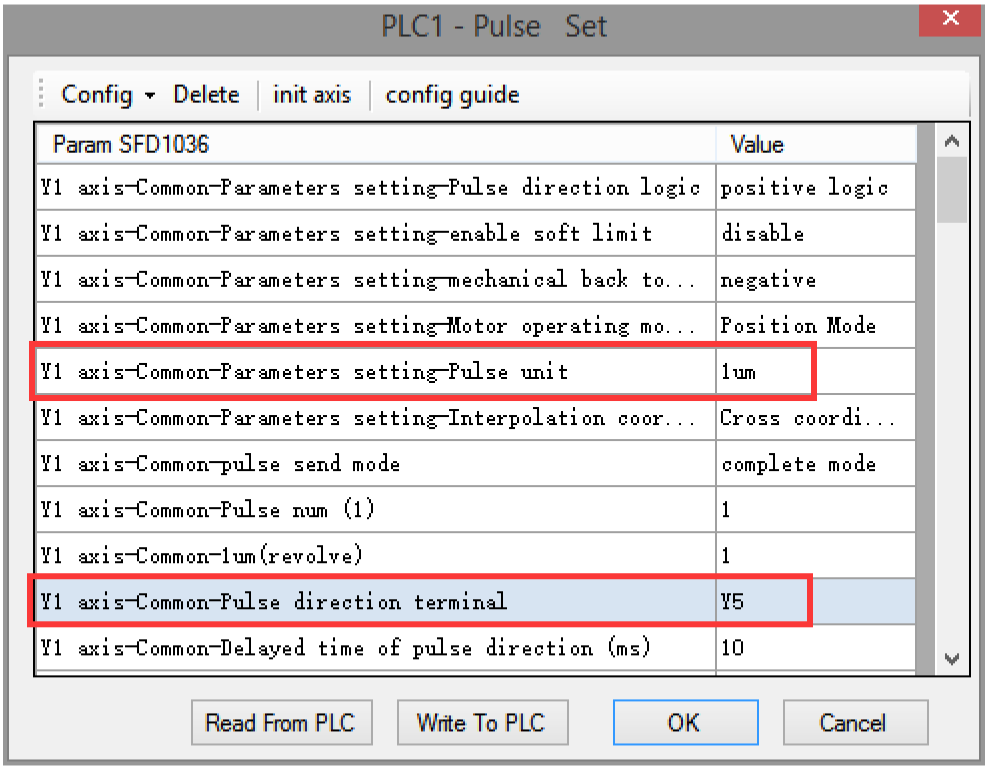
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the max speed.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3

for other optional ports.

- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 500, HSD6 = 1000, D0 = 5000, D10 = 2000, D20 = 2000, when M0 rises, execute LIN command and move to the target position at the speed of 2000Hz:

(1) If the final position is absolute mode, the target position is (5000,2000);

(2) When the final position is in the relative mode, the target position is (5500,3000).

- When the LIN instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 3: LIN line VBEM

1. Instruction overview

Linear interpolation instruction, operate according to the set maximum synthetic speed, start speed and stop speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Linear interpolation [LIN]			
16-bit instruction	-	32-bit instruction	LIN
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Start speed of axis 1 and 2	Double words, 32-bit
S3	Stop speed of axis 1 and 2	Double words, 32-bit
S4	maximum synthetic speed of axis 1 and 2	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							
S4	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

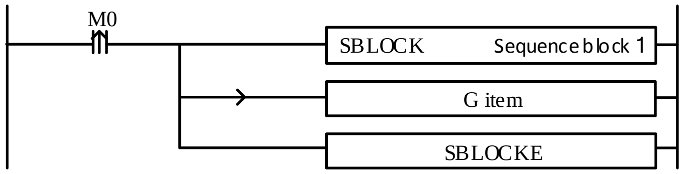
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

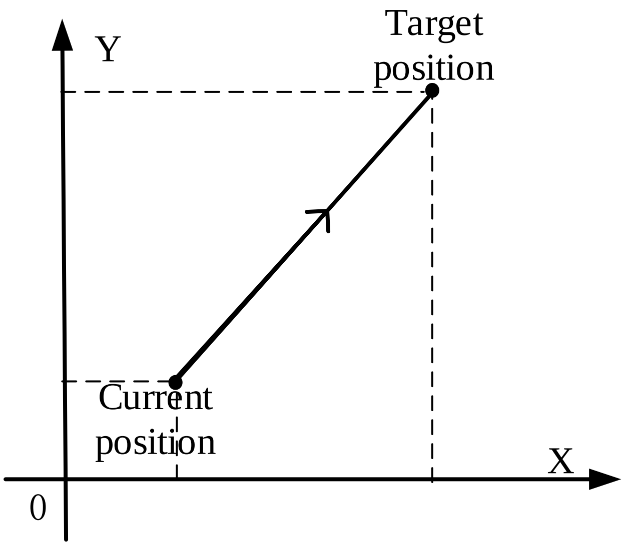
Related parameters	Setting	Note
Final position	Free to specify register address	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Start speed	Start speed at the starting point of the two axes	Must set
Stop speed	Stop speed at the end point of the two axes	Must set
Max speed	Specify the maximum smooth running speed of the two-axis combination, and specify any address.	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

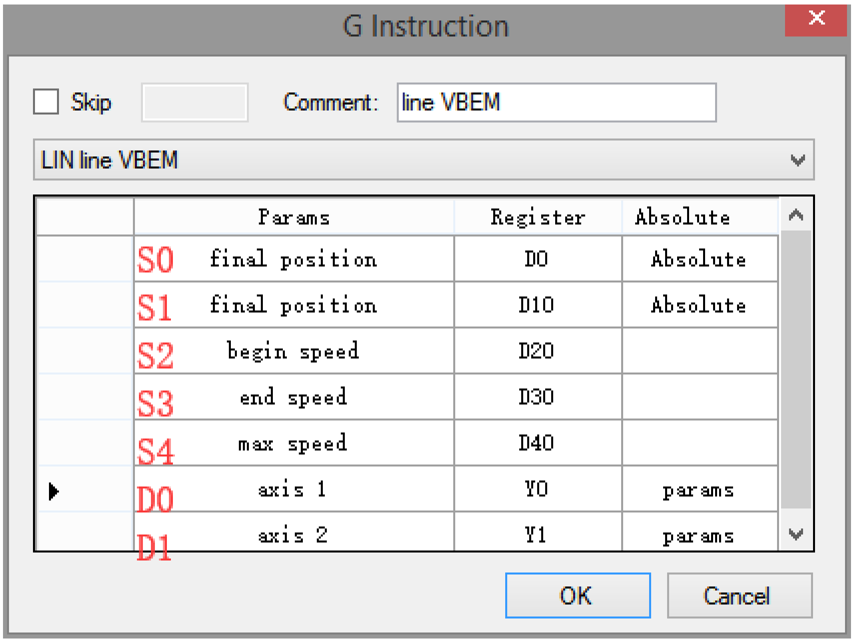


When the LIN instruction of linear interpolation (mode 3) is executed, the two axes will move rapidly from the current position to the target position at the set max synthetic speed, start speed and stop speed. As shown in the following figure:

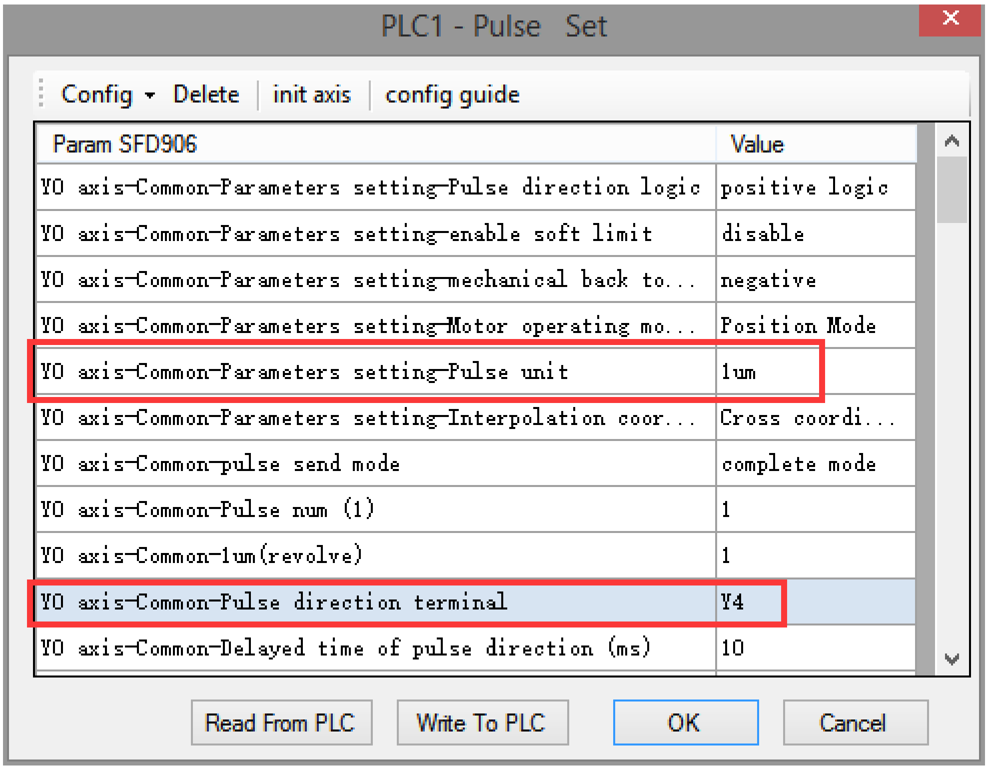


LIN linear interpolation

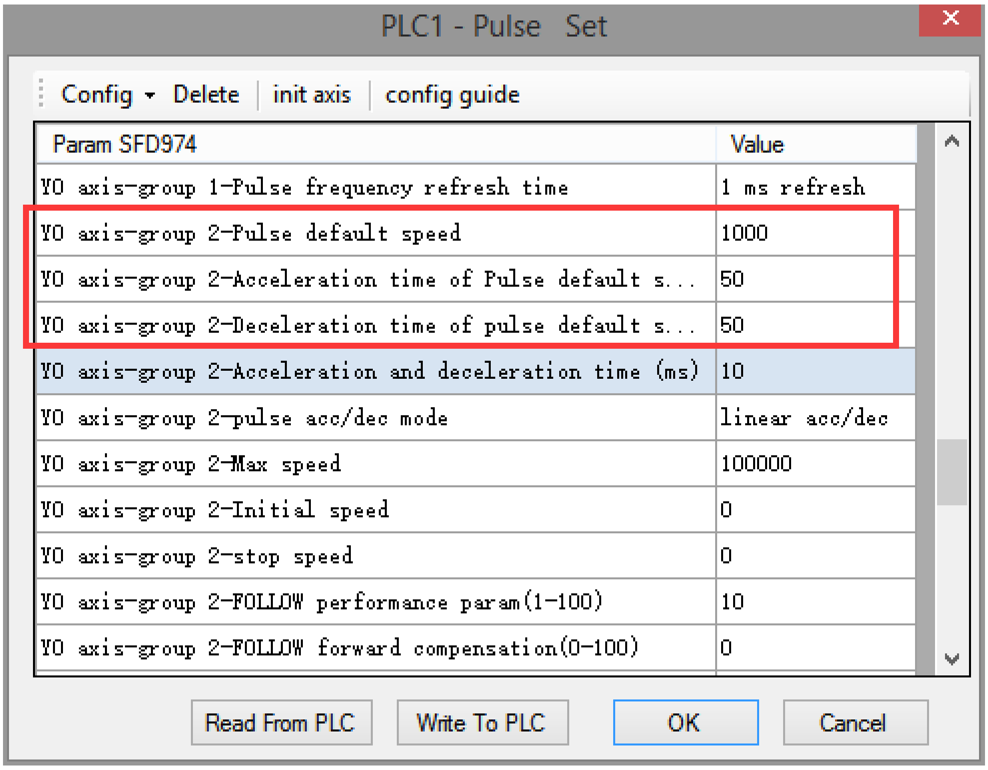
The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:



Instruction configuration

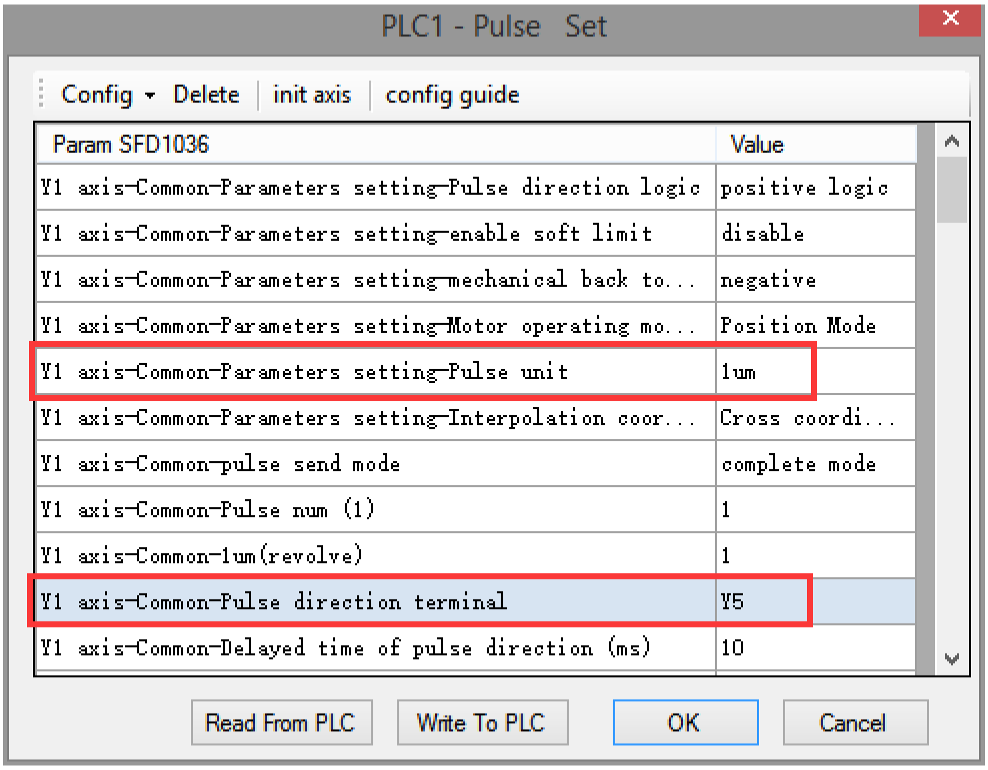


Axis Y0 system parameters (1)



Param SFD974	Value
Y0 axis-group 1-Pulse frequency refresh time	1 ms refresh
Y0 axis-group 2-Pulse default speed	1000
Y0 axis-group 2-Acceleration time of Pulse default s...	50
Y0 axis-group 2-Deceleration time of pulse default s...	50
Y0 axis-group 2-Acceleration and deceleration time (ms)	10
Y0 axis-group 2-pulse acc/dec mode	linear acc/dec
Y0 axis-group 2-Max speed	100000
Y0 axis-group 2-Initial speed	0
Y0 axis-group 2-stop speed	0
Y0 axis-group 2-FOLLOW performance param(1-100)	10
Y0 axis-group 2-FOLLOW forward compensation(0-100)	0

Axis Y0 system parameters (2)



Param SFD1036	Value
Y1 axis-Common-Parameters setting-Pulse direction logic	positive logic
Y1 axis-Common-Parameters setting-enable soft limit	disable
Y1 axis-Common-Parameters setting-mechanical back to...	negative
Y1 axis-Common-Parameters setting-Motor operating mo...	Position Mode
Y1 axis-Common-Parameters setting-Pulse unit	1um
Y1 axis-Common-Parameters setting-Interpolation coor...	Cross coordi...
Y1 axis-Common-pulse send mode	complete mode
Y1 axis-Common-Pulse num (1)	1
Y1 axis-Common-1um(revolve)	1
Y1 axis-Common-Pulse direction terminal	Y5
Y1 axis-Common-Delayed time of pulse direction (ms)	10

Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the start speed, D30 specifies the stop speed, D40 specifies the max speed.

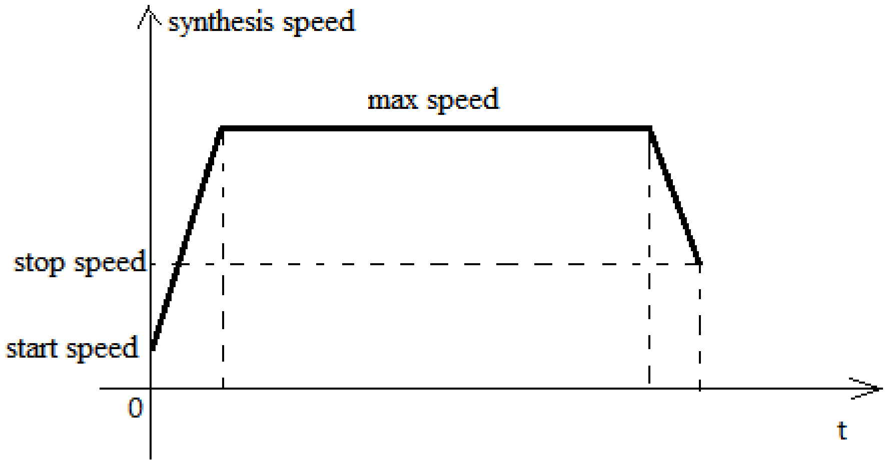
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
  - Assuming HSD2 = 500, HSD6 = 1000, D0 = 5000, D10 = 2000, D20 = 100, D30 = 50, D40 = 2000, when M0 rises, execute LIN command, accelerate from the starting point at 100Hz to 2000 Hz and stop at 50Hz after moving to the target position.

(1) If the final position is absolute mode, the target position is (5000,2000);

(2) When the final position is in the relative mode, the target position is (5500,3000).

- When the LIN instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Note: In this mode, the start speed (S2), the stop speed (S3) and the max speed (S4) are all expressed as the two-axis synthesis speed, as shown in the following figure:

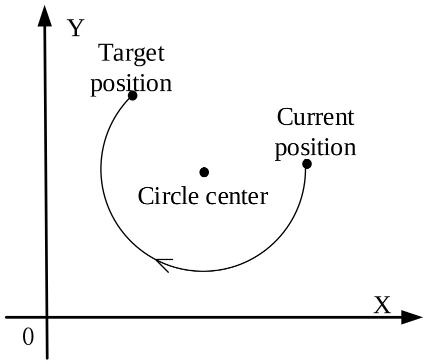


When there are multiple continuous linear/arc interpolation instructions and the speed between them needs to be constant and jump directly, the stop speed and maximum speed of the previous linear/arc interpolation can be set the same as the start speed and maximum speed of the next segment.

When the third mode is used, the initial and stop speed in the pulse parameter configuration tables of axis 1 and axis 2 are only effective for calculating the slope of pulse acceleration and deceleration.

2-4-4. Clockwise arc [CW]

CW interpolation mainly determines the arc through the current position of the arc, the target position and the coordinates of the center of the circle, as shown in the following figure:



From the above figure, we can see that when we need to draw a whole circle, we only need to set the target position to the current position. CW has three modes. The usage of CW is described below.

Mode 1: CW clockwise

1. Instruction overview

Clockwise arc interpolation instruction, operate according to the set default speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Clockwise arc interpolation [CW]			
16-bit instruction	-	32-bit instruction	CW
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the center position of axis 1 (always relative to the starting coordinates)	Double words, 32-bit
S3	Specify the center position of axis 2 (always relative to the starting coordinates)	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

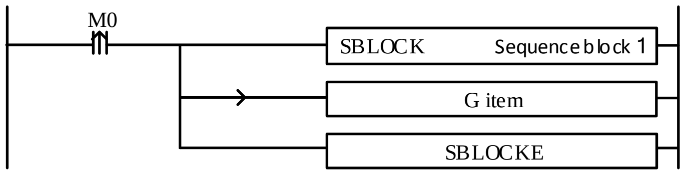
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

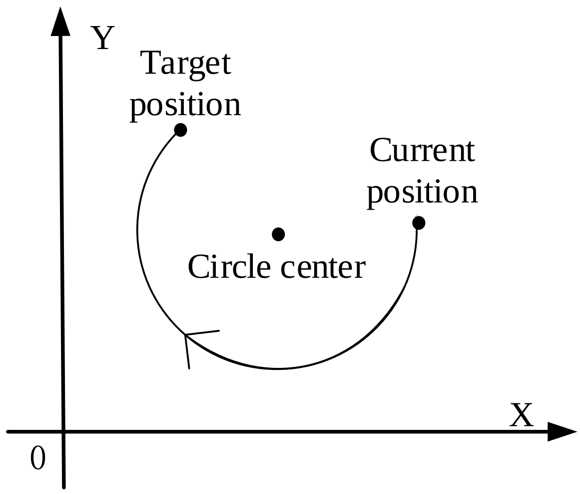
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Circle center position	The position of the center is determined by the position of the starting point and the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	Must set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



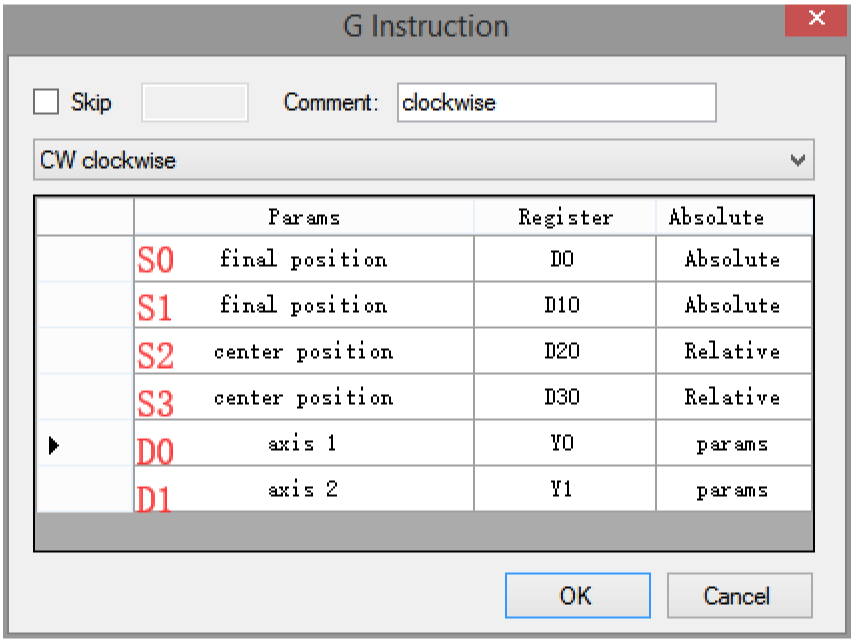
When the CW instruction of arc interpolation (mode 1) is executed, the two axes will run at the highest synthesis speed. As shown in the following figure:



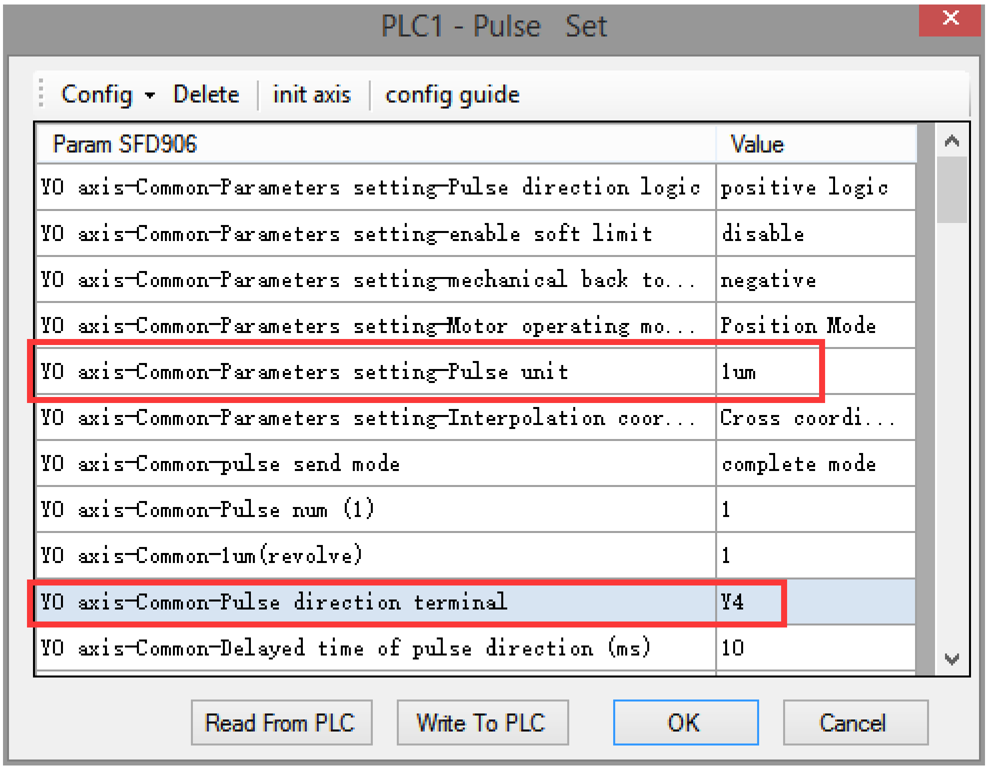
CW clockwise arc interpolation

The parameter configuration is shown in the following figure:

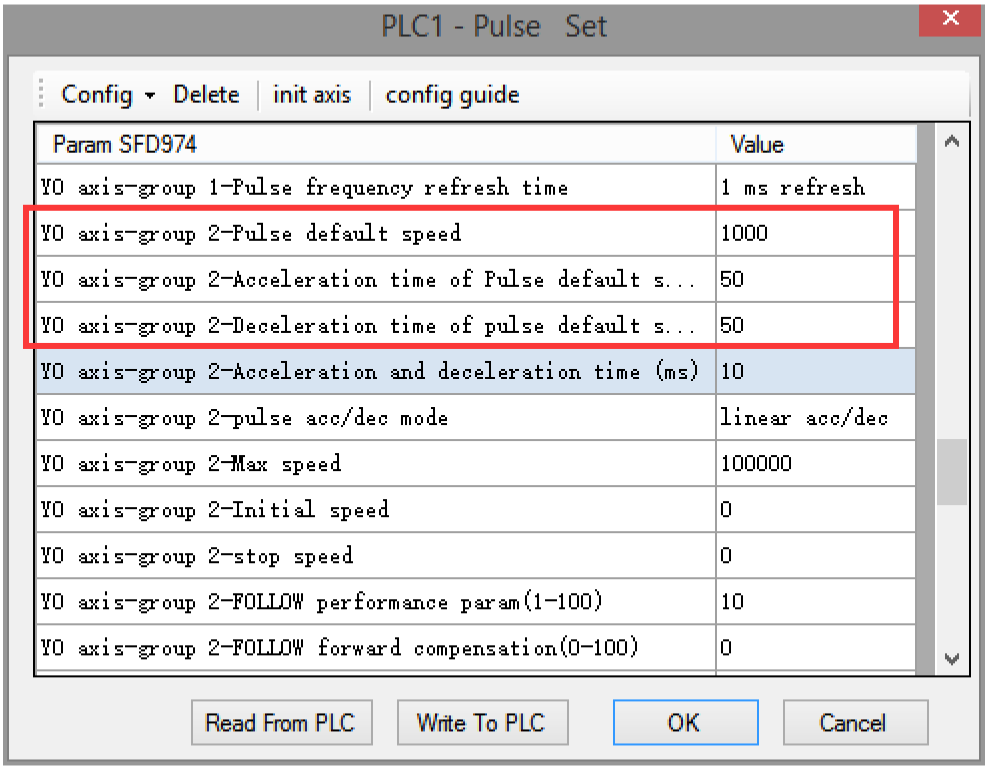
Double-click G item and pop up the configuration panel. Set it as follows:



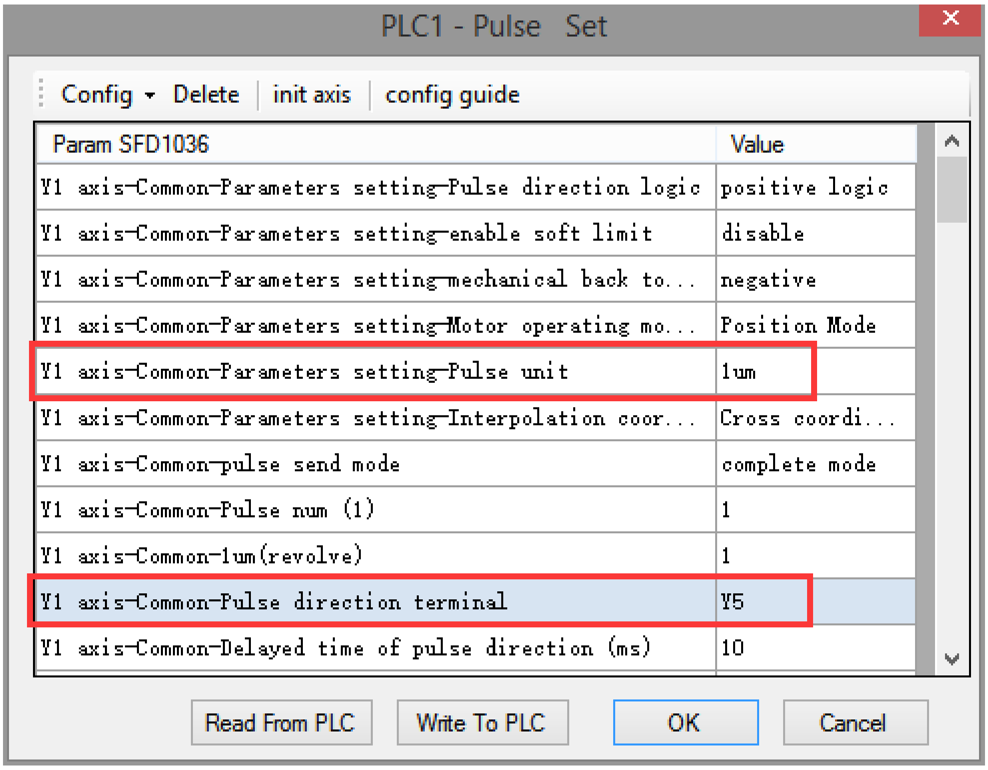
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle center of axis 1, D30 specifies the circle center of axis 2.

- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute CW command, move from the starting position (1000, 1000) to the target position at the default speed of 1000Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000), the center position is (3000,1500), and D20 = 2000, D30 = 500.

(2) When the end point is in the relative mode, the target position is (6000,3000), the center position is (3500,2000), and D20 = 2500, D30 = 1000.

- When the CW instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 2: CW clockwise VM

1. Instruction overview

Clockwise arc interpolation instruction, operate according to the set maximum synthetic speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Clockwise arc interpolation [CW]			
16-bit instruction	-	32-bit instruction	CW
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the center position of axis 1 (always relative to the starting coordinates)	Double words, 32-bit
S3	Specify the center position of axis 2 (always relative to the starting coordinates)	Double words, 32-bit
S4	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							
S4	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

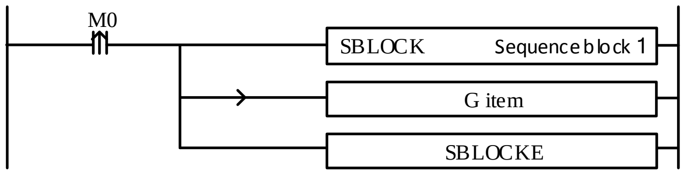
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

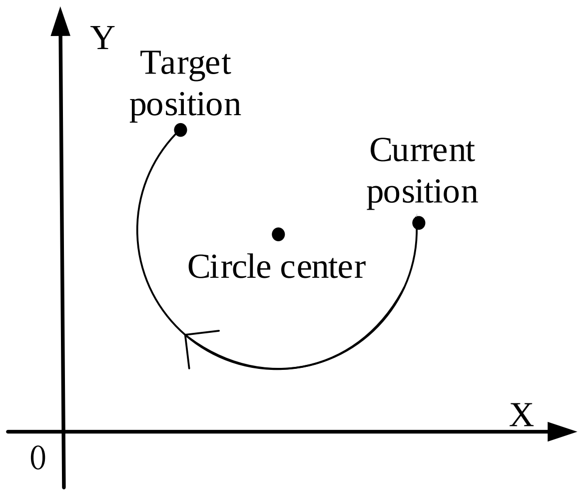
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Circle center position	The position of the center is determined by the position of the starting point and the end point	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

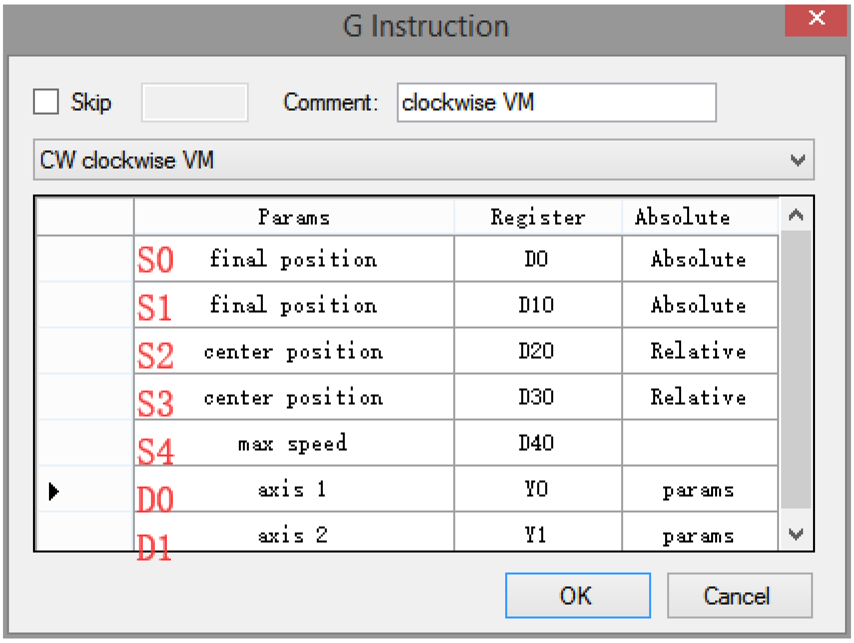


When the CW instruction of arc interpolation (mode 2) is executed, the two axes will run at the set max synthesis speed. As shown in the following figure:

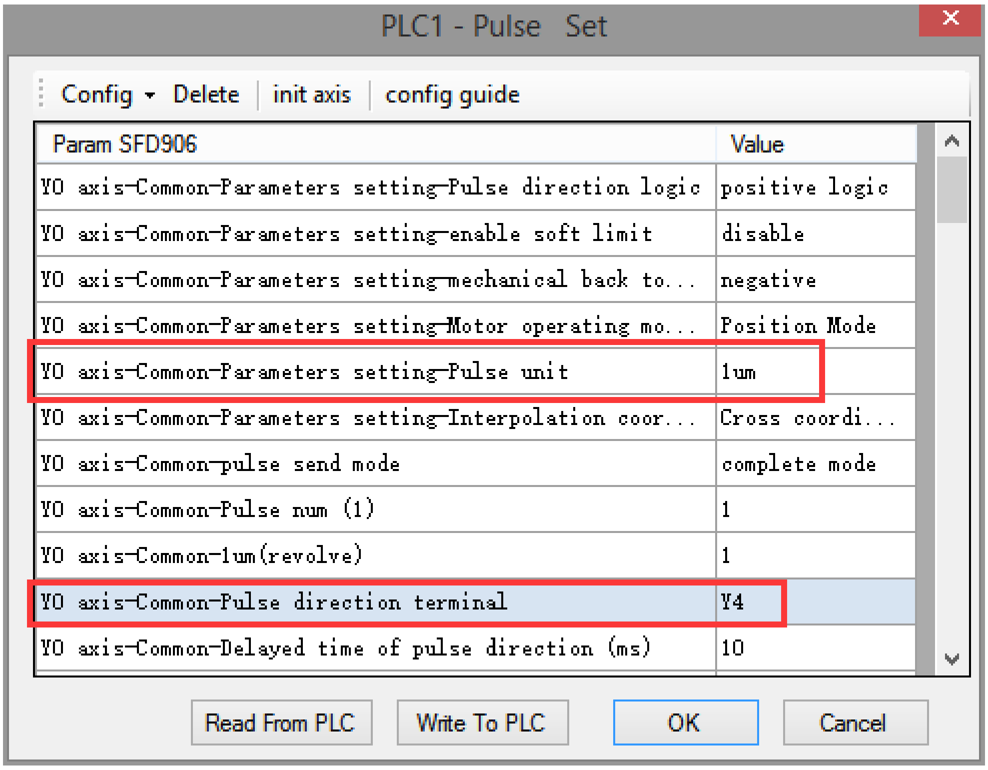


CW clockwise arc interpolation

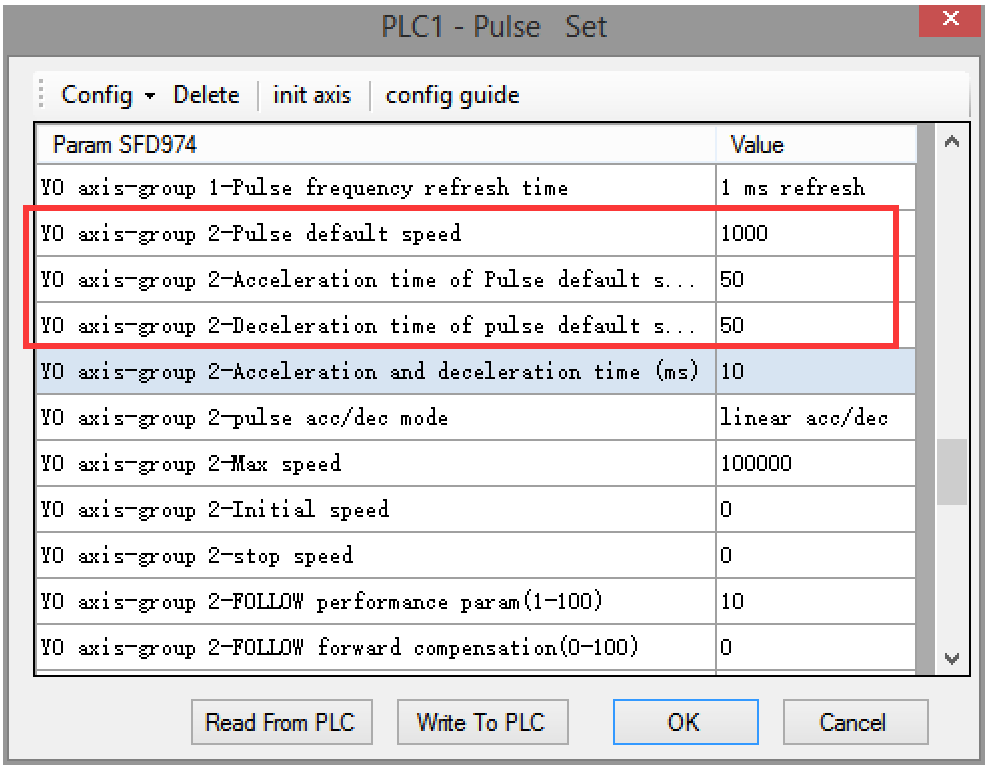
The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:



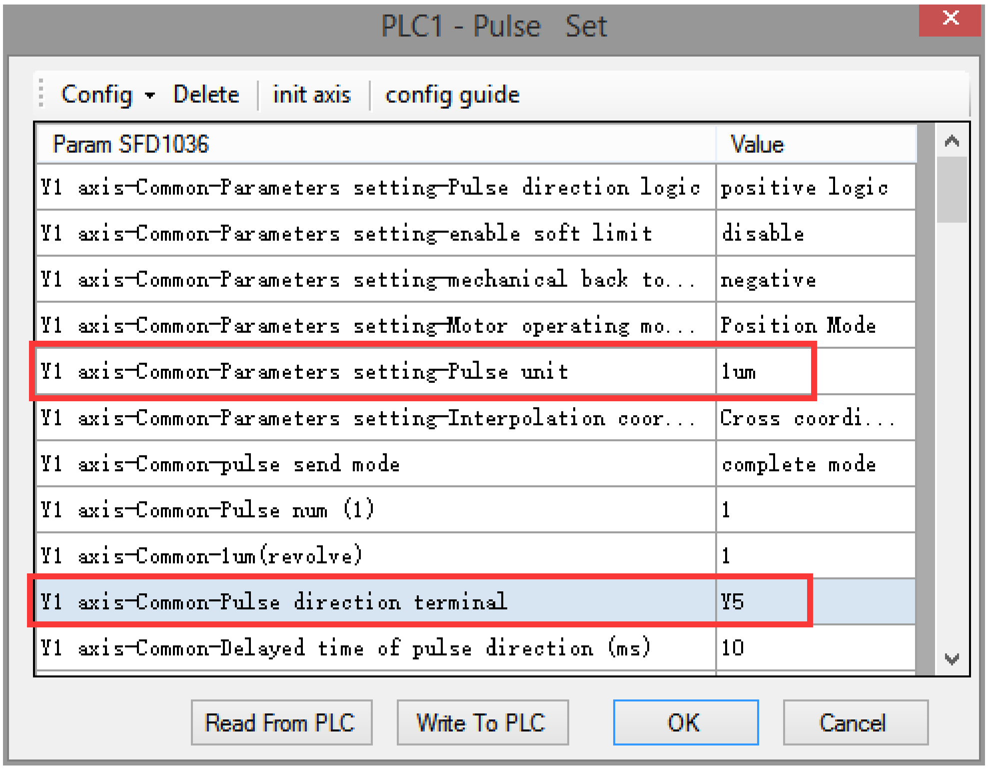
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle center of axis 1, D30 specifies the circle center of axis 2, D40 specifies the max speed.

- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 500Hz, when M0 rises, execute CW command, move from the starting position (1000, 1000) to the target position at the max speed of 500Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000), the center position is (3000,1500), and D20 = 2000, D30 = 500.

(2) When the end point is in the relative mode, the target position is (6000,3000), the center position is (3500,2000), and D20 = 2500, D30 = 1000.

- When the CW instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 3: CW clockwise VBEM

1. Instruction overview

Clockwise arc interpolation instruction, operate according to the set maximum synthetic speed, start speed and stop speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Clockwise arc interpolation [CW]			
16-bit instruction	-	32-bit instruction	CW
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the center position of axis 1 (always relative to the starting coordinates)	Double words, 32-bit
S3	Specify the center position of axis 2 (always relative to the starting coordinates)	Double words, 32-bit
S4	Specify the starting speed at the starting point of the two axes	Double words, 32-bit

S5	Specify the stop speed at the end point of the two axes	Double words, 32-bit
S6	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0~S6	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

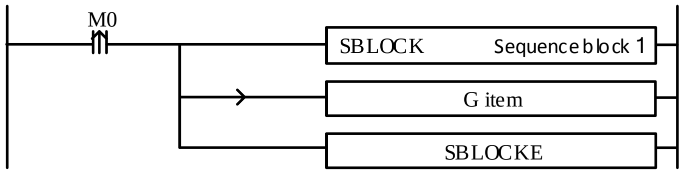
4. Parameter setting

Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Circle center position	The position of the center is determined by the position of the starting point and the end point	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Start speed	The start speed from the starting point	Must set
Stop speed	The stop speed at the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set

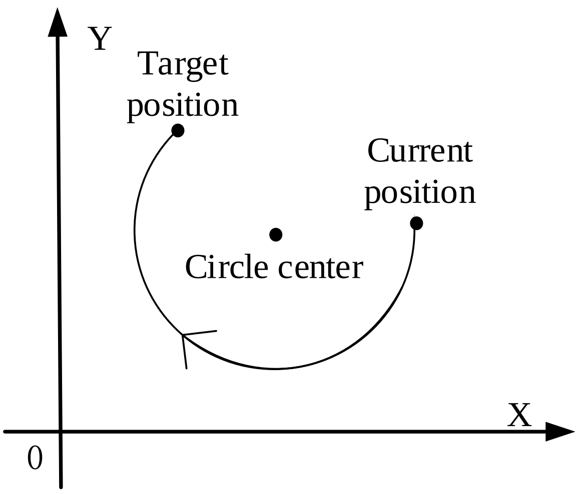
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

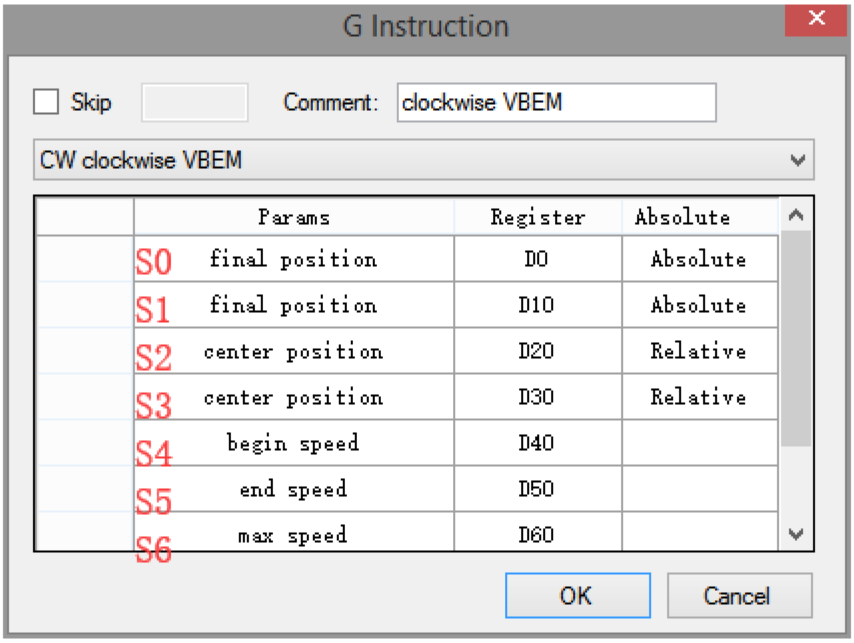


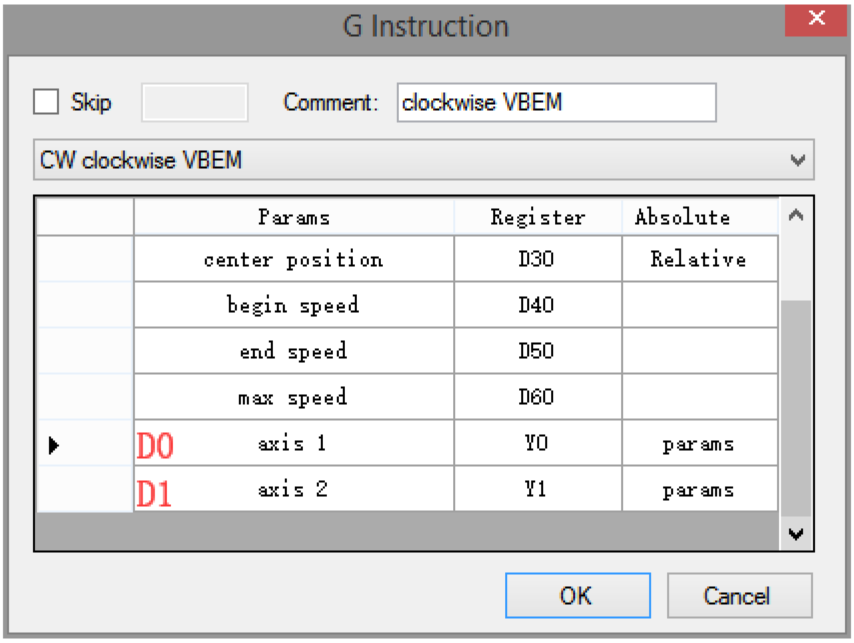
When the CW instruction of arc interpolation (mode 3) is executed, the two axes will run at the set max synthesis speed, start speed and stop speed. As shown in the following figure:



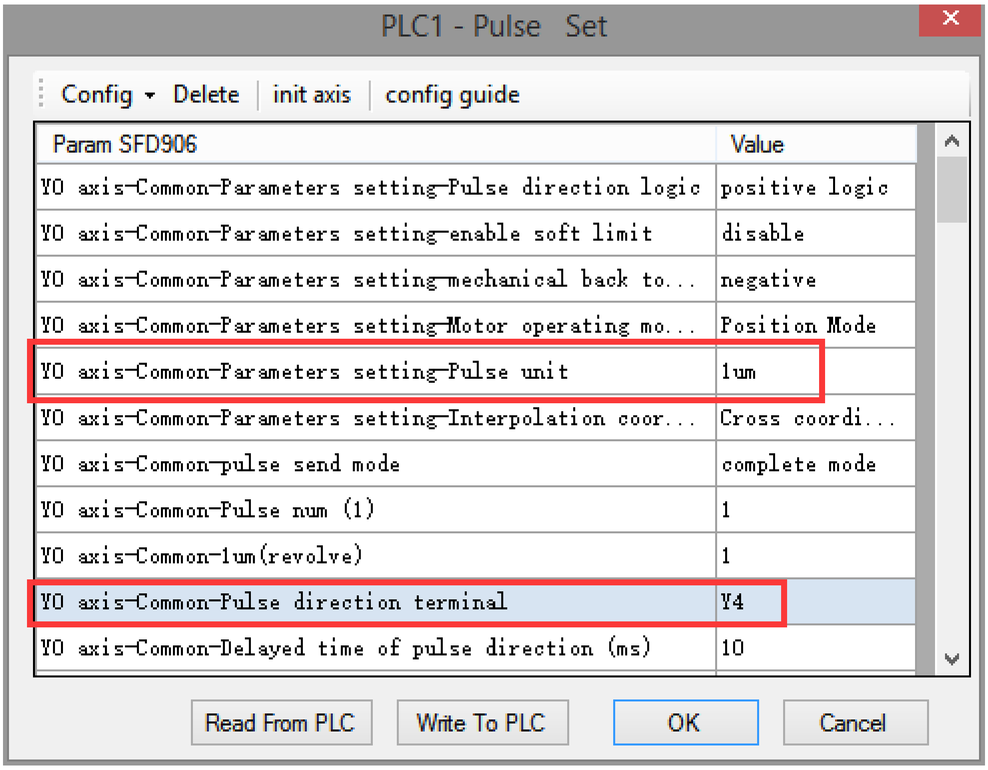
CW clockwise arc interpolation

The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:

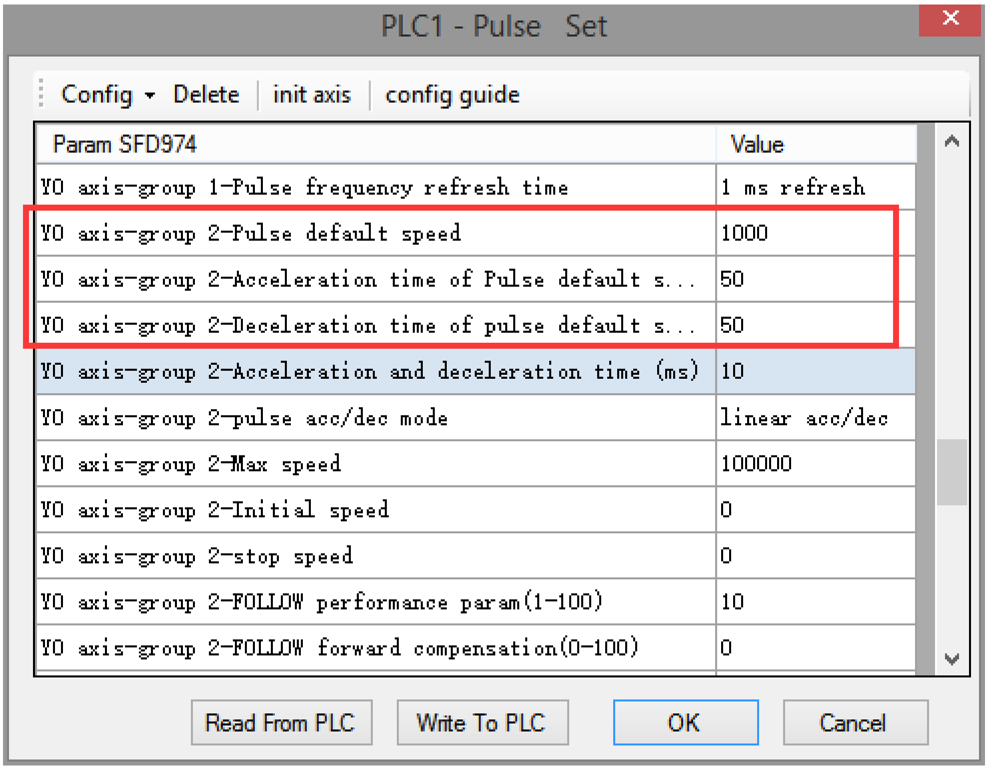




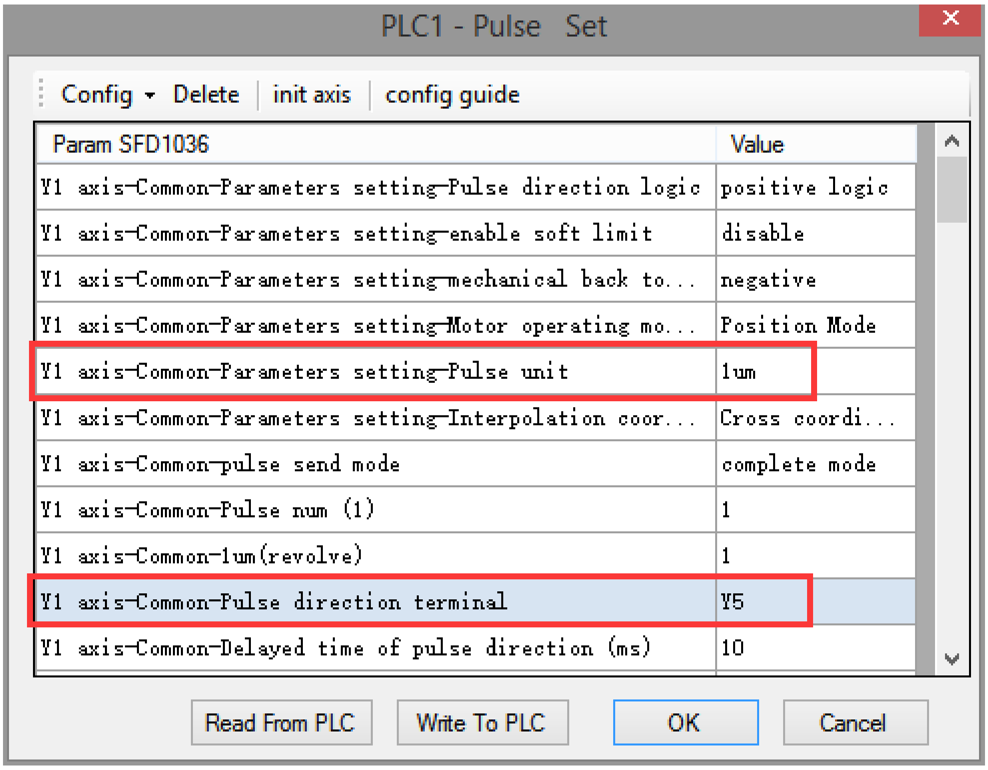
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

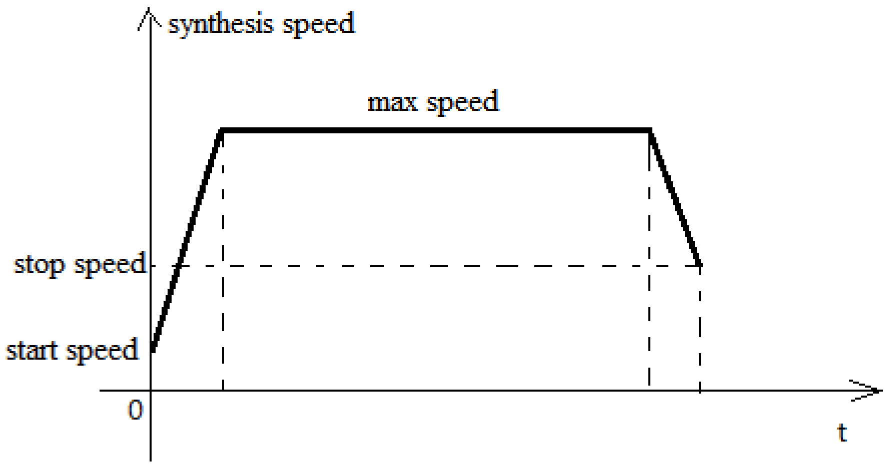
- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle center of axis 1, D30 specifies the circle center of axis 2, D40 specifies the start speed, D50 specifies the stop speed, D60 specifies the max speed.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 50Hz, D50 = 20, D60 = 2000, when M0 rises, execute CW command, accelerate from the starting position (1000,1000) at speed 50Hz to the maximum speed (2000Hz), and stop at the end speed of 20Hz when moving to the target position.

(1) When the end point is in absolute mode, the target position is (5000,2000), the center position is (3000,1500), and D20 = 2000, D30 = 500.

(2) When the end point is in the relative mode, the target position is (6000,3000), the center position is (3500,2000), and D20 = 2500, D30 = 1000.

- When the CW instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Note: In this mode, the starting speed (S4), the ending speed (S5) and the maximum speed (S6) are all expressed as the two-axis synthesis speed, as shown in the following figure:

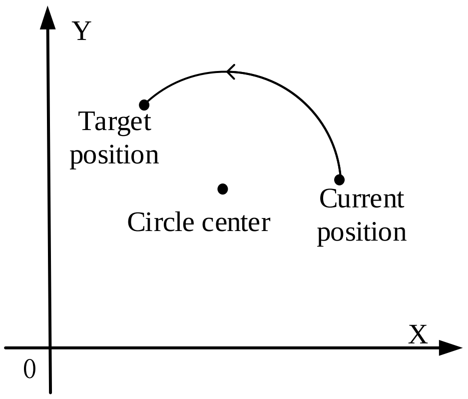


When there are multiple continuous linear/arc interpolation instructions and the speed between them needs to be constant and jump directly, the termination speed and maximum speed of the previous linear/arc interpolation can be set the same as the starting speed and maximum speed of the next segment.

When mode 3 is used, the starting and ending speed in the pulse parameter configuration tables of axis 1 and axis 2 are only effective for calculating the slope of pulse acceleration and deceleration.

2-4-5. Anticlockwise arc [CCW]

Anticlockwise arc interpolation CCW determines a section of arc mainly through the current position of arc, the target position and the counterclockwise coordinates of the center of the circle, as shown in the following figure:



With the above image, when you need to draw an entire circle, just set the target position to the current position. There are three modes of anticlockwise arc interpolation CCW, the usage of which is described below.

Mode 1: CCW　anticlockwise arc

1. Instruction overview

Anticlockwise arc interpolation instruction, operate according to the set default speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Anticlockwise arc interpolation [CCW]			
16-bit instruction	-	32-bit instruction	CCW
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the center position of axis 1 (always relative to the starting coordinates)	Double words, 32-bit
S3	Specify the center position of axis 2 (always relative to the starting coordinates)	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

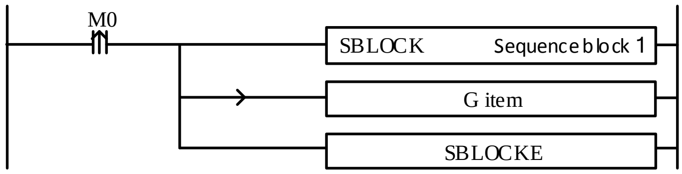
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

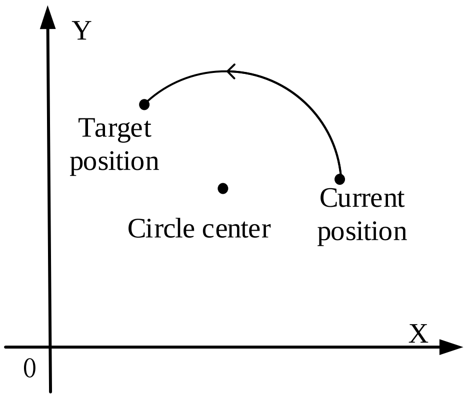
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Circle center position	The position of the center is determined by the position of the starting point and the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	Must set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



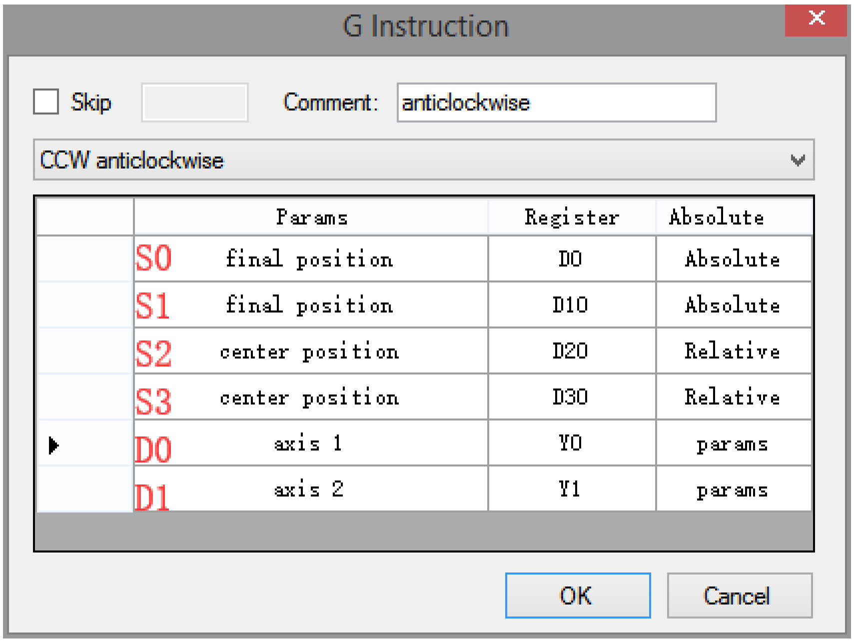
When the CCW instruction of arc interpolation (mode 1) is executed, the two axes will run at the highest synthesis speed. As shown in the following figure:



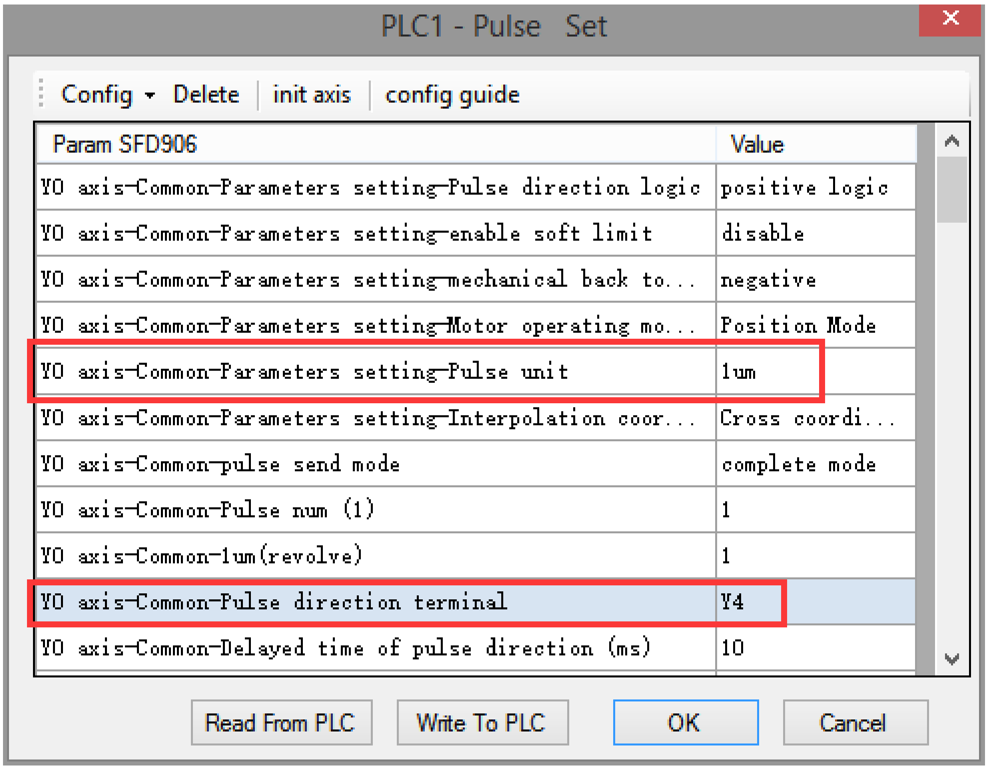
CCW clockwise arc interpolation

The parameter configuration is shown in the following figure:

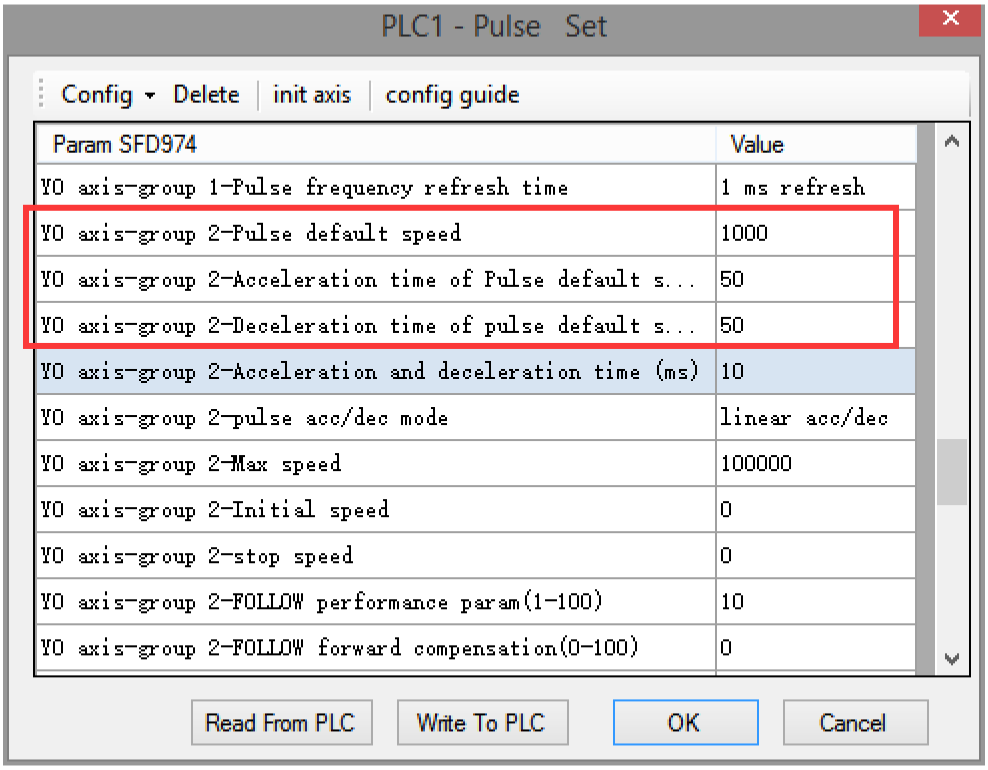
Double-click G item and pop up the configuration panel. Set it as follows:



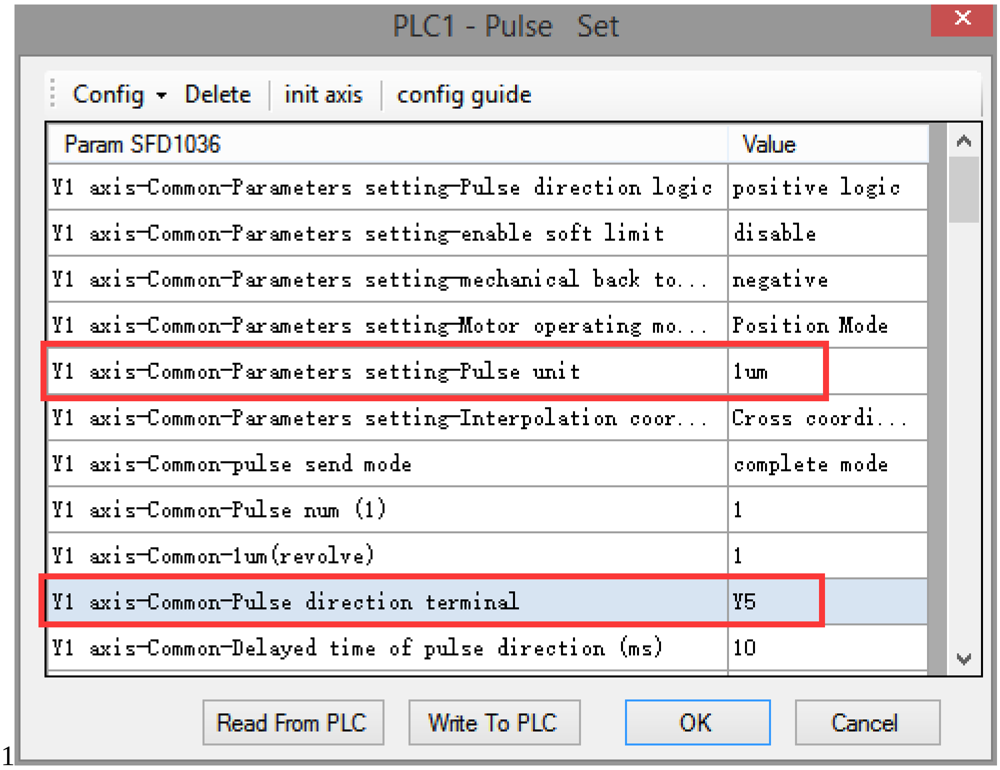
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



1

Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle center of axis 1, D30 specifies the circle center of axis 2.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute CCW command, move from the starting position (1000, 1000) to the target position at the default speed of 1000Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000), the center position is (3000,1500), and D20 = 2000, D30 = 500.

(2) When the end point is in the relative mode, the target position is (6000,3000), the center position is (3500,2000), and D20 = 2500, D30 = 1000.

- When the CCW instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 2: CCW anticlockwise VM

1. Instruction overview

Anticlockwise arc interpolation instruction, operate according to the set maximum synthetic speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Anticlockwise arc interpolation [CCW]			
16-bit instruction	-	32-bit instruction	CCW
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the center position of axis 1 (always relative to the starting coordinates)	Double words, 32-bit
S3	Specify the center position of axis 2 (always relative to the starting coordinates)	Double words, 32-bit
S4	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							
S4	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

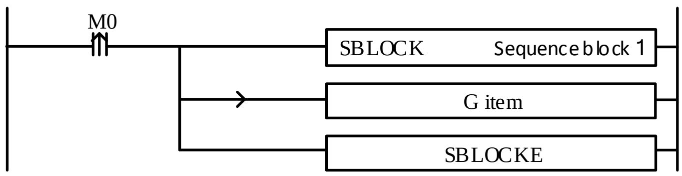
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

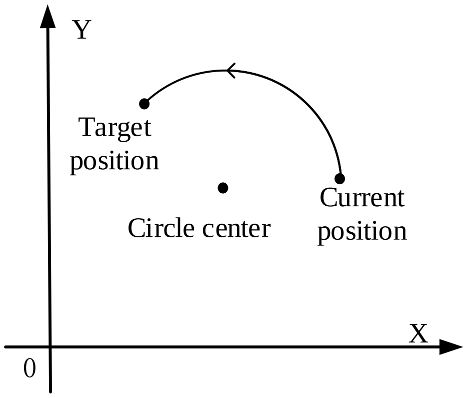
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Circle center position	The position of the center is determined by the position of the starting point and the end point	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



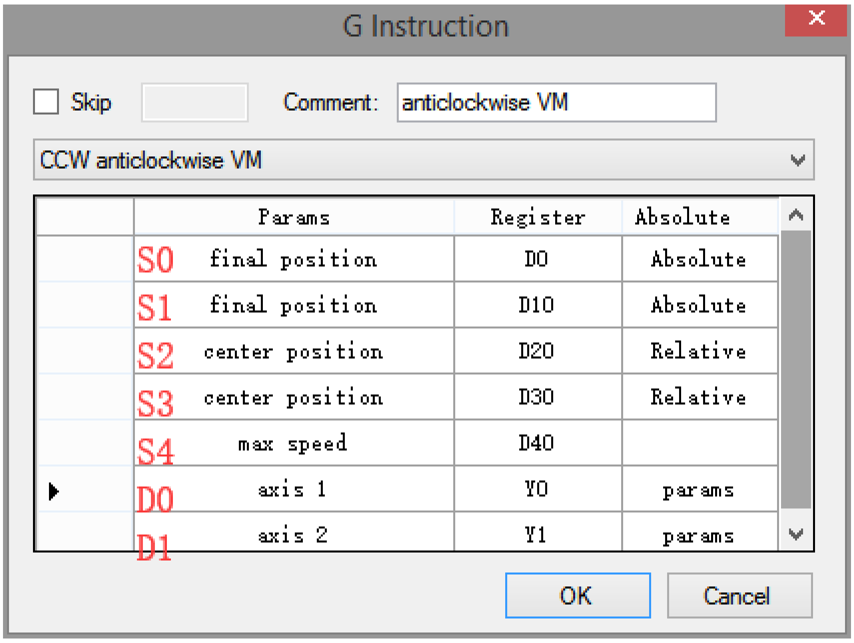
When the CCW instruction of arc interpolation (mode 2) is executed, the two axes will run at the set max synthesis speed. As shown in the following figure:



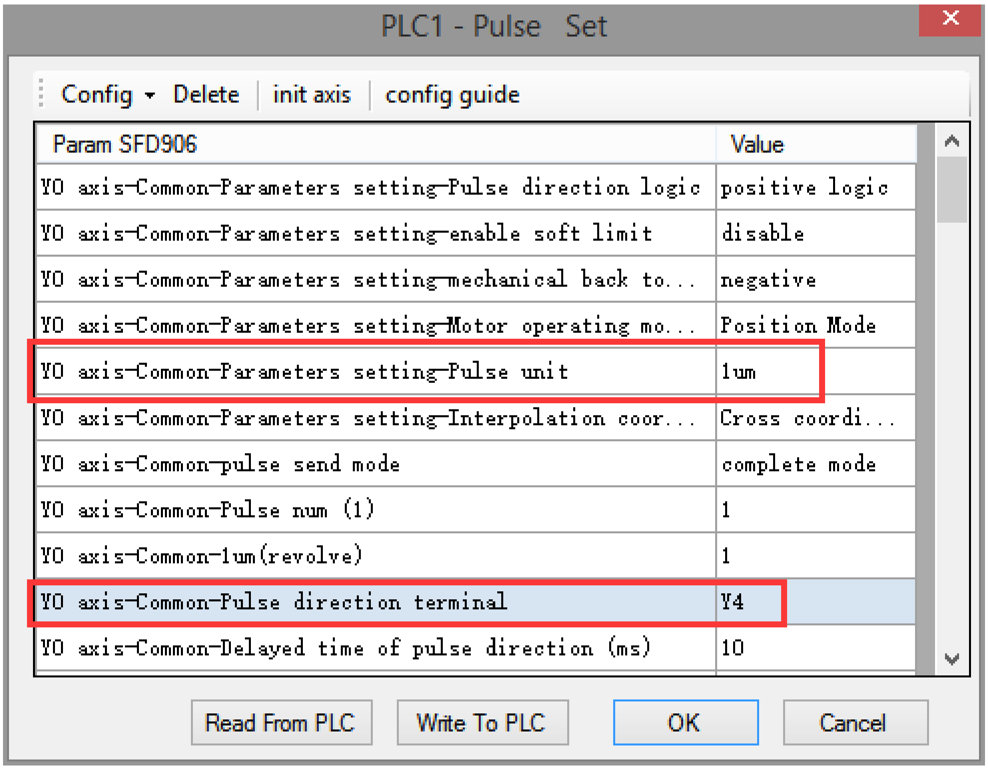
CCW clockwise arc interpolation

The parameter configuration is shown in the following figure:

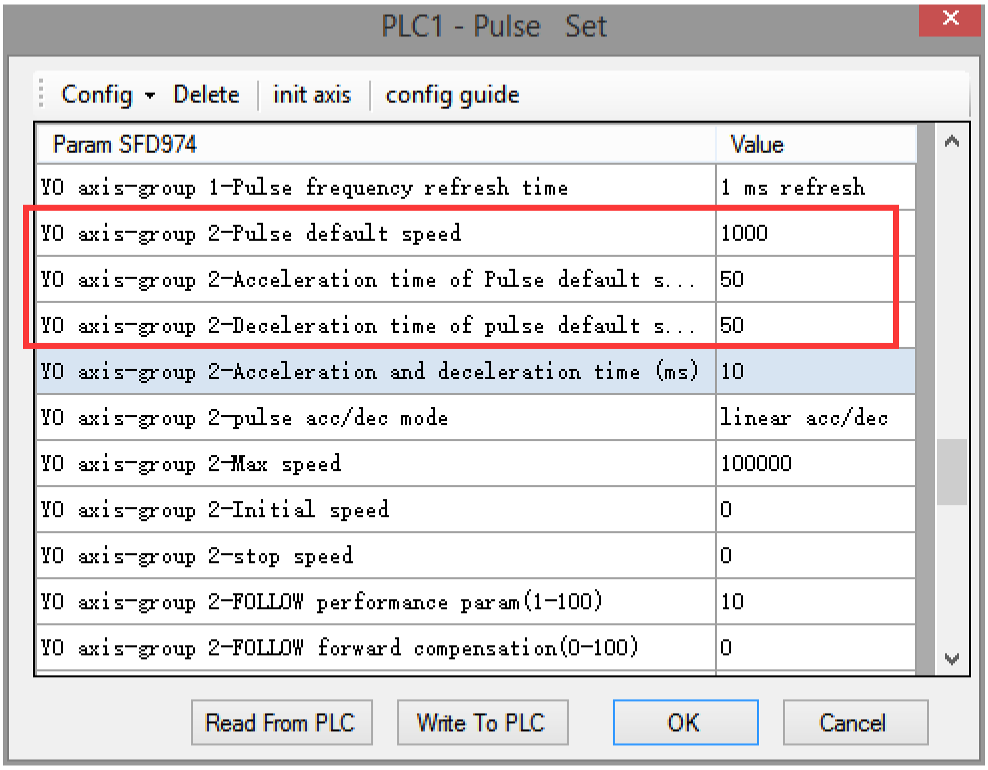
Double-click G item and pop up the configuration panel. Set it as follows:



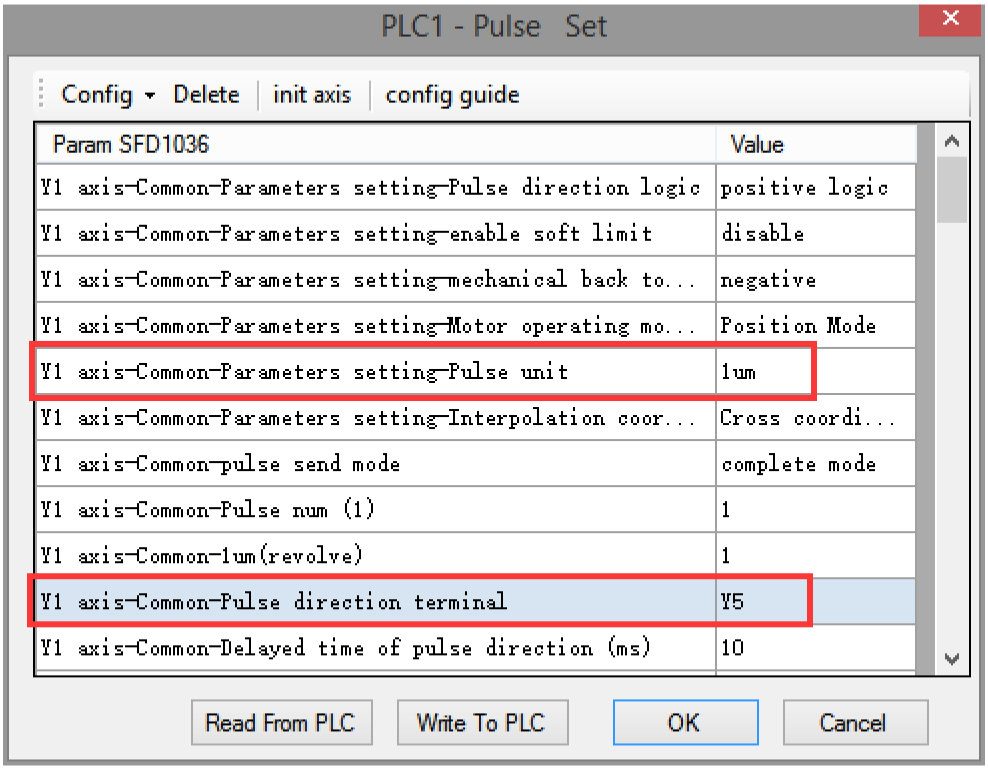
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle center of axis 1, D30 specifies the circle center of axis 2, D40 specifies the max speed.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 500Hz, when M0 rises, execute CCW command, move from the starting position (1000, 1000) to the target position at the max speed of 500Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000), the center position is (3000,1500), and D20 = 2000, D30 = 500.

(2) When the end point is in the relative mode, the target position is (6000,3000), the center position is (3500,2000), and D20 = 2500, D30 = 1000.

- When the CCW instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 3: CCW anticlockwise VBEM

1. Instruction overview

Anticlockwise arc interpolation instruction, operate according to the set maximum synthetic speed, start speed and stop speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Anticlockwise arc interpolation [CCW]			
16-bit instruction	-	32-bit instruction	CCW
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the center position of axis 1 (always relative to the starting coordinates)	Double words, 32-bit
S3	Specify the center position of axis 2 (always relative to the starting coordinates)	Double words, 32-bit
S4	Specify the starting speed at the starting point of the two axes	Double words, 32-bit
S5	Specify the stop speed at the end point of the two axes	Double words, 32-bit
S6	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0~S6	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

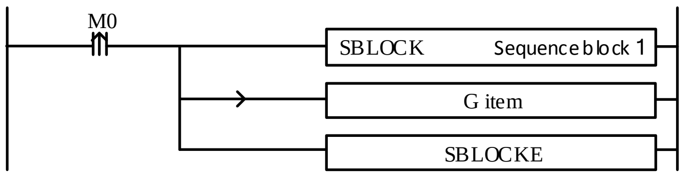
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

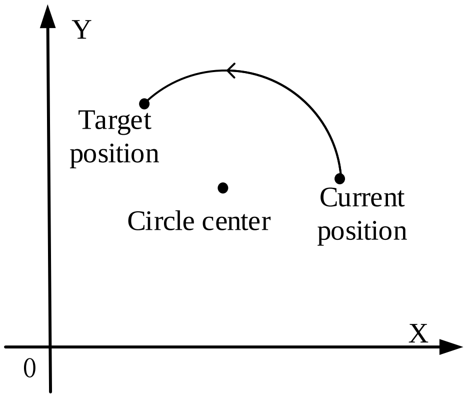
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Circle center position	The position of the center is determined by the position of the starting point and the end point	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Start speed	The start speed from the starting point	Must set
Stop speed	The stop speed at the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



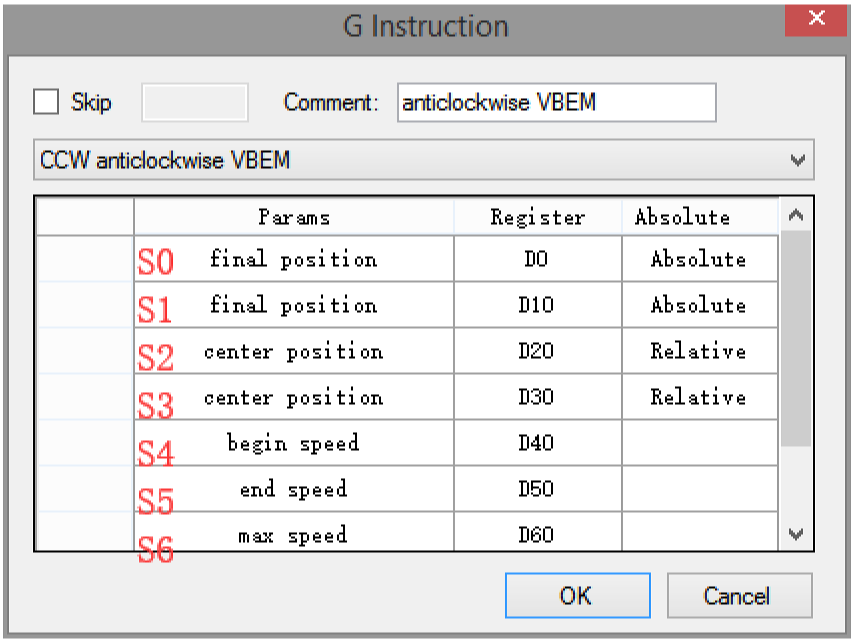
When the CCW instruction of arc interpolation (mode 3) is executed, the two axes will run at the set max synthesis speed, start speed and stop speed. As shown in the following figure:

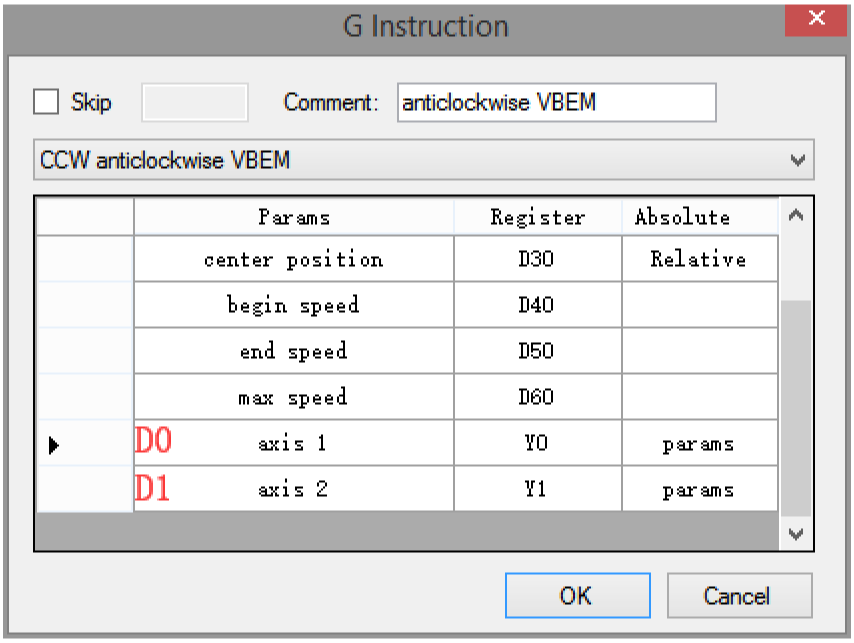


CCW clockwise arc interpolation

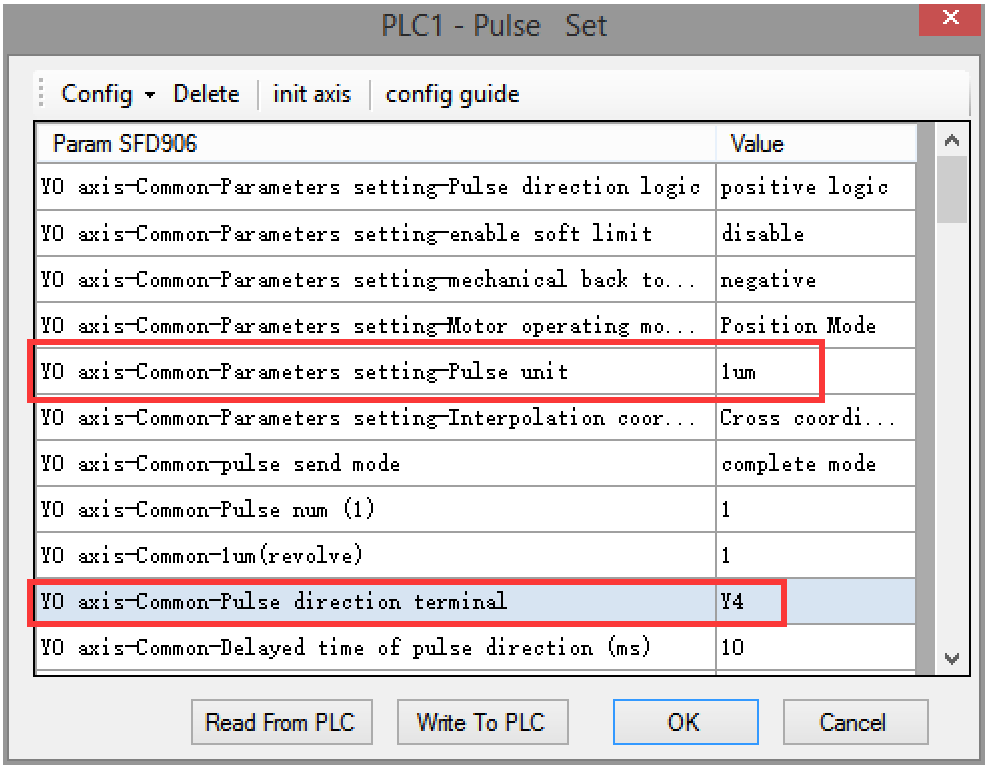
The parameter configuration is shown in the following figure:

Double-click G item and pop up the configuration panel. Set it as follows:

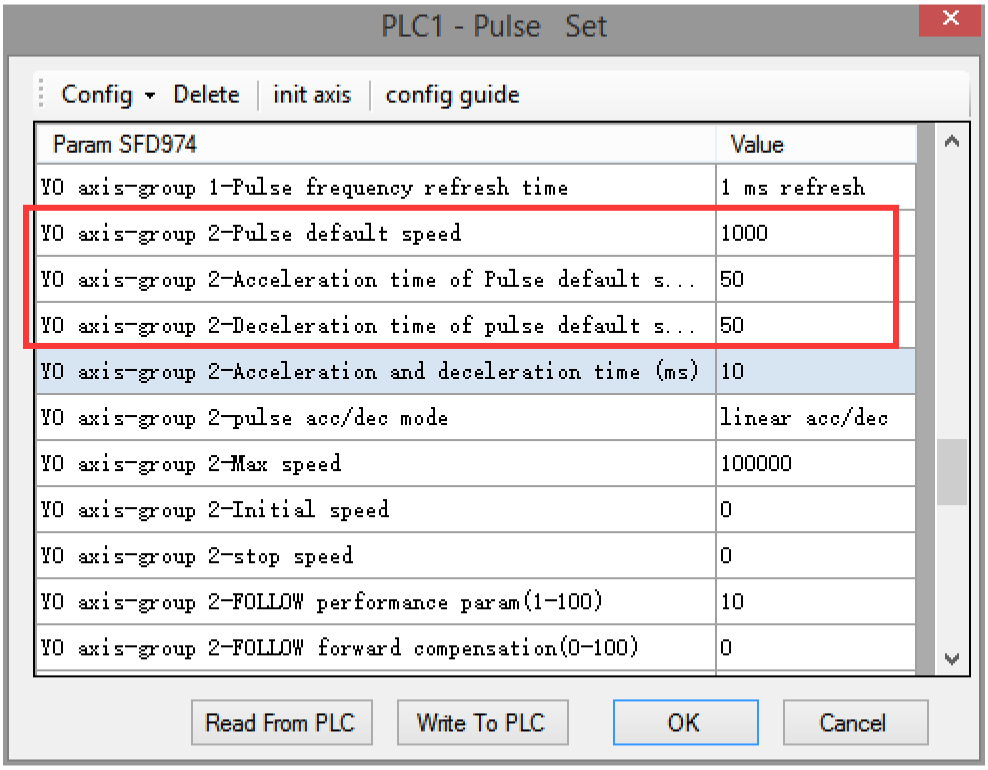




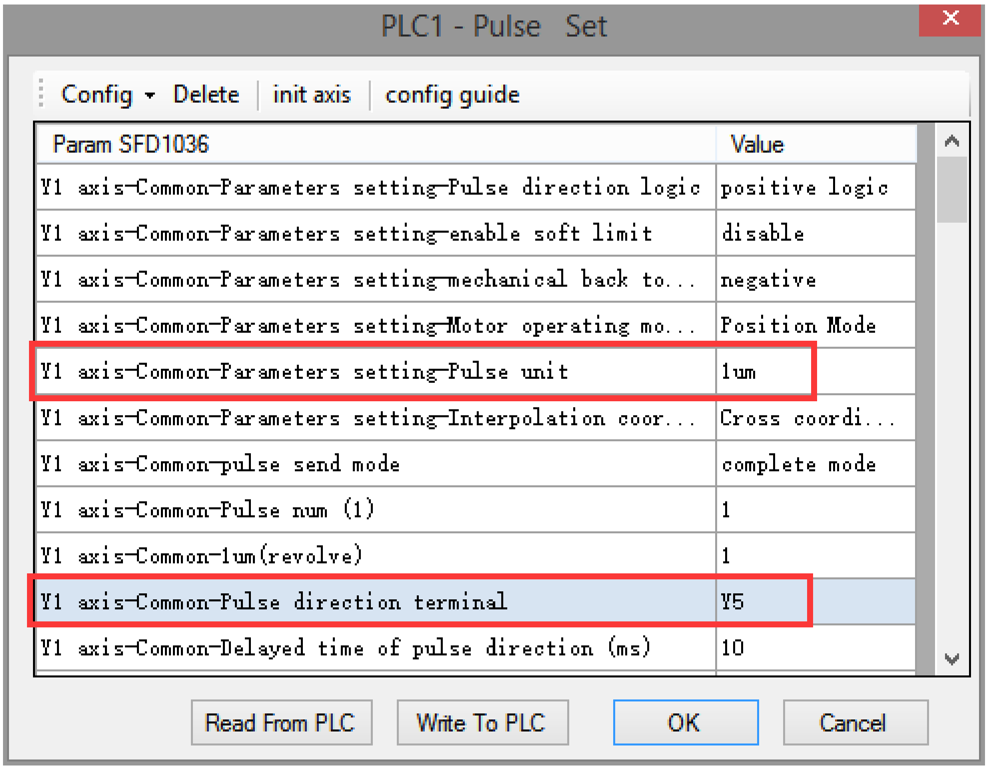
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle center of axis 1, D30 specifies the circle center of axis 2, D40 specifies the start speed, D50 specifies the stop speed, D60 specifies the max speed.

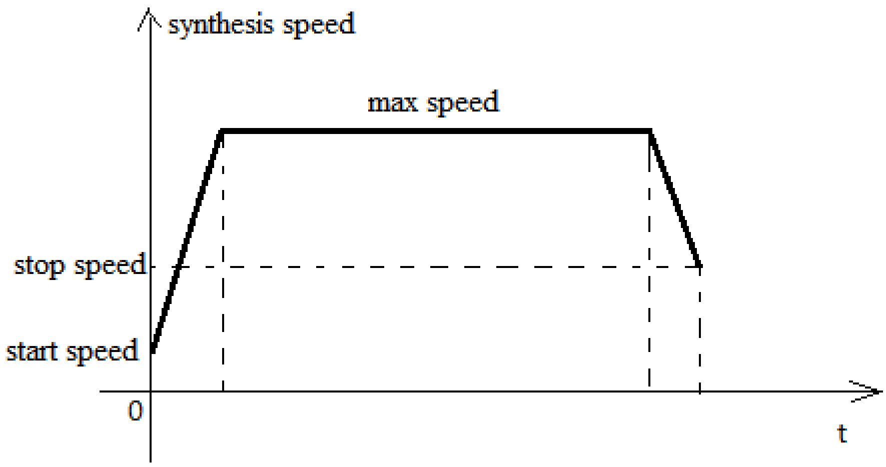
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 50Hz, D50 = 20, D60 = 2000, when M0 rises, execute CCW command, accelerate from the starting position (1000,1000) at speed 50Hz to the maximum speed (2000Hz), and stop at the end speed of 20Hz when moving to the target position.

(1) When the end point is in absolute mode, the target position is (5000,2000), the center position is (3000,1500), and D20 = 2000, D30 = 500.

(2) When the end point is in the relative mode, the target position is (6000,3000), the center position is (3500,2000), and D20 = 2500, D30 = 1000.

- When the CCW instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Note: In this mode, the starting speed (S4), the ending speed (S5) and the maximum speed (S6) are all expressed as the two-axis synthesis speed, as shown in the following figure:

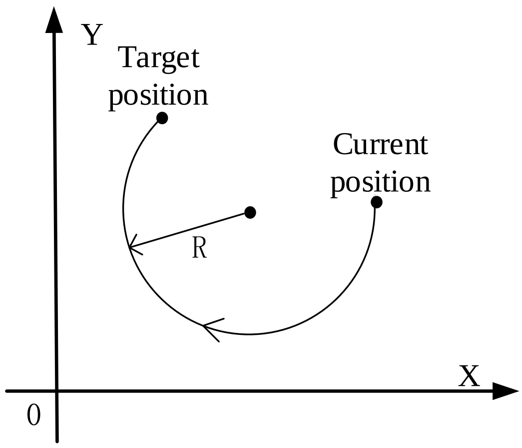


When there are multiple continuous linear/arc interpolation instructions and the speed between them needs to be constant and jump directly, the termination speed and maximum speed of the previous linear/arc interpolation can be set the same as the starting speed and maximum speed of the next segment.

When mode 3 is used, the starting and ending speed in the pulse parameter configuration tables of axis 1 and axis 2 are only effective for calculating the slope of pulse acceleration and deceleration.

2-4-6. Clockwise arc [CW_R]

Clockwise arc interpolation CW_R is mainly based on the current position of the arc, the target position and the length of the radius of the circle, clockwise to determine a section of the arc, as shown in the following figure:



With the above figure, when the target position is set at the same position as the current one, the next circle can not be determined, so this mode can not draw a whole circle. There are three modes of CW_R. The usage of CW_R is described below.

Mode 1: CW_R clockwise arc

1. Instruction overview

Clockwise arc interpolation instruction, operate according to the set default speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Clockwise arc interpolation [CW_R]			
16-bit instruction	-	32-bit instruction	CW_R
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the radius of the arc	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

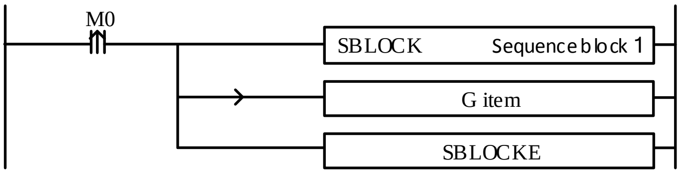
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

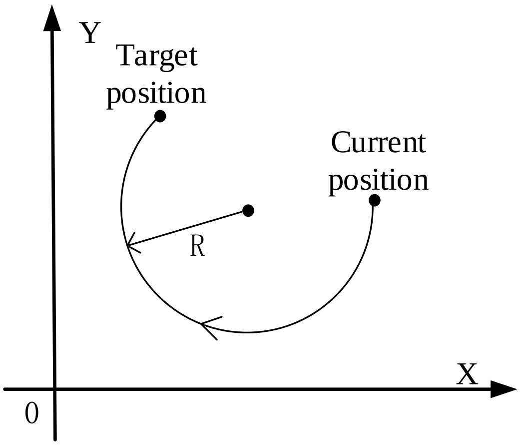
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Radius	The path of an arc varies with its radius.	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	Must set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



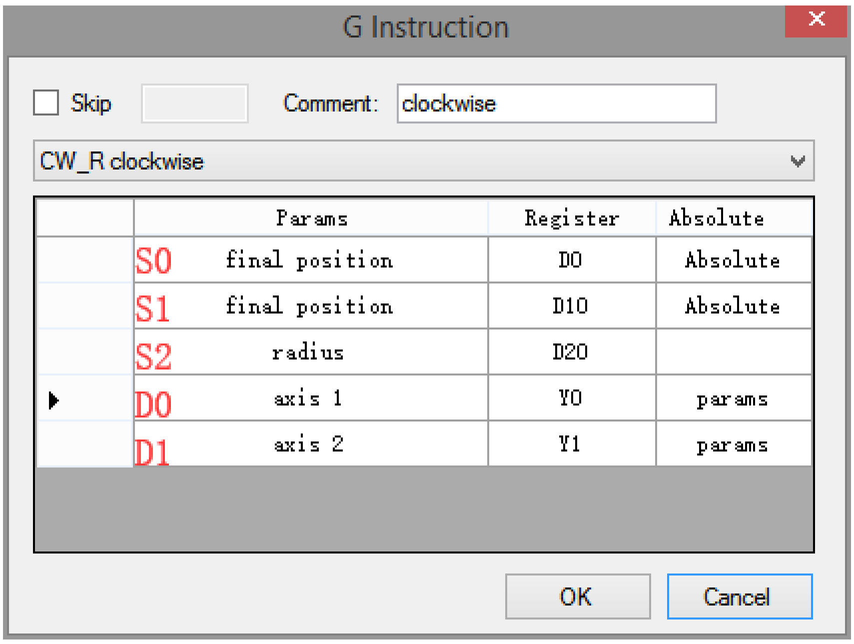
When the CW_R instruction of arc interpolation (mode 1) is executed, the two axes will run at the highest synthesis speed. As shown in the following figure:



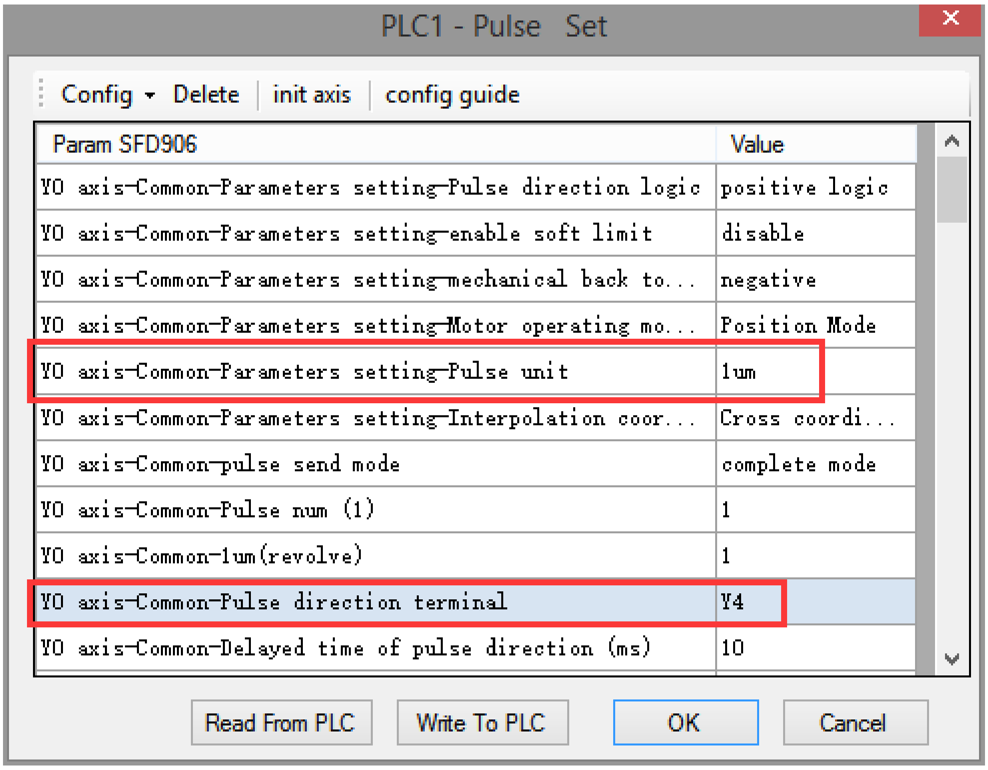
CW_R clockwise arc interpolation

The parameter configuration is shown in the following figure:

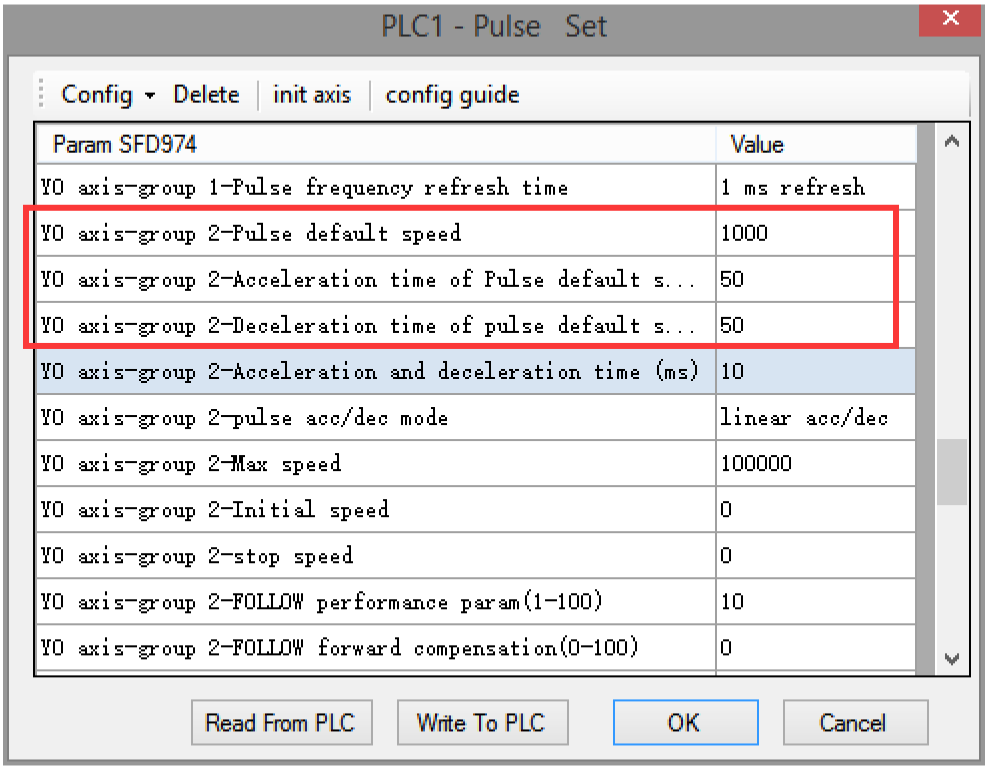
Double-click G item and pop up the configuration panel. Set it as follows:



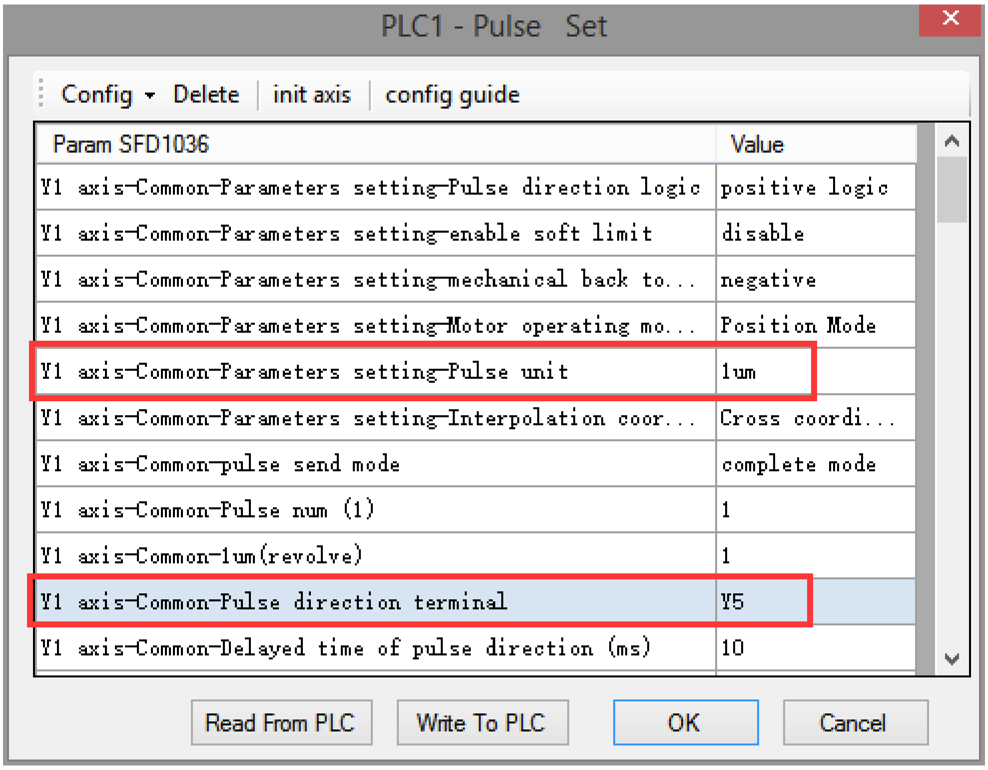
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle radius. The path of an arc varies with its radius.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute CW_R command, move from the starting position (1000, 1000) to the target position at the default speed of 1000Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the CW_R instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.
- When the radius is positive, the arc is inferior; when the radius is negative, it is major arc.

Mode 2: CW_R clockwise arc VM

1. Instruction overview

Clockwise arc interpolation instruction, operate according to the set maximum synthetic speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Clockwise arc interpolation [CW_R]			
16-bit instruction	-	32-bit instruction	CW_R
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the radius of the arc	Double words, 32-bit
S3	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

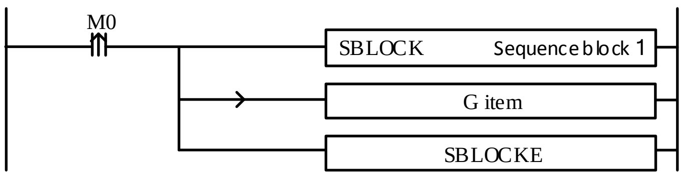
4. Parameter setting

Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set

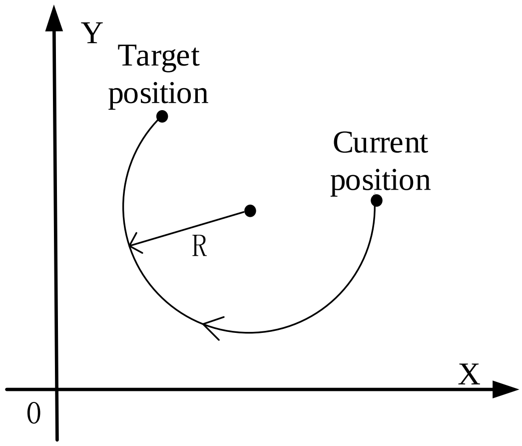
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Radius	The path of an arc varies with its radius.	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

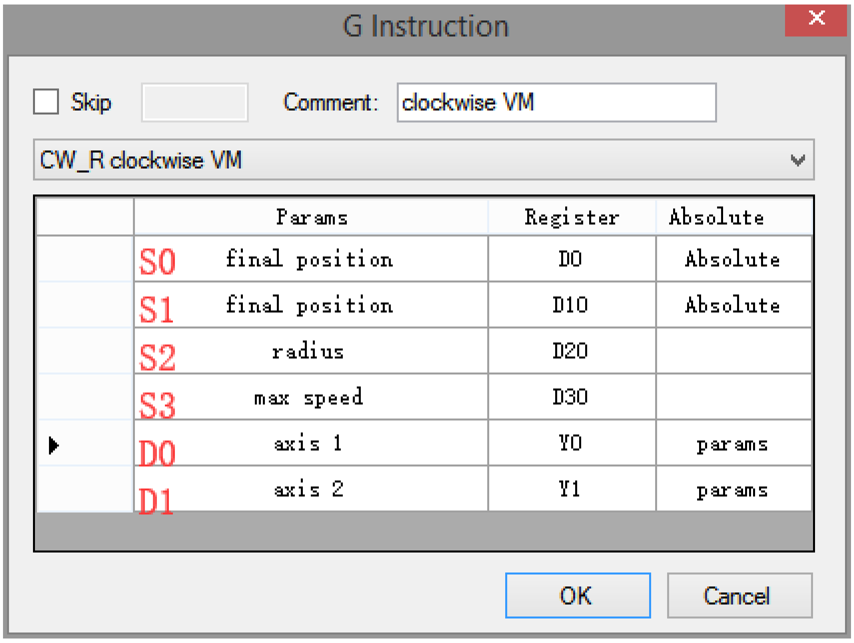


When the CW_R instruction of arc interpolation (mode 2) is executed, the two axes will run at the set max synthesis speed. As shown in the following figure:

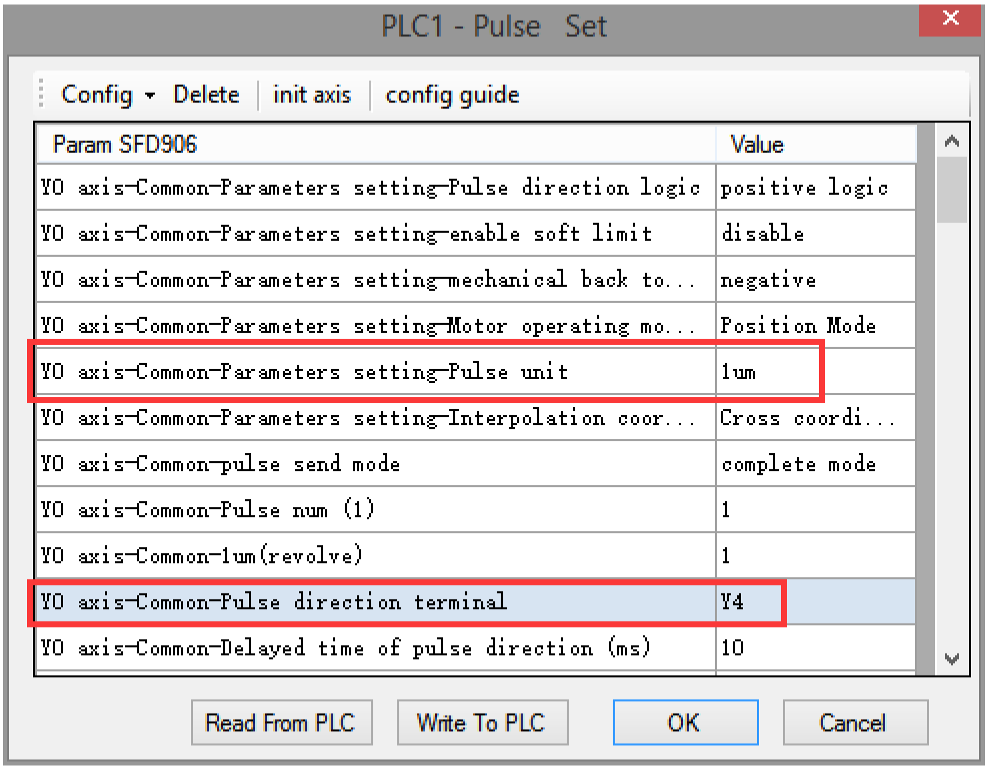


CW_R clockwise arc interpolation

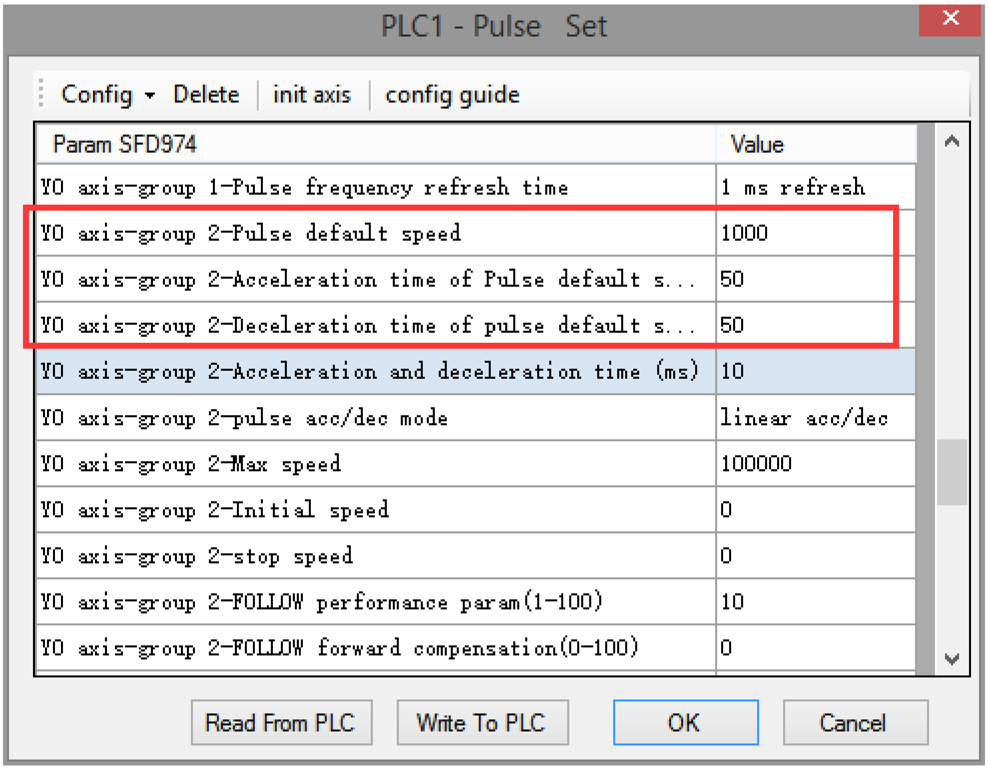
The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:



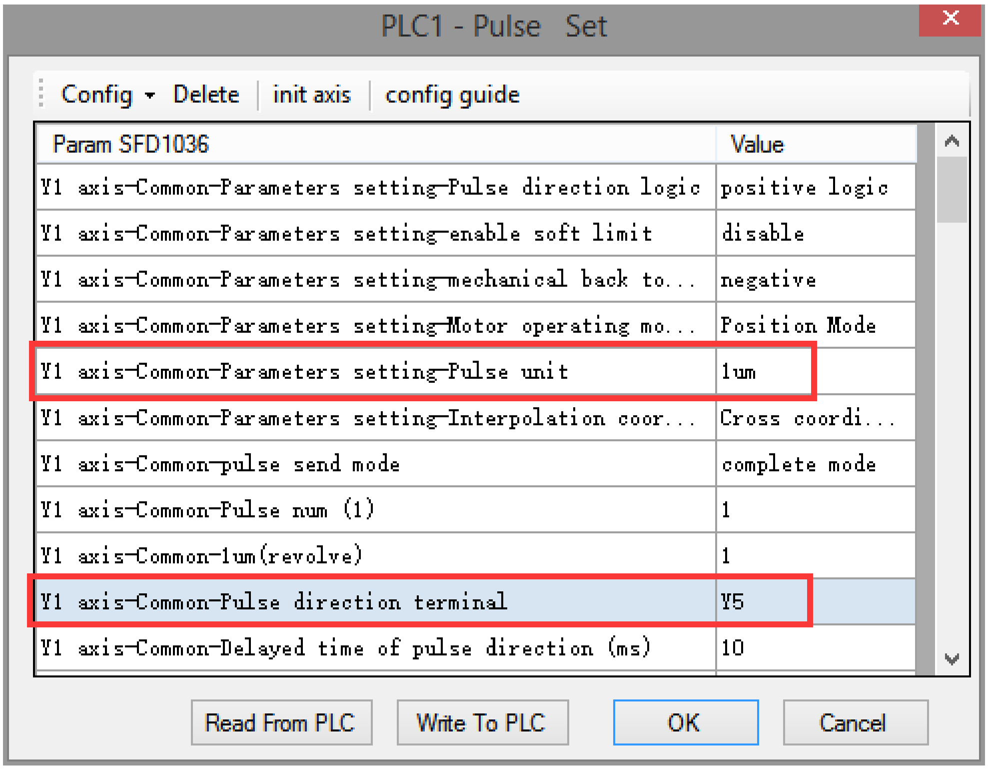
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the radius (the radius is different and the path is different), D30 specifies the max speed.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 500Hz, when M0 rises, execute CW_R command, move from the starting position (1000, 1000) to the target position at the max speed of 500Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the CW_R instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.
- When the radius is positive, the arc is inferior; when the radius is negative, it is major arc.

Mode 3: CW_R clockwise arc VBEM

1. Instruction overview

Clockwise arc interpolation instruction, operate according to the set maximum synthetic speed, start speed and stop speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Clockwise arc interpolation [CW_R]			
16-bit instruction	-	32-bit instruction	CW_R
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the radius of the arc	Double words, 32-bit
S3	Specify the starting speed at the starting point of the two axes	Double words, 32-bit
S4	Specify the stop speed at the end point of the two axes	Double words, 32-bit
S5	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0~S5	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

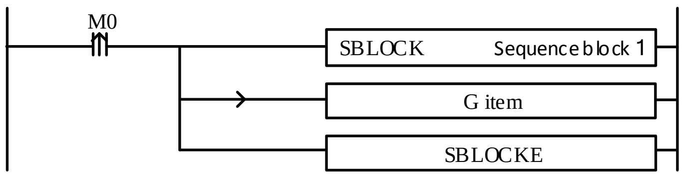
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

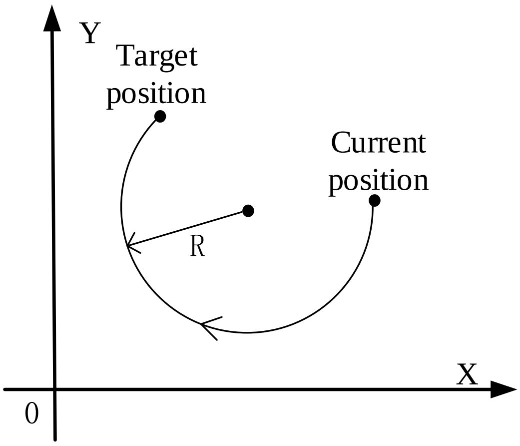
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
radius	The radius is different and the path is different	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Start speed	The start speed from the starting point	Must set
Stop speed	The stop speed at the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

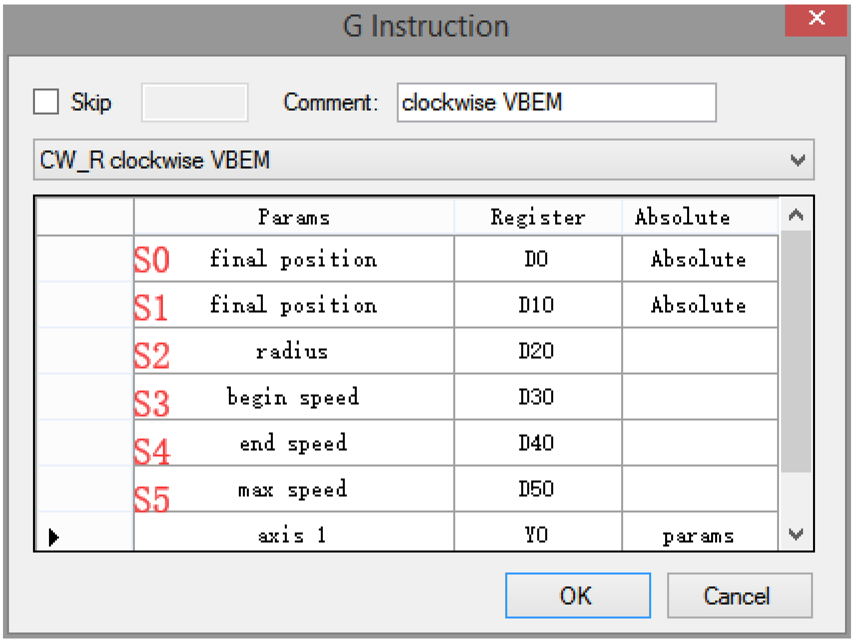


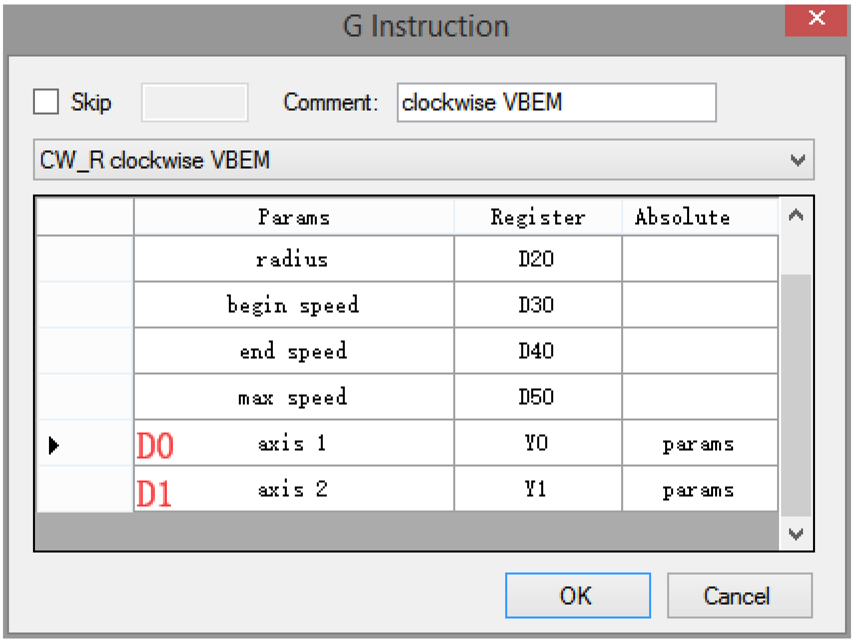
When the CW_R instruction of arc interpolation (mode 3) is executed, the two axes will run at the set max synthesis speed, start speed and stop speed. As shown in the following figure:



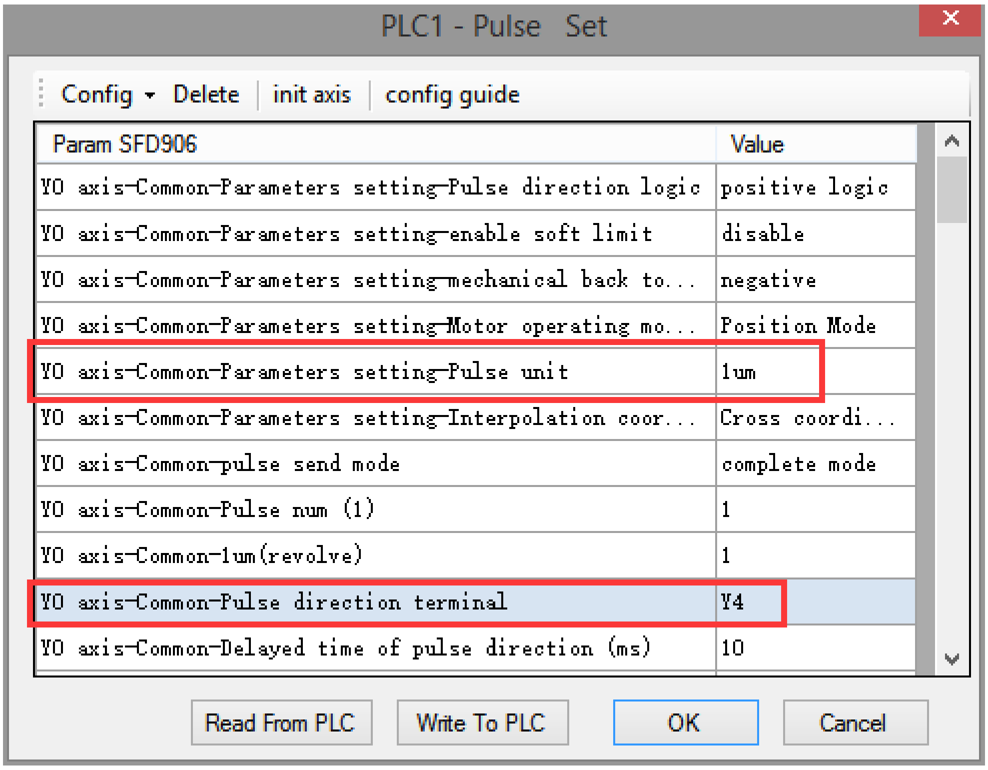
CW_R clockwise arc interpolation

The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:

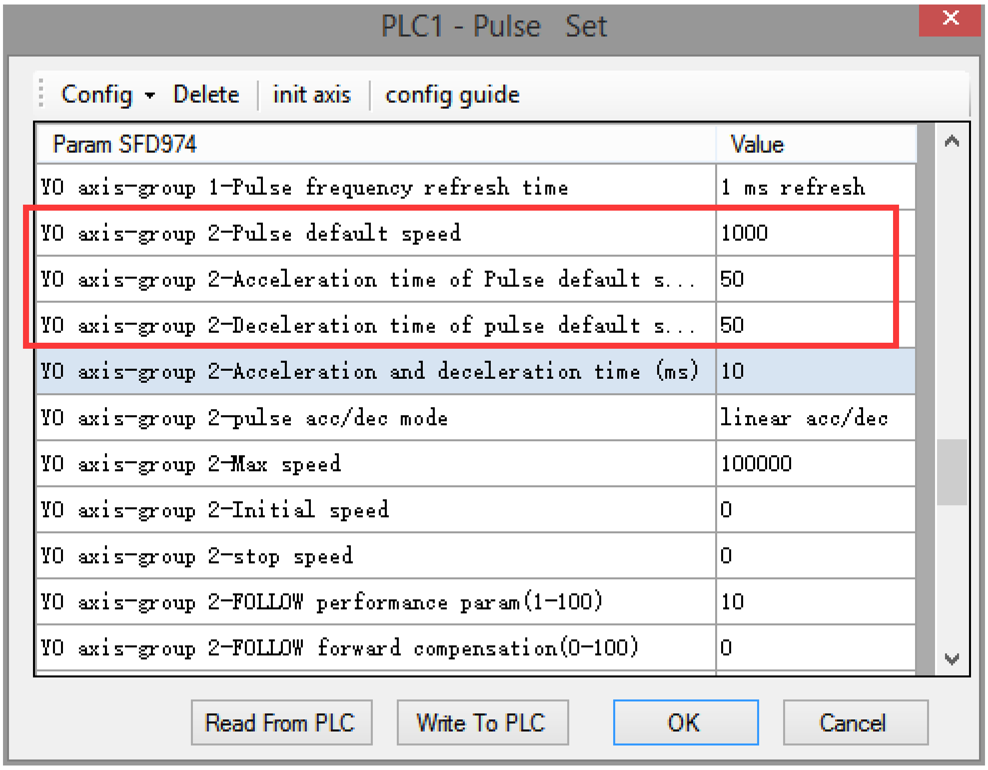




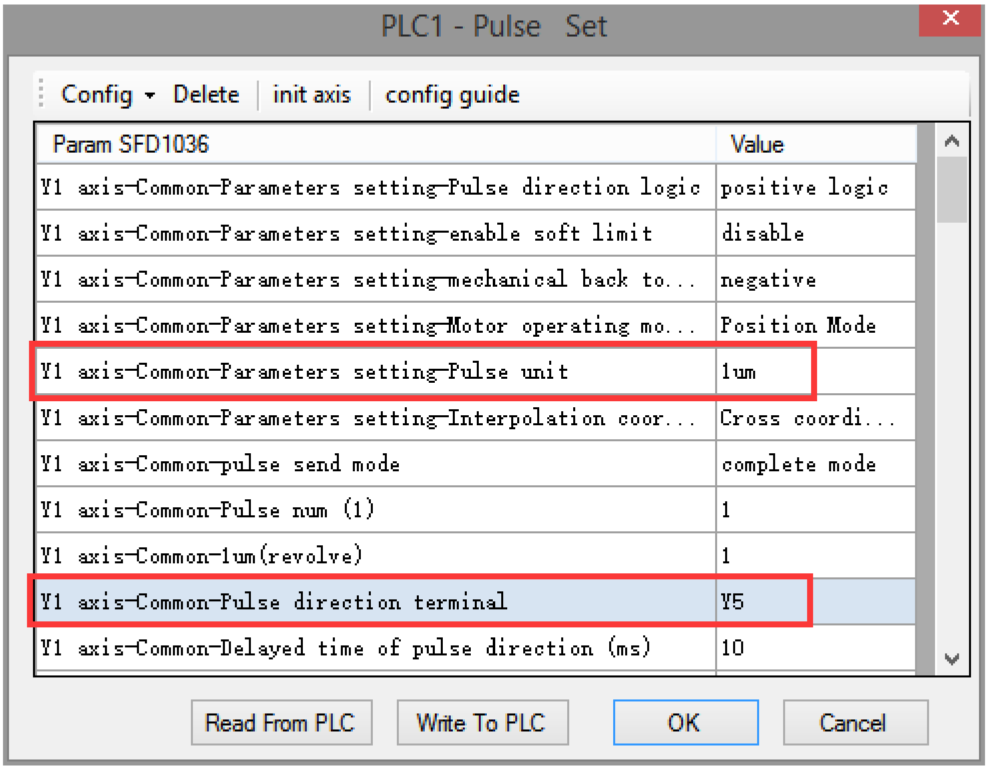
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the radius, D30 specifies the start speed, D40 specifies the stop speed, D50 specifies the max speed.

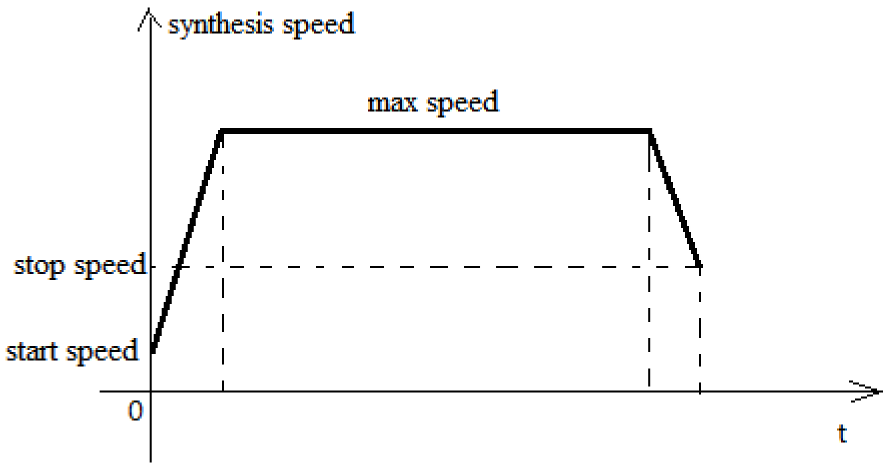
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 50Hz, D50 = 20, D60 = 2000, when M0 rises, execute CW_R command, accelerate from the starting position (1000,1000) at speed 50Hz to the maximum speed (2000Hz), and stop at the end speed of 20Hz when moving to the target position.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the CW_R instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.
- When the radius is positive, the arc is inferior; when the radius is negative, it is major arc.

Note: In this mode, the starting speed (S3), the ending speed (S4) and the maximum speed (S5) are all expressed as the two-axis synthesis speed, as shown in the following figure:

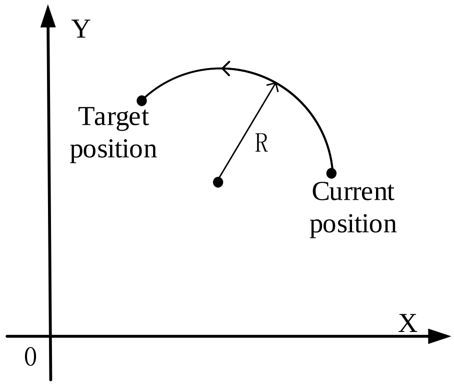


When there are multiple continuous linear/arc interpolation instructions and the speed between them needs to be constant and jump directly, the termination speed and maximum speed of the previous linear/arc interpolation can be set the same as the starting speed and maximum speed of the next segment.

When mode 3 is used, the starting and ending speed in the pulse parameter configuration tables of axis 1 and axis 2 are only effective for calculating the slope of pulse acceleration and deceleration.

2-4-7. Anticlockwise arc [CCW_R]

Anticlockwise arc interpolation CCW_R is mainly based on the current position of the arc, the target position and the length of the radius of the circle, clockwise to determine a section of the arc, as shown in the following figure:



With the above figure, when the target position is set at the same position as the current one, the next circle can not be determined, so this mode can not draw a whole circle. There are three modes of CCW_R. The usage of CCW_R is described below.

Mode 1: CCW_R anticlockwise arc

1. Instruction overview

Anticlockwise arc interpolation instruction, operate according to the set default speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Anticlockwise arc interpolation [CCW_R]			
16-bit instruction	-	32-bit instruction	CCW_R
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the radius of the arc	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

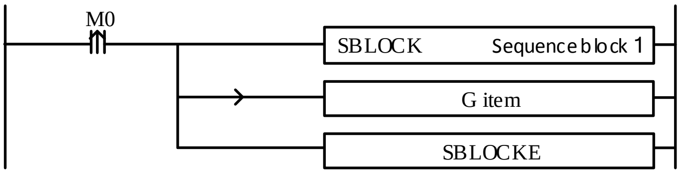
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

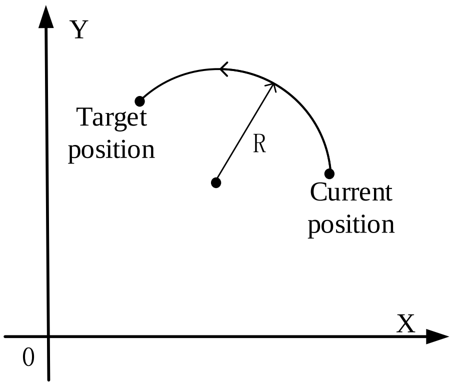
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Radius	The path of an arc varies with its radius.	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	Must set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



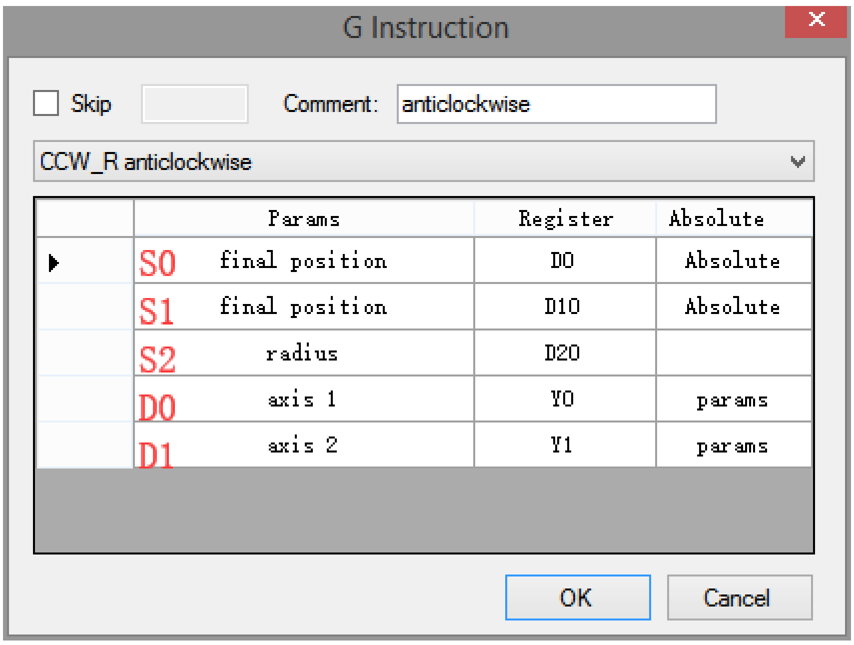
When the CCW_R instruction of arc interpolation (mode 1) is executed, the two axes will run at the highest synthesis speed. As shown in the following figure:



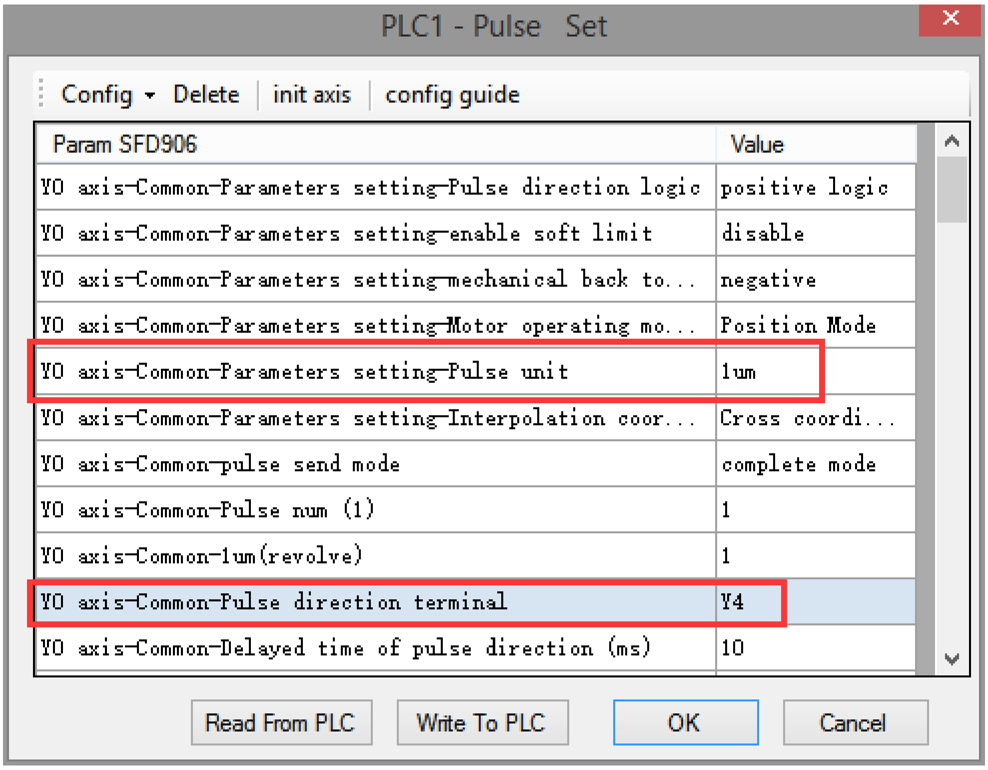
CCW_R anticlockwise arc interpolation

The parameter configuration is shown in the following figure:

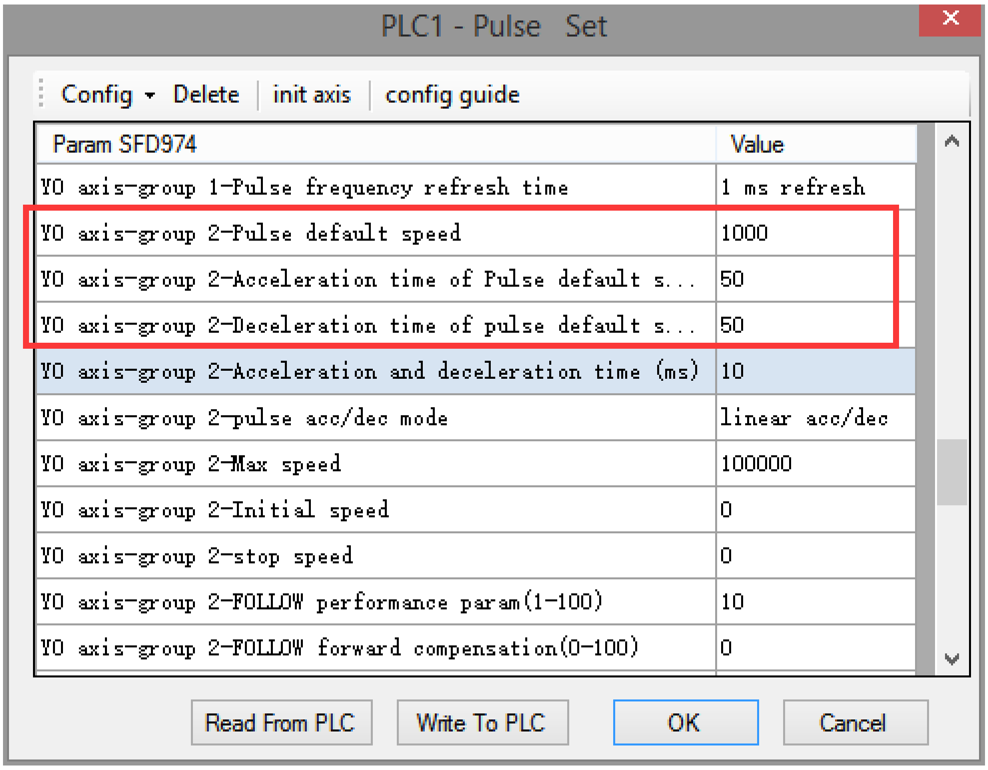
Double-click G item and pop up the configuration panel. Set it as follows:



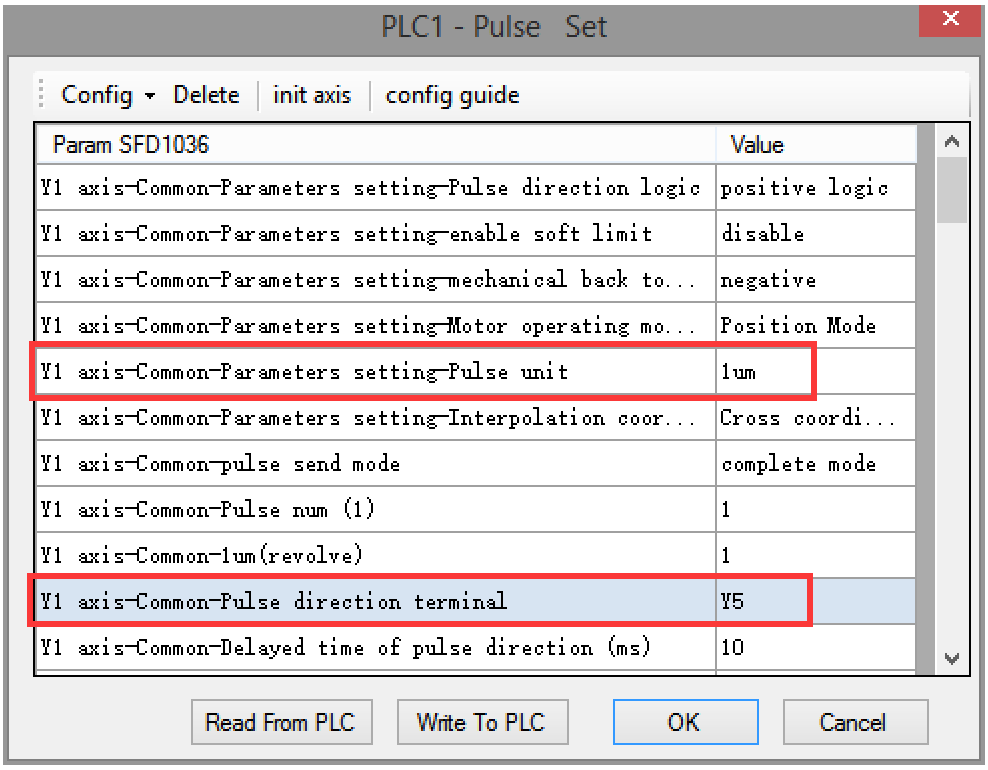
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the circle radius.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute CCW_R command, move from the starting position (1000, 1000) to the target position at the default speed of 1000Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the CCW_R instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.
- When the radius is positive, the arc is inferior; when the radius is negative, it is major arc.

Mode 2: CCW_R anticlockwise arc VM

1. Instruction overview

Anticlockwise arc interpolation instruction, operate according to the set maximum synthetic speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Anticlockwise arc interpolation [CCW_R]			
16-bit instruction	-	32-bit instruction	CCW_R
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the radius of the arc	Double words, 32-bit
S3	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

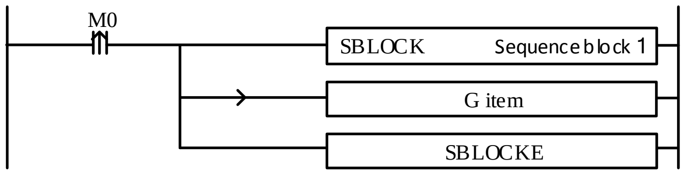
4. Parameter setting

Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set

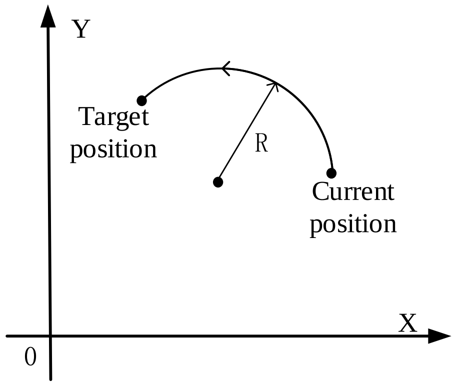
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Radius	The path of an arc varies with its radius.	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



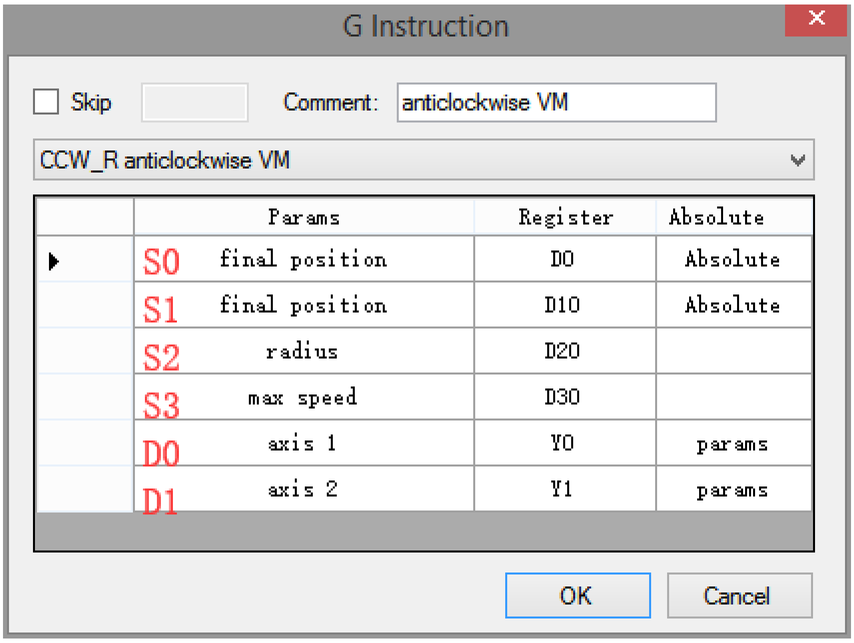
When the CCW_R instruction of arc interpolation (mode 2) is executed, the two axes will run at the set max synthesis speed. As shown in the following figure:



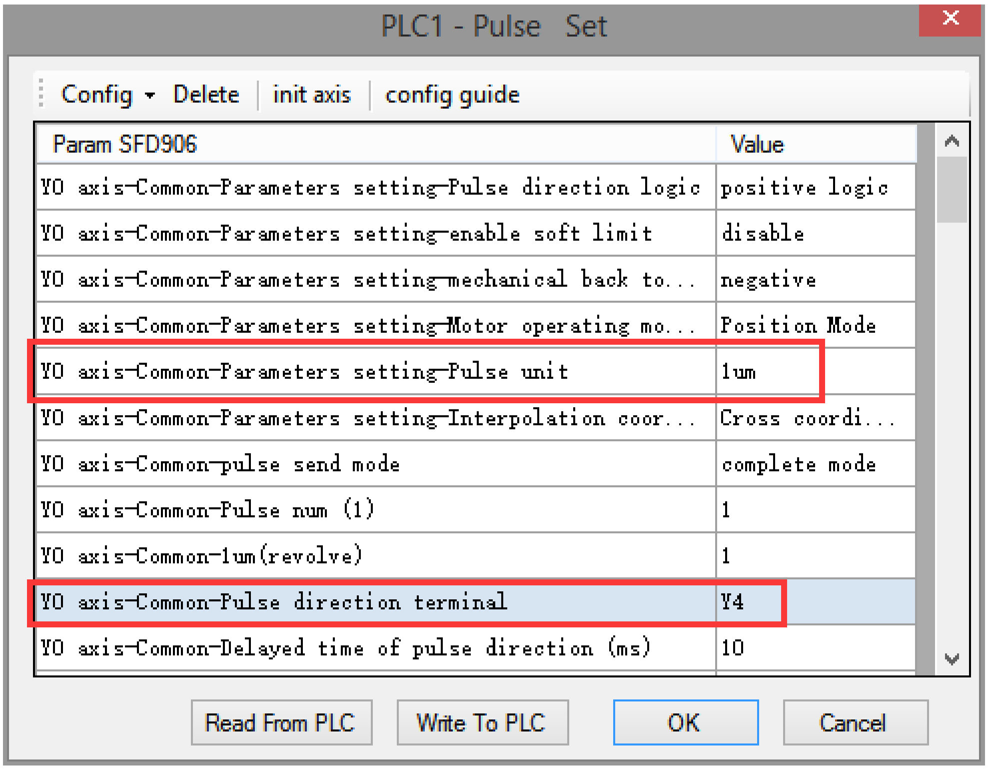
CCW_R anticlockwise arc interpolation

The parameter configuration is shown in the following figure:

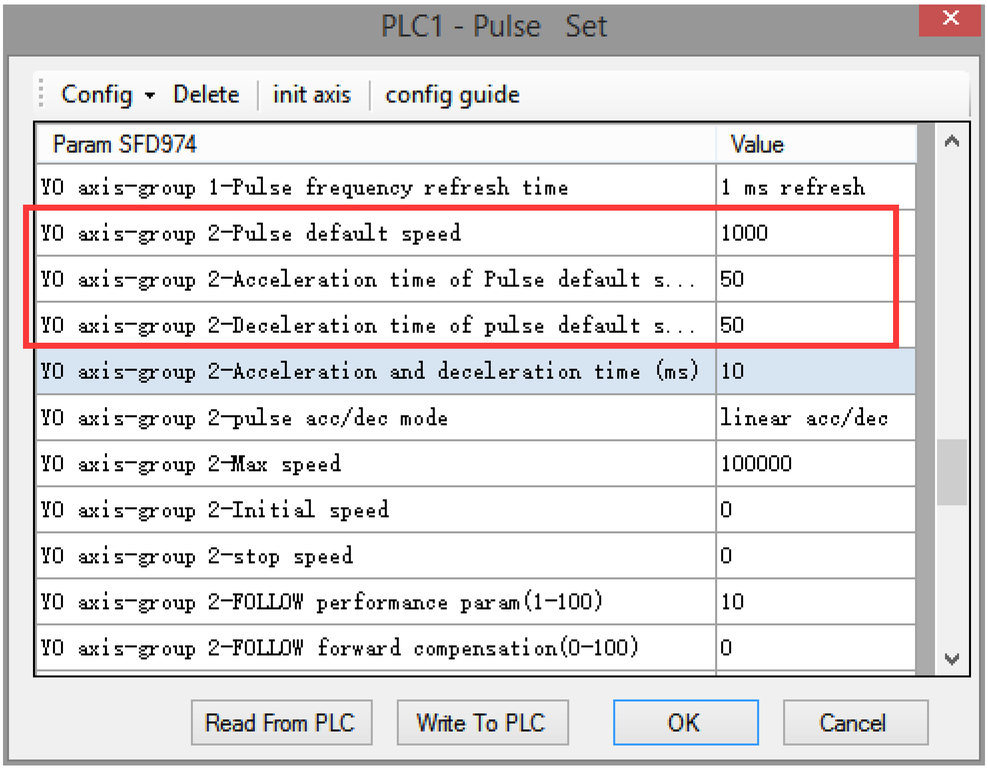
Double-click G item and pop up the configuration panel. Set it as follows:



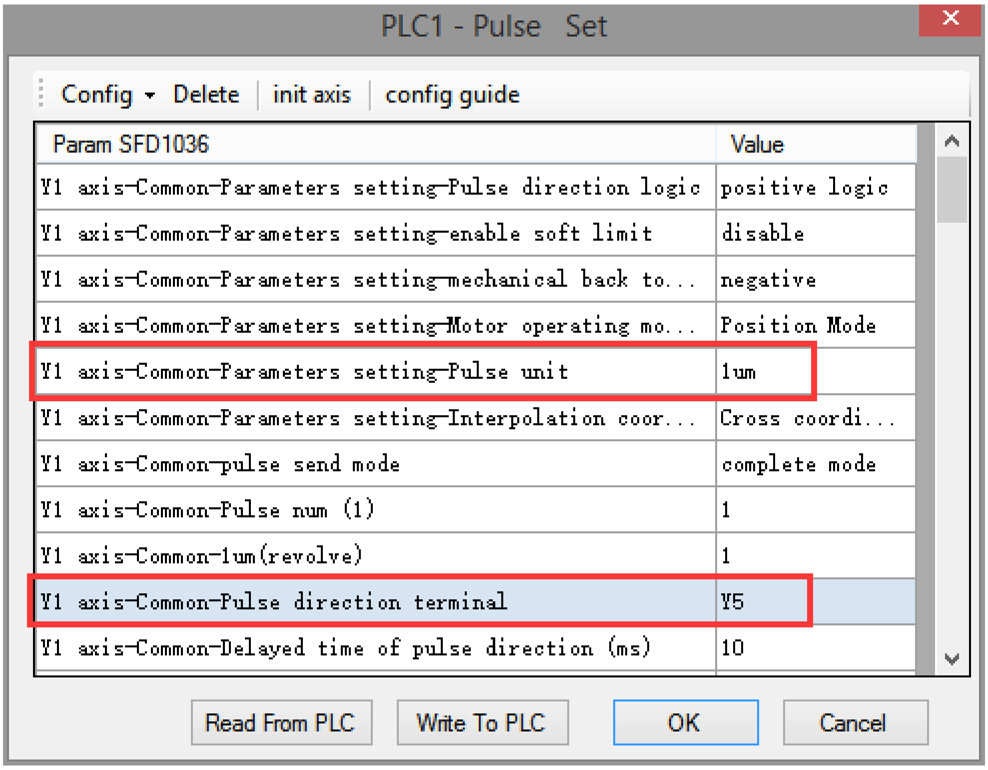
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the radius (the radius is different and the path is different), D30 specifies the max speed.

- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D30 = 500Hz, when M0 rises, execute CCW_R command, move from the starting position (1000, 1000) to the target position at the max speed of 500Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the CCW_R instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.
- When the radius is positive, the arc is inferior; when the radius is negative, it is major arc.

Mode 3: CCW_R anticlockwise arc VBEM

1. Instruction overview

Anticlockwise arc interpolation instruction, operate according to the set maximum synthetic speed, start speed and stop speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Anticlockwise arc interpolation [CCW_R]			
16-bit instruction	-	32-bit instruction	CCW_R
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the radius of the arc	Double words, 32-bit
S3	Specify the starting speed at the starting point of the two axes	Double words, 32-bit
S4	Specify the stop speed at the end point of the two axes	Double words, 32-bit
S5	Max speed of the two axes	Double words, 32-bit

D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0~S5	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

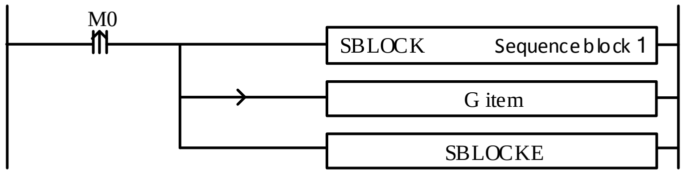
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

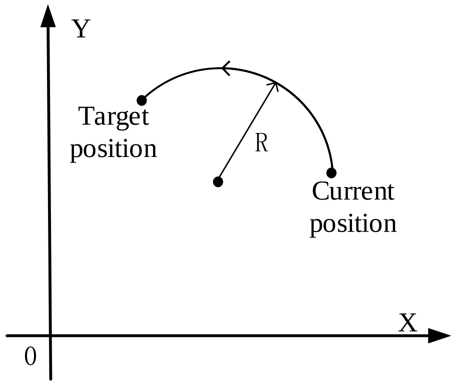
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
radius	The radius is different and the path is different	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Start speed	The start speed from the starting point	Must set
Stop speed	The stop speed at the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

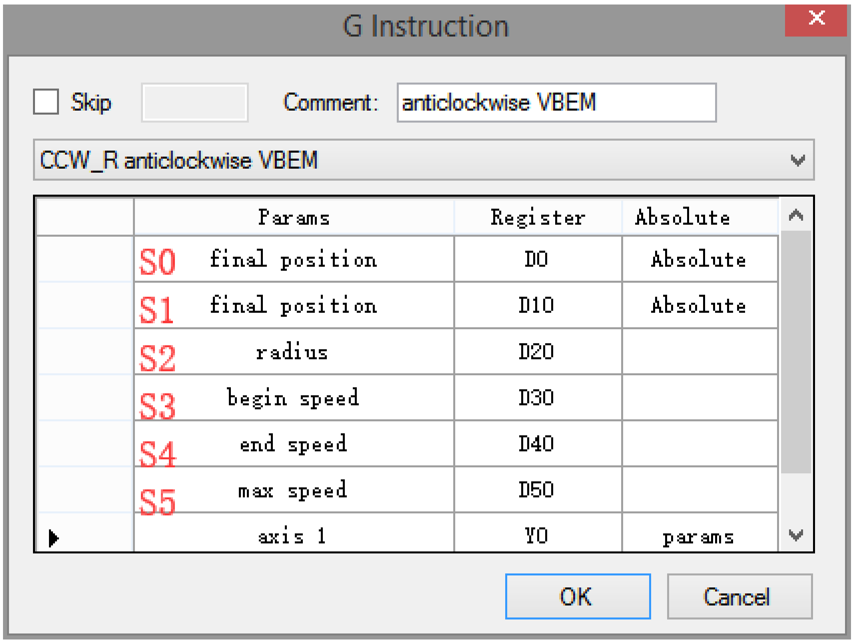


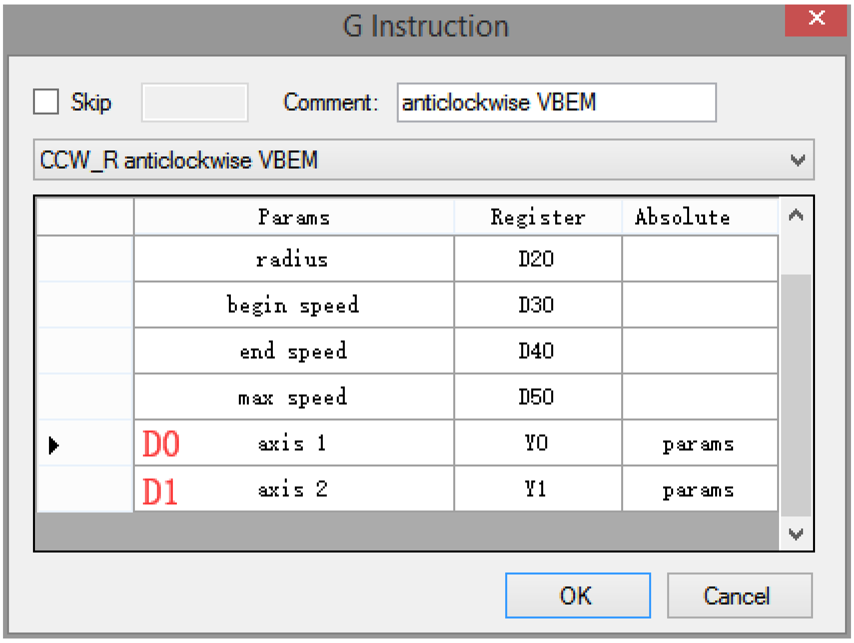
When the CCW_R instruction of arc interpolation (mode 3) is executed, the two axes will run at the set max synthesis speed, start speed and stop speed. As shown in the following figure:



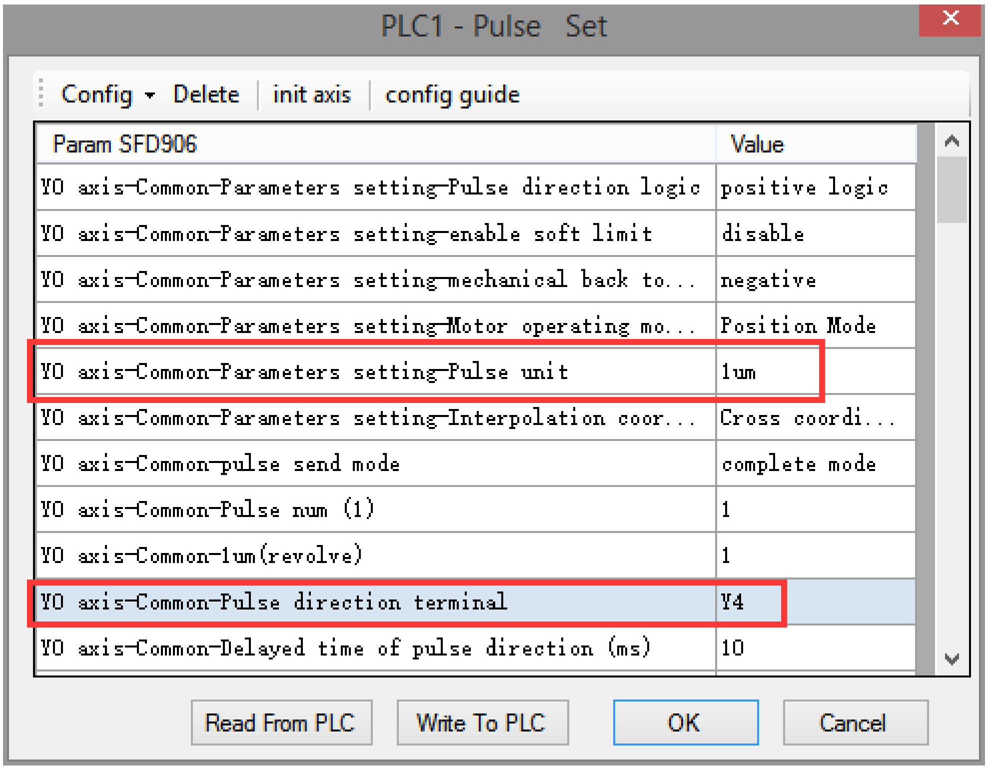
CCW_R anticlockwise arc interpolation

The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:

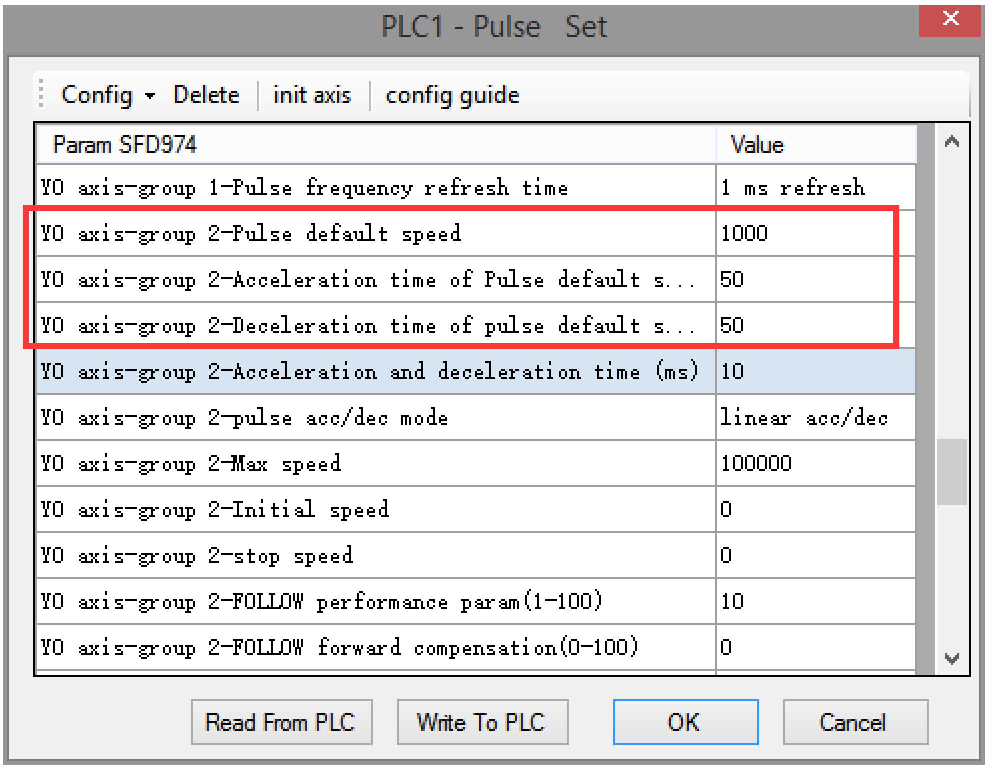




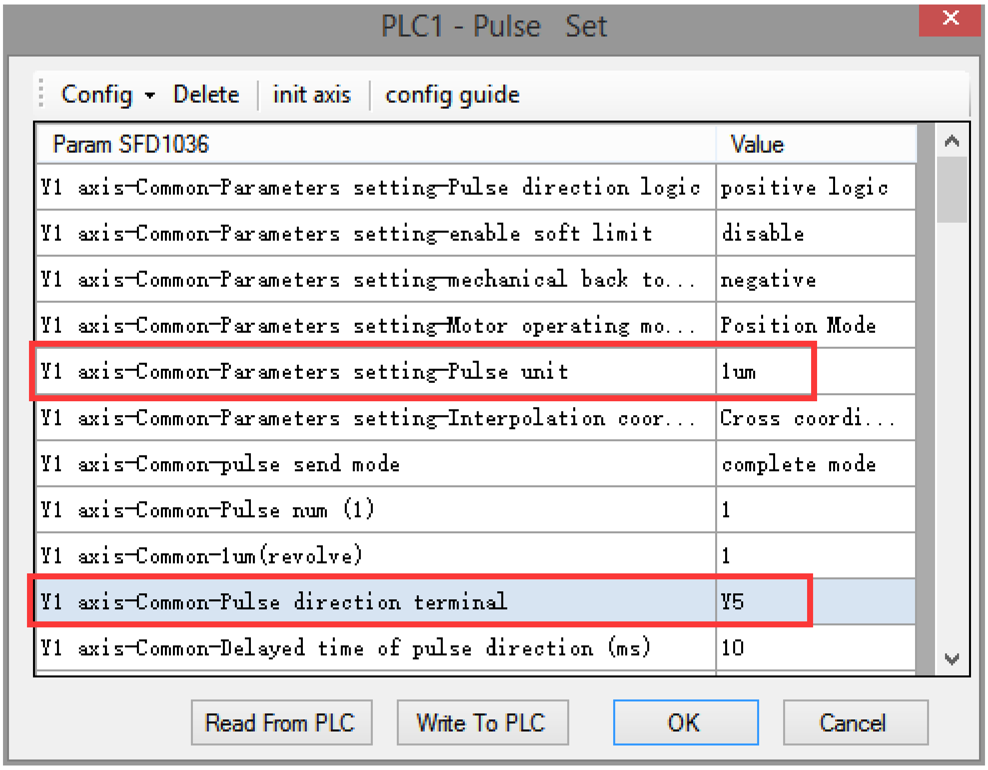
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the radius, D30 specifies the start speed, D40 specifies the stop speed, D50 specifies the max speed.

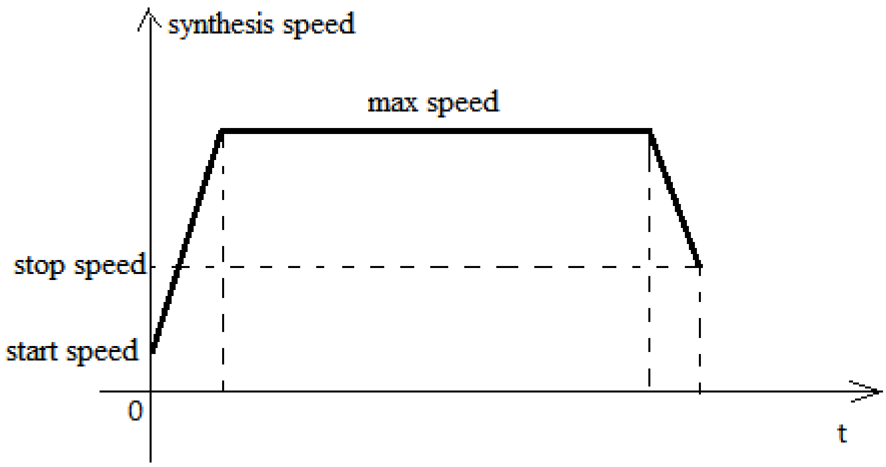
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D30 = 50Hz, D40 = 20, D50 = 2000, when M0 rises, execute CCW_R command, accelerate from the starting position (1000,1000) at speed 50Hz to the maximum speed (2000Hz), and stop at the end speed of 20Hz when moving to the target position.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the CCW_R instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.
- When the radius is positive, the arc is inferior; when the radius is negative, it is major arc.

Note: In this mode, the starting speed (S3), the ending speed (S4) and the maximum speed (S5) are all expressed as the two-axis synthesis speed, as shown in the following figure:



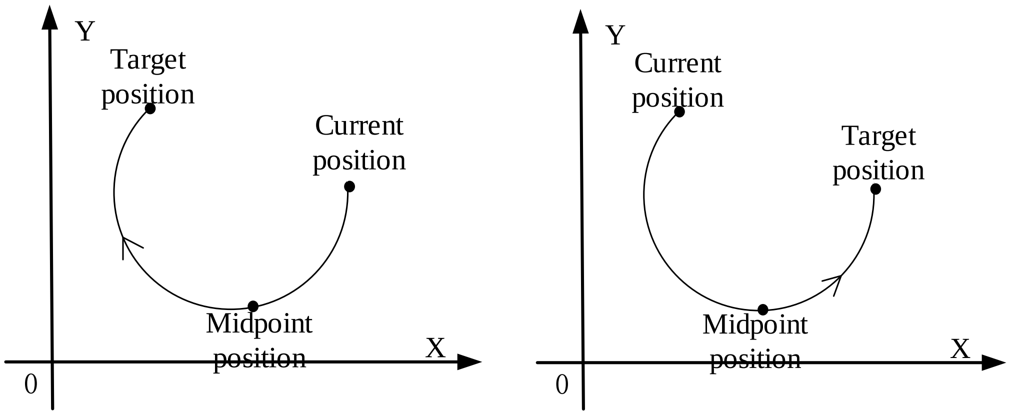
When there are multiple continuous linear/arc interpolation instructions and the speed between them needs to be constant and jump directly, the termination speed and maximum speed of the previous linear/arc interpolation can be set the same as the starting speed and maximum speed of the next segment.

When mode 3 is used, the starting and ending speed in the pulse parameter configuration tables of axis 1 and axis 2 are only effective for calculating the slope of pulse acceleration and deceleration.

2-4-8. Three points arc [ARC]

Three-point arc interpolation ARC mainly determines a section of arc clockwise or counter-clockwise through the current position of the arc, the target position and a midpoint position on the arc.

Note: The midpoint position on the arc refers to any point position between the current position and the target position on the drawn arc. As shown in the following figure:



When the target position is set to the same position as the current position (that is, two points become a point), the next circle can not be determined by two points (in three points, as long as two points coincide or three points are in a straight line, it can not form an arc), so this mode can not draw a whole circle. Three-point arc interpolation ARC has three modes, the following will be used one by one.

Mode 1: ARC three-point arc

1. Instruction overview

Three-point arc interpolation instruction, operate according to the set default speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Three-point arc interpolation [ARC]			
16-bit instruction	-	32-bit instruction	ARC
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Axis 1 midpoint position	Double words, 32-bit
S3	Axis 2 midpoint position	Double words, 32-bit
D0	Pulse output port of axis 1	Bit

D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	●	●	●	●							
S1	●	●	●	●							
S2	●	●	●	●							
S3	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

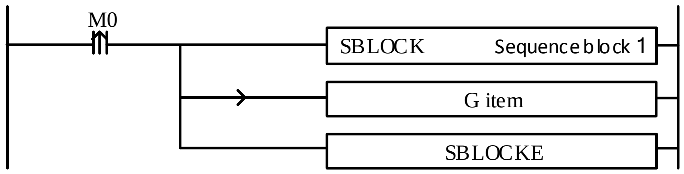
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

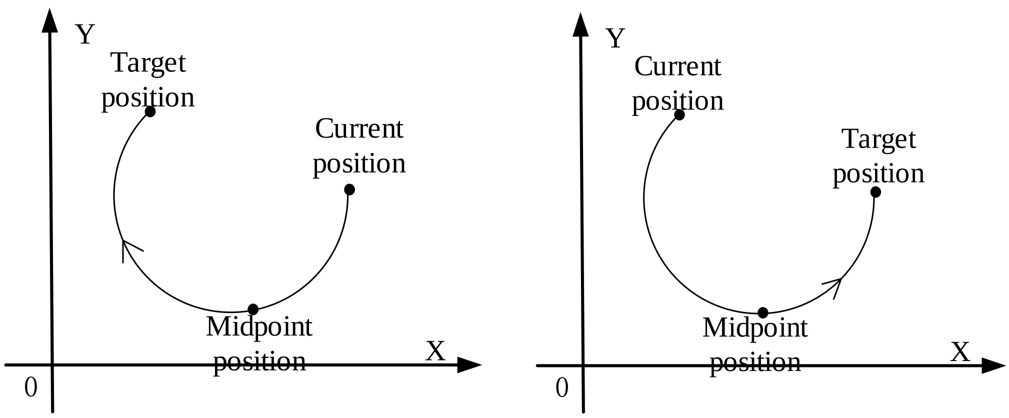
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Midpoint position	Determining the position of the midpoint of an arc according to its path	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	Must set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》

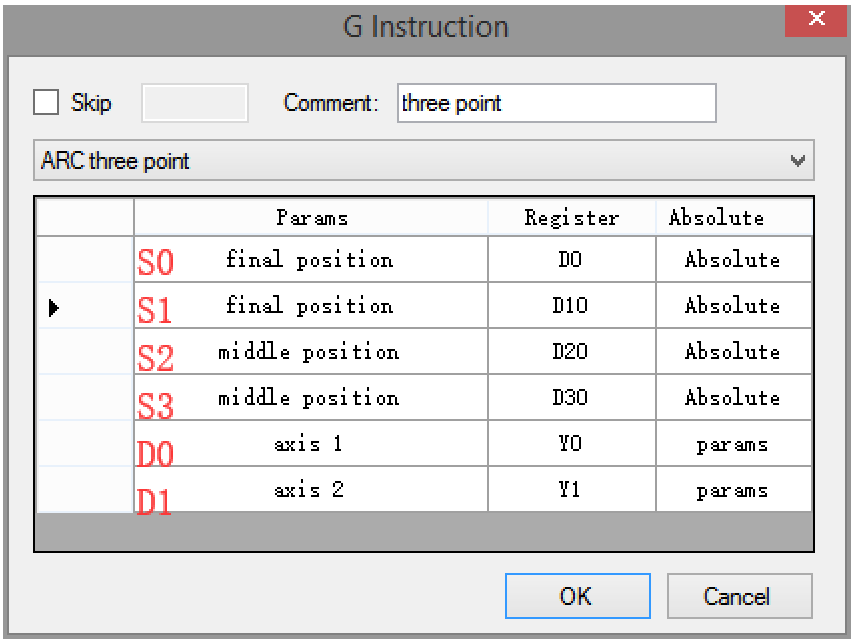


When the ARC instruction of arc interpolation (mode 1) is executed, the two axes will run at the highest synthesis speed. As shown in the following figure:

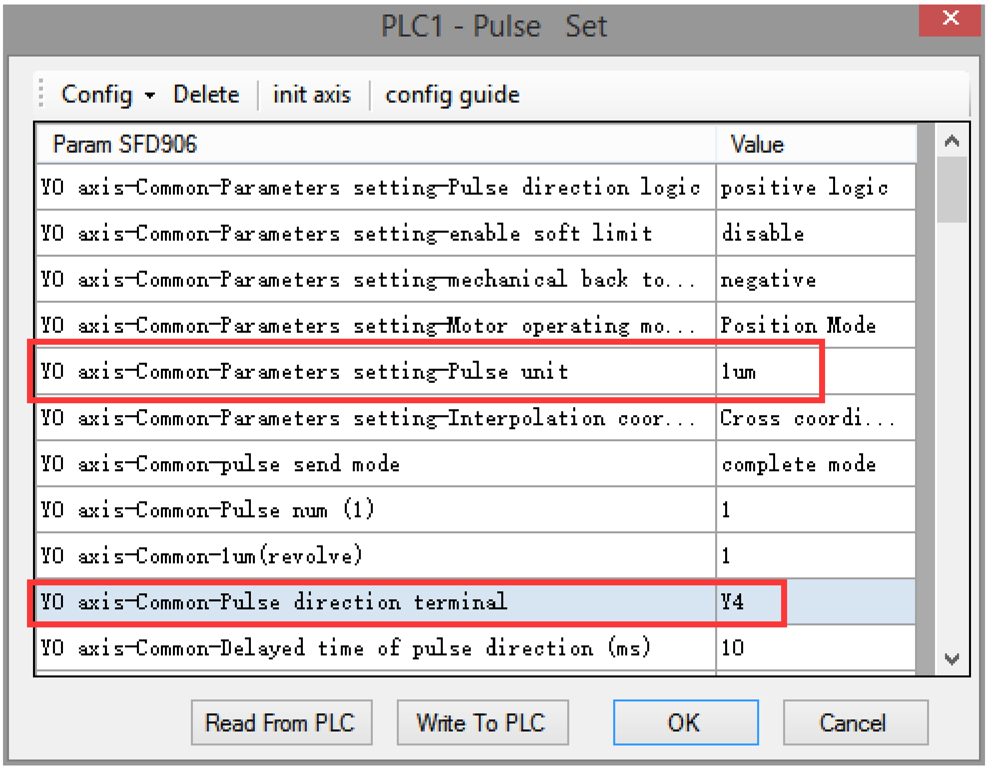


ARC arc interpolation

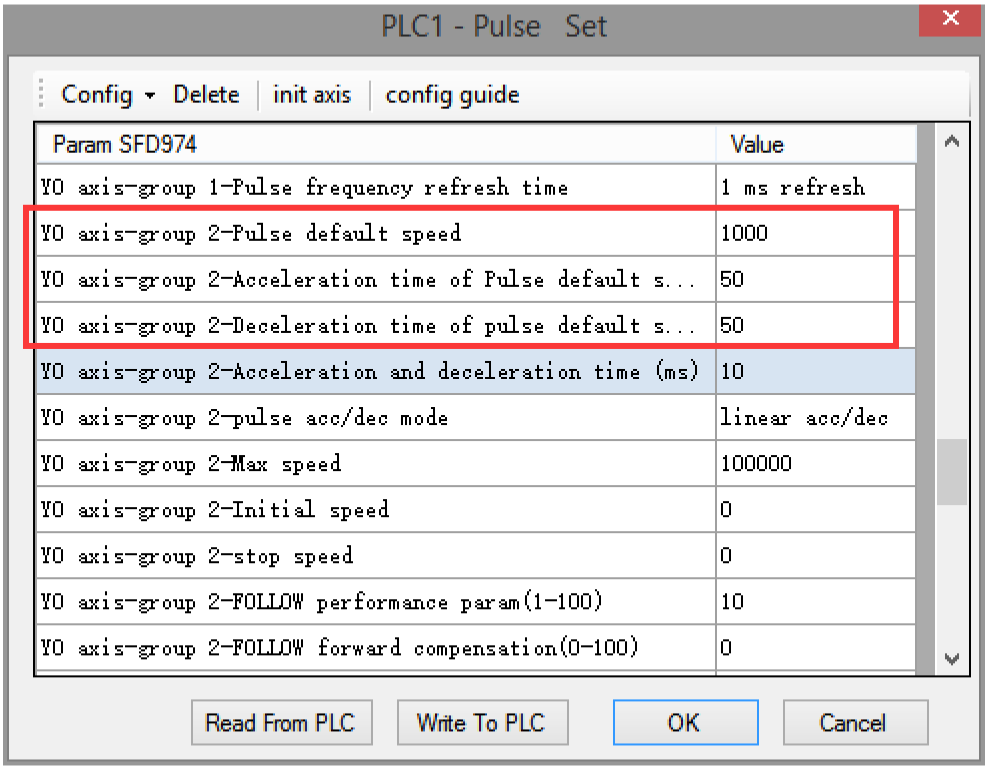
The parameter configuration is shown in the following figure:
Double-click G item and pop up the configuration panel. Set it as follows:



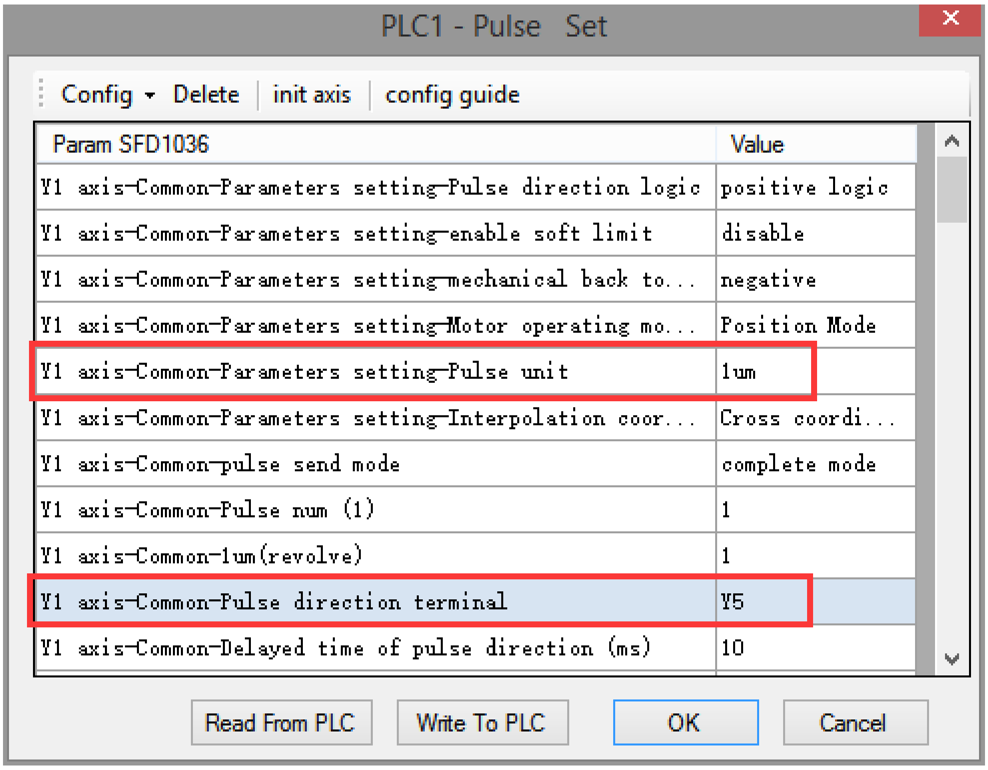
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the midpoint of axis 1 and D30 specifies the midpoint of axis 2.
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, when M0 rises, execute ARC command, move from the starting position (1000, 1000) to the target position at the default speed of 1000Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the ARC instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 2: ARC three-point arc VM

1. Instruction overview

Three-point arc interpolation instruction, operate according to the set maximum synthetic speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Three-point arc interpolation [ARC]			
16-bit instruction	-	32-bit instruction	ARC
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Specify the midpoint of axis 1	Double words, 32-bit
S3	Specify the midpoint of axis 2	Double words, 32-bit
S4	Max speed of the two axes	Double words, 32-bit
D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0~S4	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

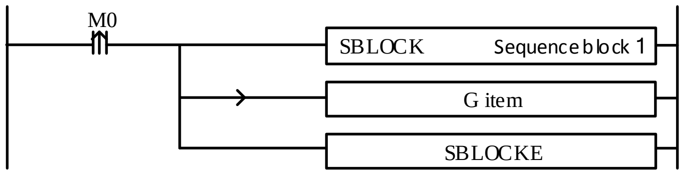
4. Parameter setting

Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set

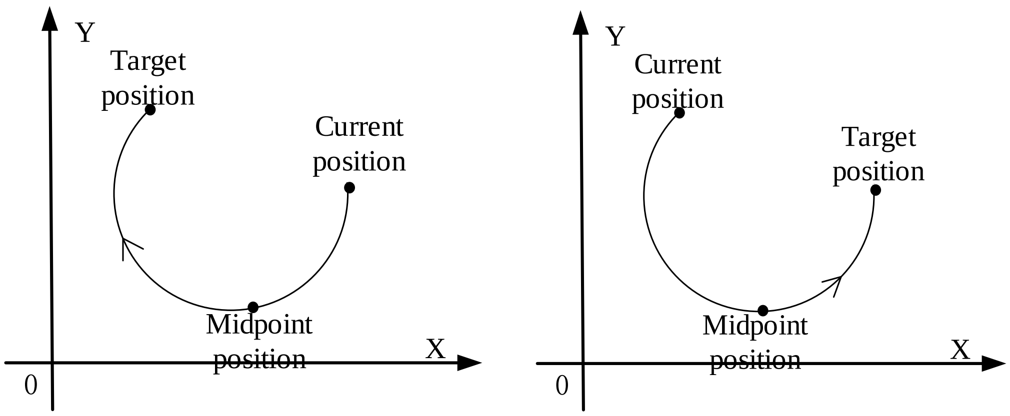
Midpoint position	Determining the midpoint position according to the arc path	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



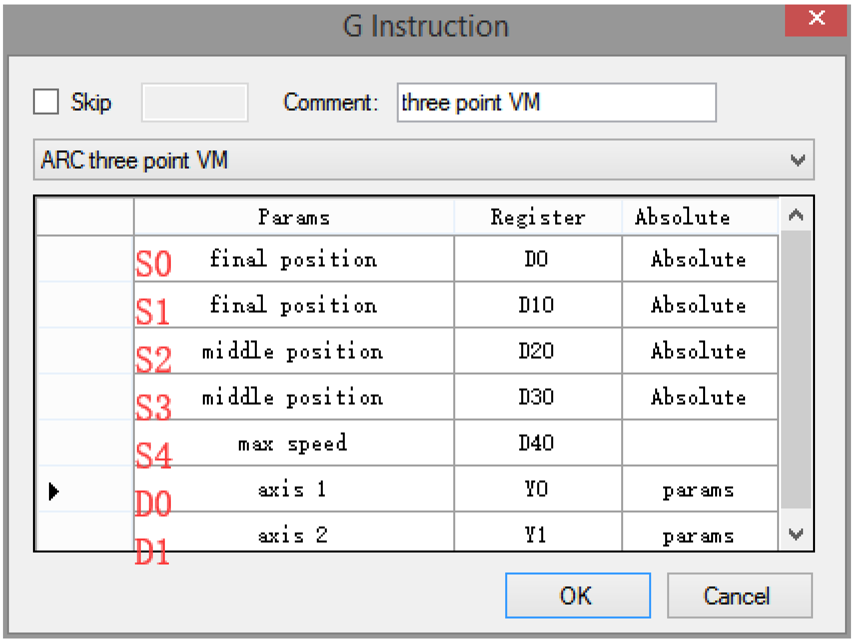
When the ARC instruction of arc interpolation (mode 2) is executed, the two axes will run at the set max synthesis speed. As shown in the following figure:



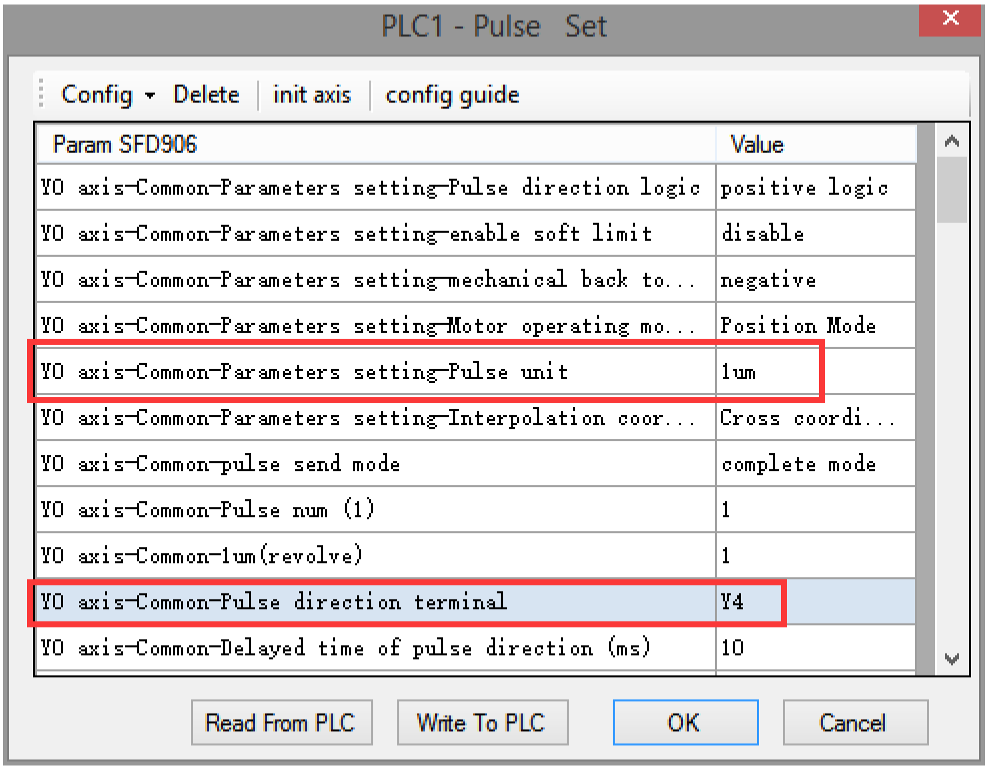
ARC arc interpolation

The parameter configuration is shown in the following figure:

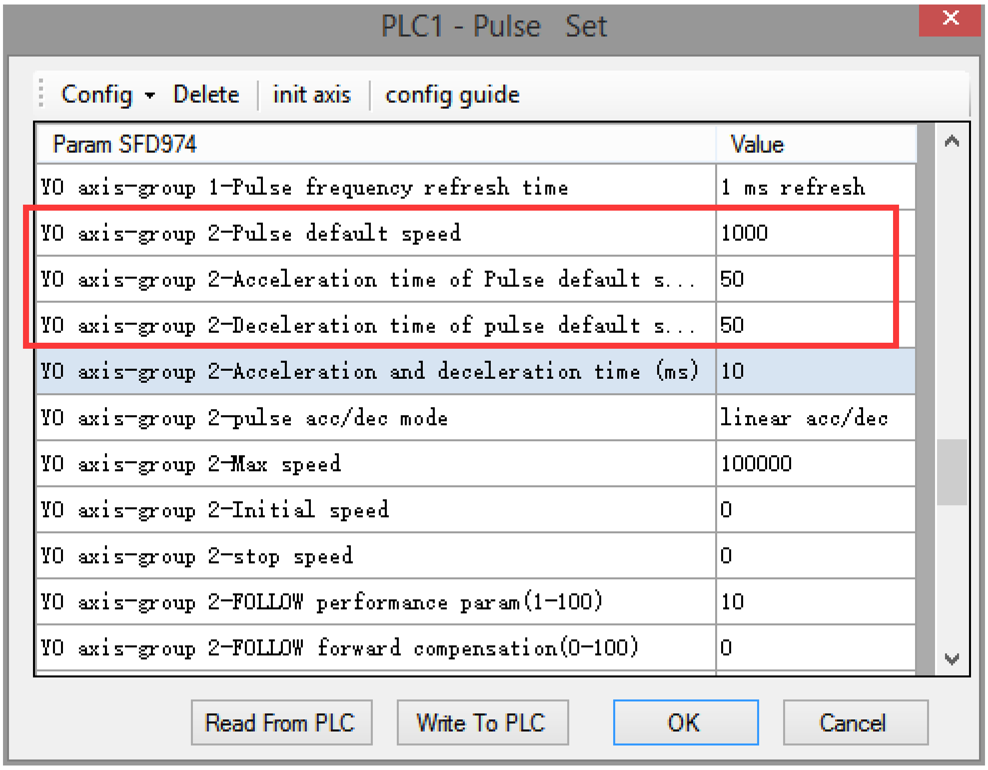
Double-click G item and pop up the configuration panel. Set it as follows:



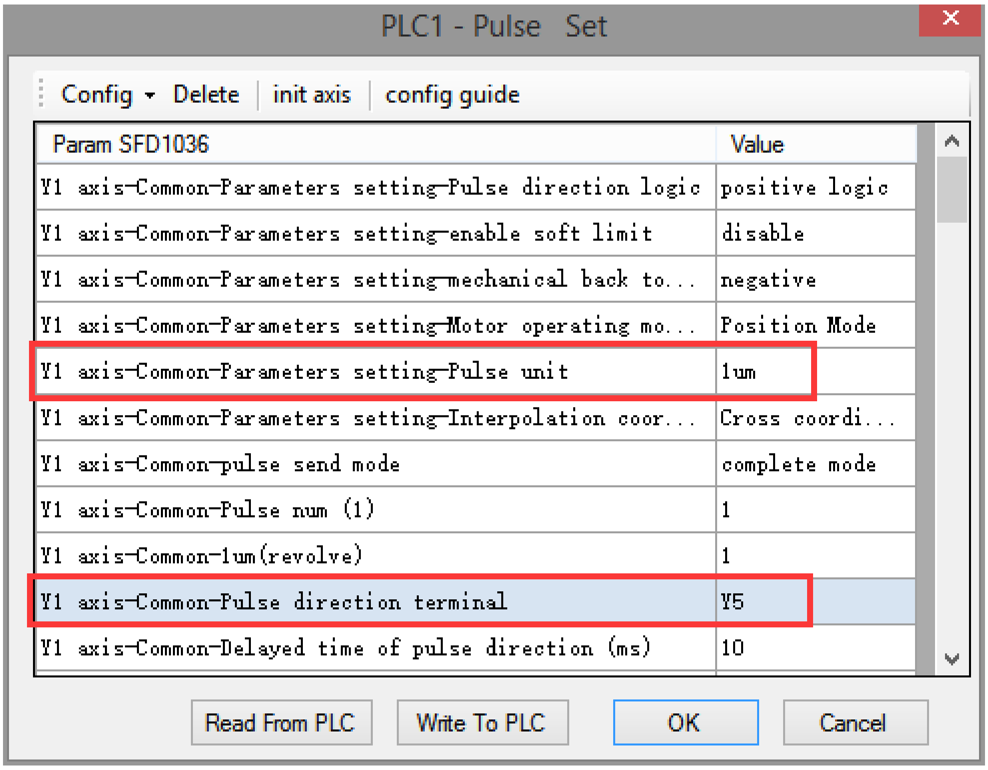
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the midpoint of axis 1 and D30 specifies the midpoint of axis 2, D40 specifies the max speed.

- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 500Hz, when M0 rises, execute ARC command, move from the starting position (1000, 1000) to the target position at the max speed of 500Hz.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the ARC instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Mode 3: ARC three-point arc VBEM

1. Instruction overview

Three-point arc interpolation instruction, operate according to the set maximum synthetic speed, start speed and stop speed. This instruction can only be used in BLOCK. See Section 2-2 for specific usage.

Three-point arc interpolation [ARC]			
16-bit instruction	-	32-bit instruction	ARC
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Axis 1 target position	Double words, 32-bit
S1	Axis 2 target position	Double words, 32-bit
S2	Axis 1 midpoint position	Double words, 32-bit
S3	Axis 2 midpoint position	Double words, 32-bit
S4	Specify the starting speed at the starting point of the two axes	Double words, 32-bit
S5	Specify the stop speed at the end point of the two axes	Double words, 32-bit
S6	Max speed of the two axes	Double words, 32-bit

D0	Pulse output port of axis 1	Bit
D1	Pulse output port of axis 2	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0~S6	●	●	●	●							

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D0		●					
D1		●					

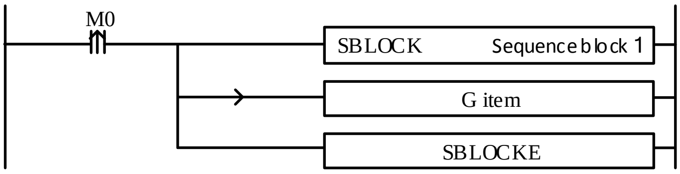
*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

4. Parameter setting

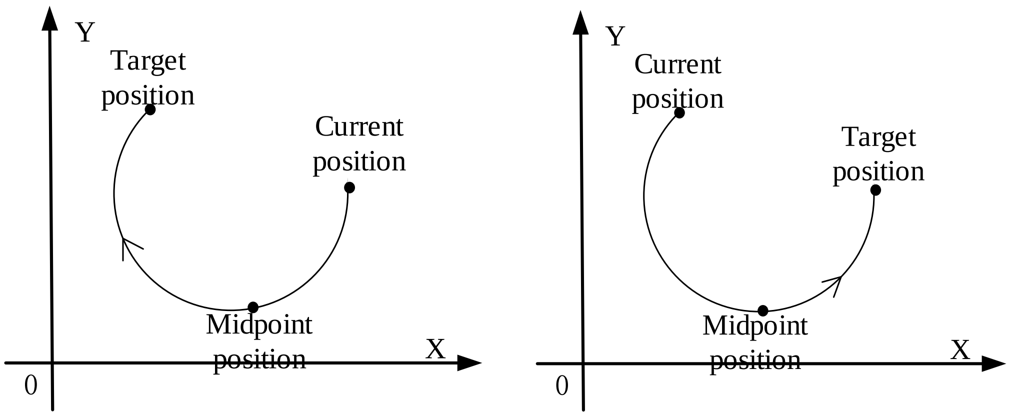
Related parameters	Setting	Note
Final position	Determine the end point position according to relative/absolute mode	Must set
Relative/absolute	Relative: the above position as a reference; absolute: the origin as a reference	Must set
Midpoint position	Determine the midpoint position according to the shape of the arc	Must set
Max speed	Specify maximum smooth running speed of two axes	Must set
Start speed	The start speed from the starting point	Must set
Stop speed	The stop speed at the end point	Must set
Pulse output port of axis 1	Arbitrary specify pulse output point	Must set
Pulse output port of axis 2	Arbitrary specify pulse output point	Must set
Direction port of axis 1	Arbitrarily specify idle output points, set in system parameters	Must set
Direction port of axis 2	Arbitrarily specify idle output points, set in system parameters	Must set
Pulse unit	The pulse number or equivalent are acceptable. Set in axis 1 system parameters	Must set
Default speed	set in axis 1 group 2 parameters	No need to set
Acceleration time	Set in axis 1 group 2 parameters	No need to set
Deceleration time	Set in axis 1 group 2 parameters	No need to set

Function and action

《Instruction format》



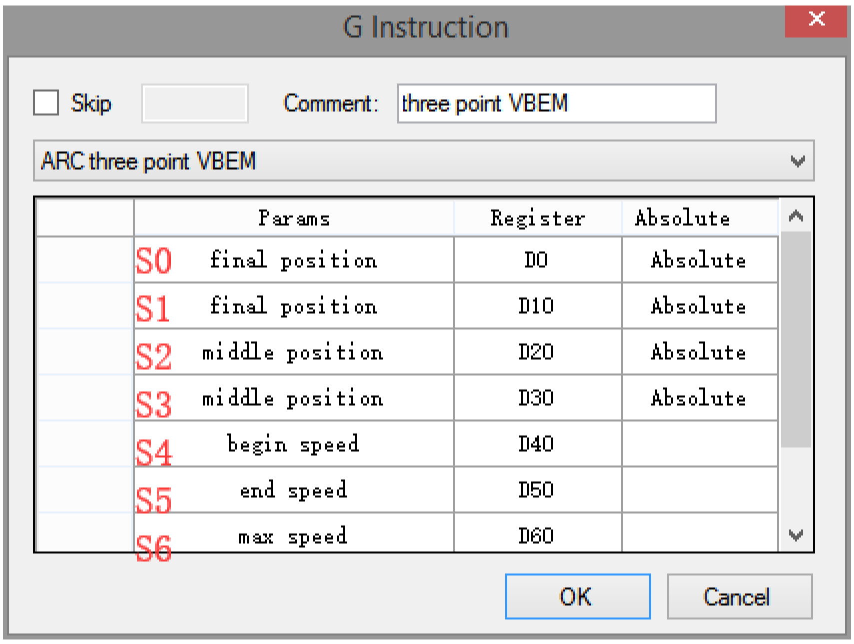
When the ARC instruction of arc interpolation (mode 3) is executed, the two axes will run at the set max synthesis speed, start speed and stop speed. As shown in the following figure:

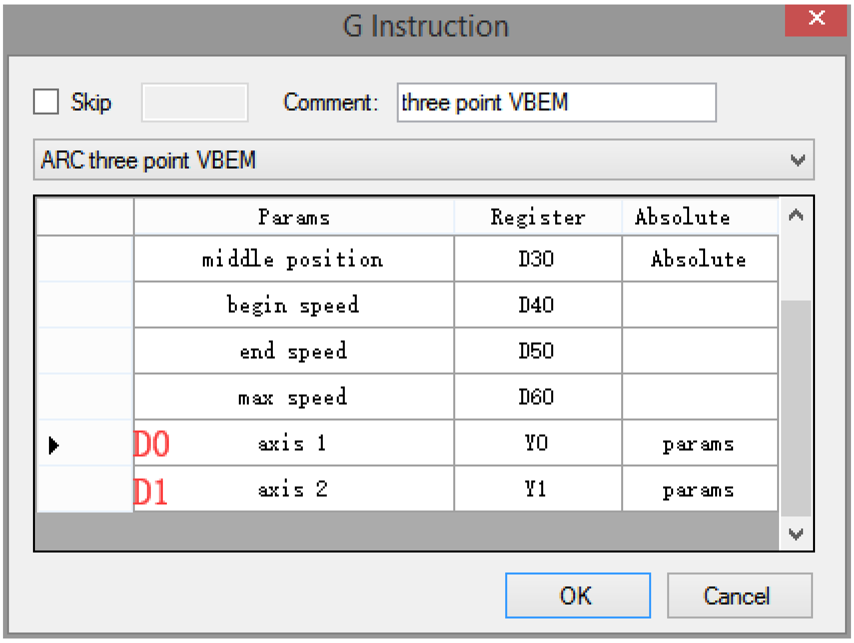


ARC arc interpolation

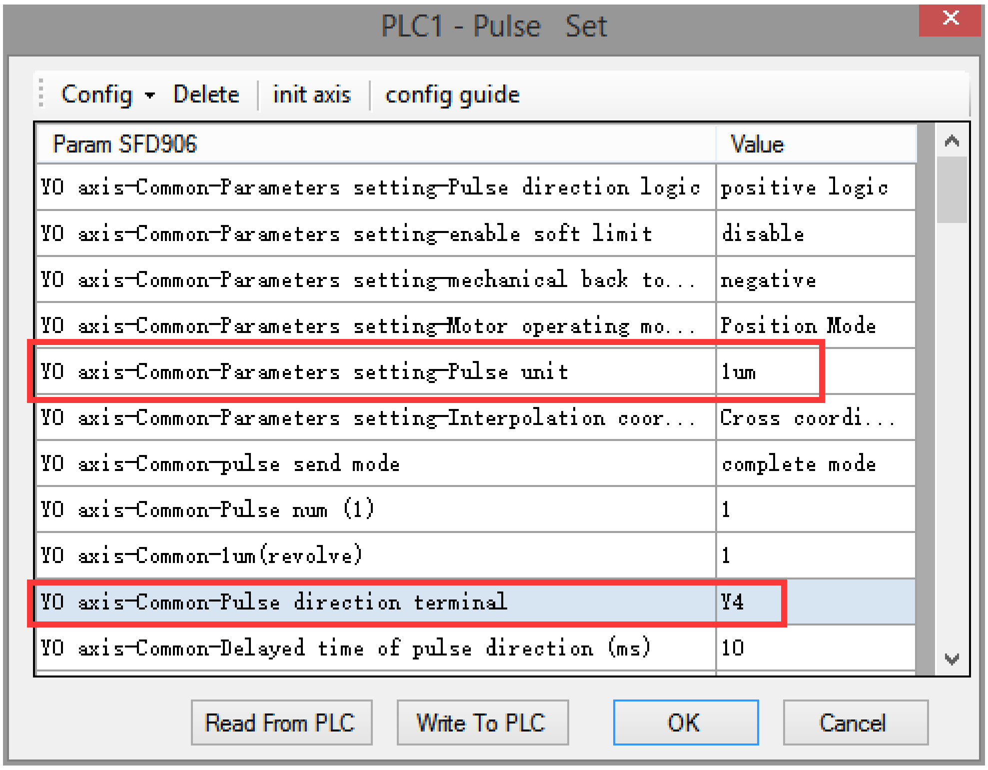
The parameter configuration is shown in the following figure:

Double-click G item and pop up the configuration panel. Set it as follows:

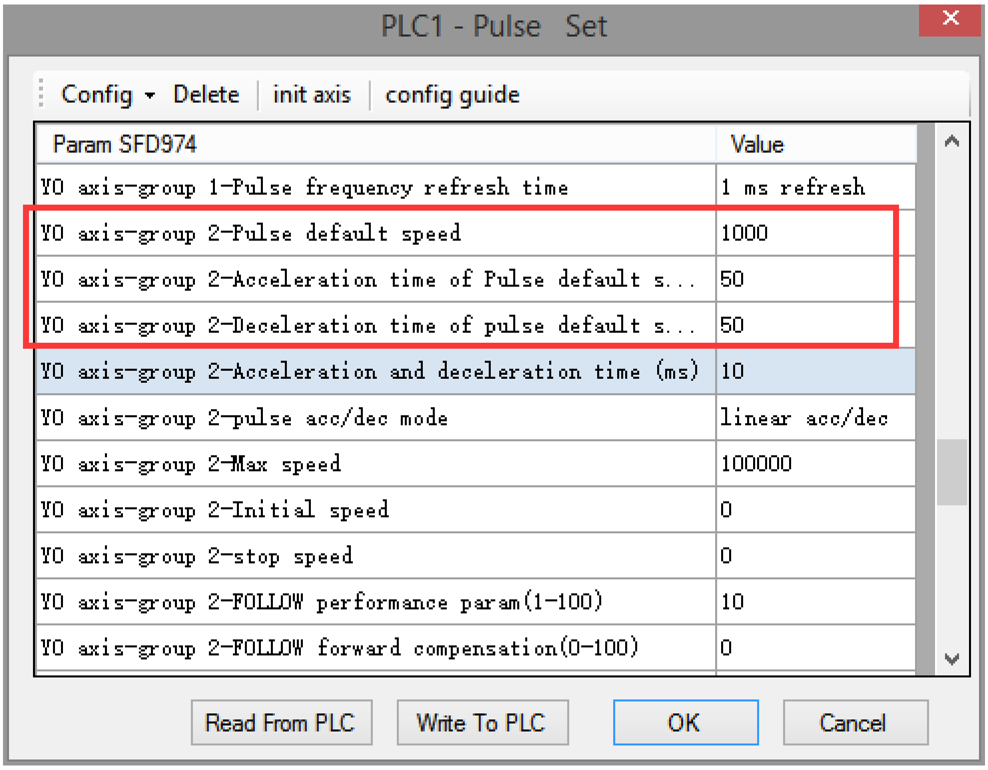




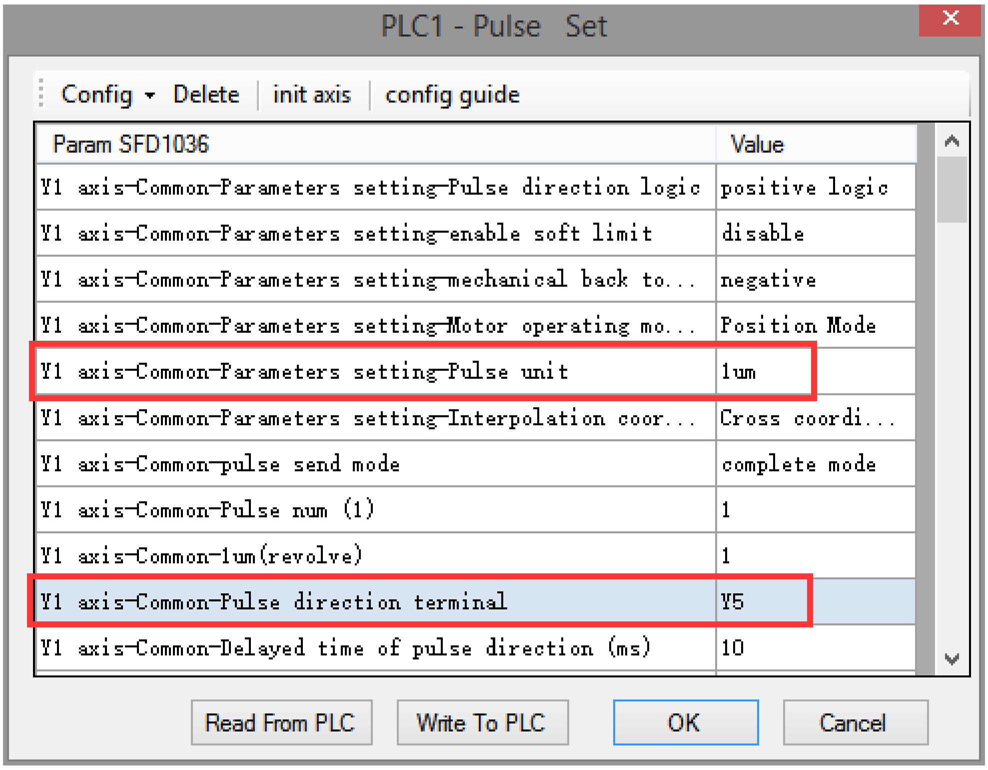
Instruction configuration



Axis Y0 system parameters (1)



Axis Y0 system parameters (2)



Axis Y1 system parameters

- As shown in the figure, D0 specifies the final position of axis 1 and D10 specifies the final position of axis 2, D20 specifies the midpoint position of axis 1, D30 specifies the midpoint position of axis 2, D40 specifies the start speed, D50 specifies the stop speed, D60 specifies the max speed.

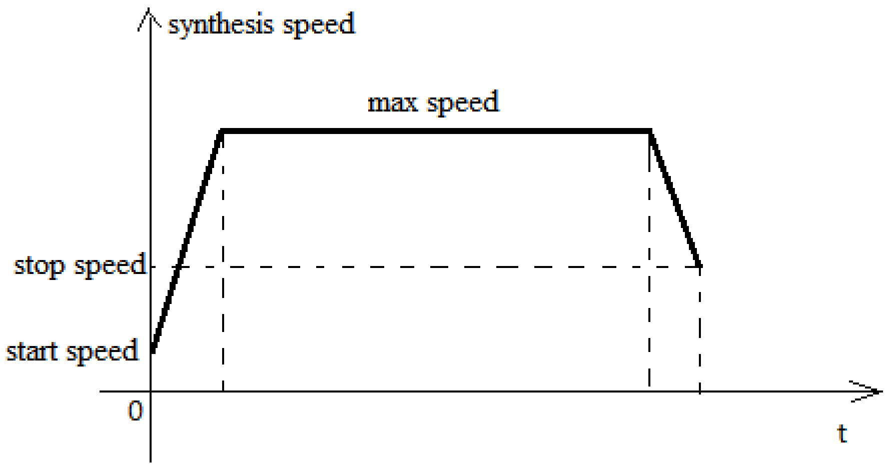
- Y0 is the pulse output port of axis 1 and Y1 is the pulse output port of axis 2. See Sections 2-3 for other optional ports.
- The directional terminals are Y4 and Y5, it is ON for the forward pulse and OFF for the reverse pulse.
- Pulse frequency range: 1Hz ~ 100KHz; Acceleration and deceleration time: 0 ~ 65535ms.
- Location movement can be viewed in equivalent cumulative registers HSD2 and HSD6.
- Assuming HSD2 = 1000, HSD6 = 1000, D0 = 5000, D10 = 2000, D40 = 50Hz, D50 = 20, D60 = 2000, when M0 rises, execute ARC command, accelerate from the starting position (1000,1000) at speed 50Hz to the maximum speed (2000Hz), and stop at the end speed of 20Hz when moving to the target position.

(1) When the end point is in absolute mode, the target position is (5000,2000)

(2) When the end point is in the relative mode, the target position is (6000,3000)

- When the ARC instruction is running, the pulse flag bit corresponding to its output port Y will be set on.
- The completion of the interpolation instruction can be judged by BLOCK executing flag bit. For example, the flag bit of BLOCK1 is SM300, when SM300 changes from ON to OFF, it means that BLOCK1 has finished executing.

Note: In this mode, the starting speed (S4), the ending speed (S5) and the maximum speed (S6) are all expressed as the two-axis synthesis speed, as shown in the following figure:



When there are multiple continuous linear/arc interpolation instructions and the speed between them needs to be constant and jump directly, the termination speed and maximum speed of the previous linear/arc interpolation can be set the same as the starting speed and maximum speed of the next segment.

When mode 3 is used, the starting and ending speed in the pulse parameter configuration tables of axis 1 and axis 2 are only effective for calculating the slope of pulse acceleration and deceleration.

2-4-9. Follow [FOLLOW] [FOLLOW_AB]

Follow-up instructions are divided into single-phase incremental follow-up [FOLLOW] and AB phase follow-up [FOLLOW_AB], which will be described in detail below.

1. Instruction overview

Single-phase/AB-phase high-speed counter follow instructions. The instructions can be written directly in the main program or process.

Follow instruction [FOLLOW] [FOLLOW_AB]			
16-bit instruction	FOLLOW, FOLLOW_AB	32-bit instruction	-
Execution condition	Rise/fall edge of coil	Suitable model	XDM, XDME, XLME
Firmware	V3.3 and above	Software	V3.3 and above

2. Operand

Operand	Function	Type
S0	Single-phase/AB phase high speed counter	Double words, 32-bit
S1	Register address of multiplication coefficient	Single word, 16-bit
S2	Register address of division coefficient	Single word, 16-bit
S3	System parameter block number	Single word, 16-bit
D	Pulse output port	Bit

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S0	Only can be High speed counter										
S1	●	●	●	●						●	●
S2	●	●	●	●						●	●
S3	●	●	●	●					●	●	●

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●					

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM; DS denotes DS DHS; M denotes M HM SM; S denotes S HS; T denotes T HT; C denotes C HC.**

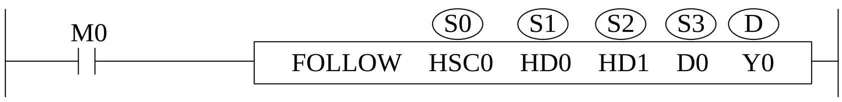
4. Parameter setting

Related parameters	Settings	Note
High speed counter	The high-speed counter corresponding to FOLLOW must be single-phase incremental mode The high-speed counter corresponding to FOLLOW_AB must be AB phase mode.	Must set
Multiplication coefficient/division coefficient	Range: -1000~1000 and not equal to 0 (follow-up instructions will not be executed when out of range). The multiplication coefficient/division coefficient is negative to indicate the positive count and send the reverse pulse. Dynamic modifications can take effect immediately.	Must set
System parameter block number	System parameters corresponding to pulse output axis, the range is 1~4	Must set
Pulse output port	Arbitrary designated pulse output point	Must set
Pulse direction	It can be set in the selected system parameter block or set separately.	Must set
Pulse unit	Must set to pulse number, please set in the system parameter of the output axis	Must set
FOLLOW performance parameter	1~100 (report error when out of range), default value is 50	No need to set
FOLLOW feedforward compensation	0~100 (report error when out of range), default value is 0	No need to set
Positive/negative limit	Hard limit can be set in system parameters of output axis	No need to set
Positive/negative value of soft limit	Soft limit can be set in system parameters of output axis	No need to set

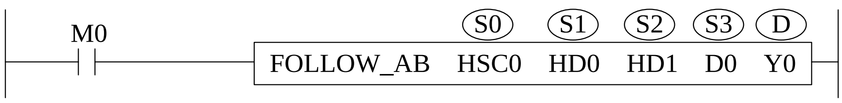
Function and action

《Instruction format》

For single-phase incremental mode high speed counter:



For AB-phase mode high speed counter:



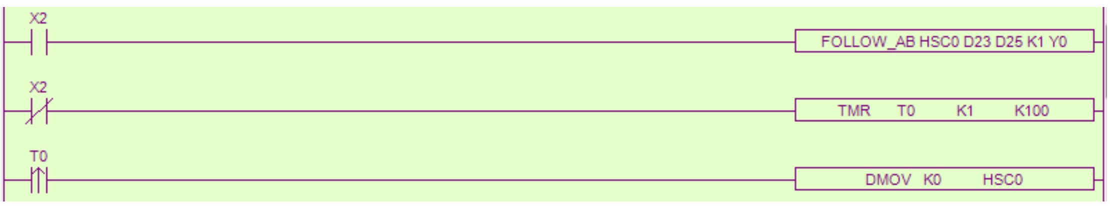
- FOLLOW/FOLLOW_AB instruction is a servo function. Through the pulse feedback of encoder or hand pulse generator, the frequency and number of input pulses are measured by PLC in real time. Through the proportional relationship between multiplication coefficient and division coefficient, the corresponding pulse frequency and the number of pulses are output to control the stepping or servo motor.
- This instruction is generally used for manual adjustment of CNC system, and it is used for advancing and retreating of the operating table of the pulse generator by hand. It can also be used in some special projects where precise synchronous control is needed.
- Pulse output is based on the variation of HSC0, that is to say, in 4-time mode, if the multiplier/divider coefficient is 1, the output of the pulse is equal to 4 times the input of the pulse. The number of pulses at the output port is stored in the pulse cumulative register, namely HSD0 (double word), HSD4 (double word)... And so on.
- For FOLLOW instructions, the high-speed counter inputs a single-phase pulse, so the number of Y-port pulses is increasing regardless of the input inversion, and the corresponding pulse direction terminal is always ON, which will not be OFF when inversion occurs.
- For FOLLOW_AB instruction, the input of high-speed counter is AB phase pulse. Y port will increase and decrease with the increase of input pulse, and the direction is the same as that of high-speed counter input.
- The forward and reverse flag bit of the follow-up instruction is the direction flag bit of the high-speed counter.
- When the Y0 port outputs the pulse, the SM1000 will be set on.
- Follow-up instruction supports hard limit, soft limit, emergency stop and slow stop functions. See the description of the parameters of the pulse system.
- XDM-24/32 supports 4 channels, XDM-60T10 supports 10 FOLLOW instructions, and can execute 4 or 10 FOLLOW instructions simultaneously.

Note:

(1) During operation, the corresponding HSCD and HSD can not be changed arbitrarily. If it needs to be cleared, it must be cleared at the same time.

(2) If the high-speed counter needs to be cleared, the clearing instruction must be executed after the condition of FOLLOW or FOLLOW_AB is disconnected and at least two scanning cycles are spaced.

For example, after disconnecting the condition X2, a short delay is made, and the clearing instruction is executed after the time is up.



(3) It is forbidden to write two (or more) follow-up instructions to the same high-speed counter

in the program.

(4) It is forbidden to have both FOLLOW (or FOLLOW_AB) and CNT (or CNT_AB) instructions for the same high-speed counter in the program.

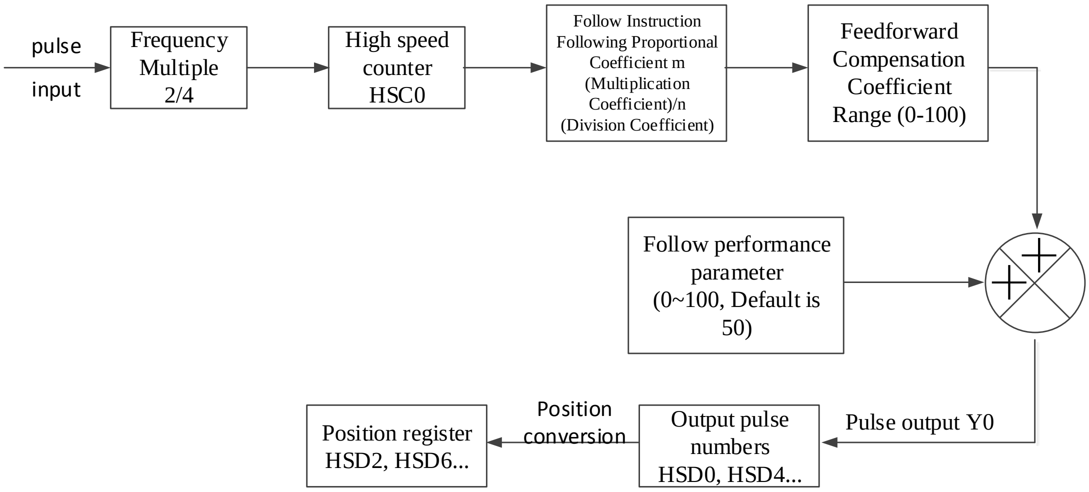
(5) The follow-up instruction can be executed simultaneously with the interpolation instruction, but the output port can not overlap.

(6) High-speed counting must be given pulse input by external input terminal, and can not be used by HSCW writing mode.

(7) Follow-up instructions cannot use the same high-speed counter as high-speed counting read-write instructions. When FOLLOW instructions need to write multiple instructions from the same high-speed counting source, they can be written in different processes, and only one process can be conducted at the same time.

(8) FOLLOW instruction resource conflict is corresponding to AB phase high-speed counting resource conflict.

The following is instruction diagram of FOLLOW/ FOLLOW_AB(take Y0 as an example):



The relationship between follow-up instructions and motion control instructions:

(1) The follow-up command can be used separately from the motion control command. However, when manual pulse generator is needed to adjust the coordinate position, it is necessary to establish the relationship between follow-up and motion control.

(2) When the pulse mode is equivalent, the change of the number of pulses is converted to the change of the position of the corresponding output axis, which is reflected in the HSD2 (double-word) register, so that the follow-up instructions and the motion control system constitute an organic whole. Therefore, the following changes can be directed either to axis 1 or to axis 2.

(3) The change of position is consistent with the change of pulse, which can only increase but not decrease.

FOLLOW performance parameters:

The function of this parameter is similar to the rigidity function of servo driver. The smaller the setting value of this parameter is, the smaller the servo rigidity will be (the greater the delay); the larger the setting value of this parameter is, the greater the servo rigidity will be (the smaller the delay will be). Setting range: 1 ~ 100 (error will be reported if exceeding range), default setting is 50.

FOLLOW feedforward compensation:

(1) There is always a certain delay between receiving and sending out pulses in PLC. In order to reduce the lag effect, the feedforward compensation parameters can be modified to compensate for the lag effect, so that the pulse output has a certain advance, to offset the lag effect. However, if the feedforward parameters are set large, it may lead to entering the compensation cycle, which will lead to the continuous jitter of the motor at the end of the follow-up. Setting range: 0-100 (error will be reported when exceeding the range), default is 0, equivalent to no feedforward compensation.

(2) Normally, this parameter does not need to be set.

Limit bit description (fit for all motion instructions):

(1) When the positive motion is detected, the rising edge of the positive limit is detected, and the deceleration begins until it stops. At this time, only the negative motion can be achieved. In the process of negative motion, only when the descending edge of positive limit is detected, can two-way motion be achieved.

(2) When the negative motion is detected, the rising edge of the negative limit is detected, and the deceleration begins until it stops. At this time, only the positive motion can be achieved. In the process of positive motion, only after the negative limit drop edge is detected, can the two-way motion be achieved.

(3) When the instruction starts to execute, it can only move negatively if it is in the positive limit. If it is in the negative limit, it can only move forward.

2-5. Hardware wiring and precautions

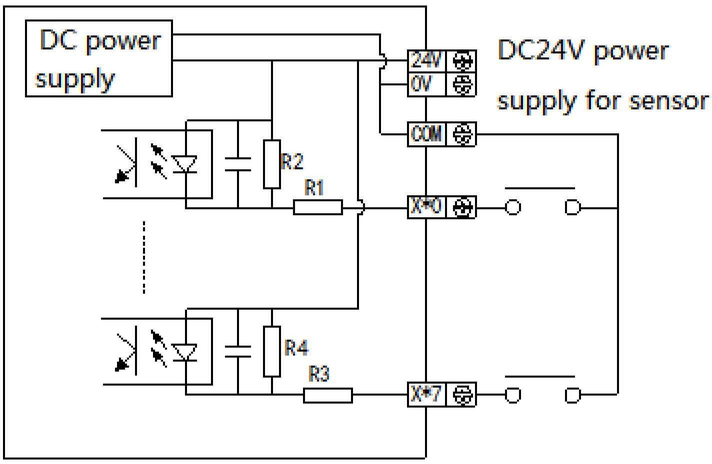
2-5-1. Input wiring

XD series PLC input is divided into NPN and PNP modes (XL series only supports NPN type wiring). The internal structure and wiring mode of the two modes are introduced below.

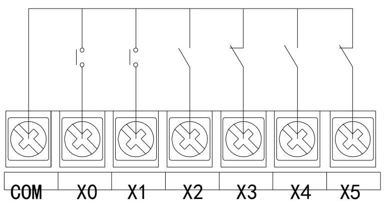
2-5-1-1. XD series PLC input wiring

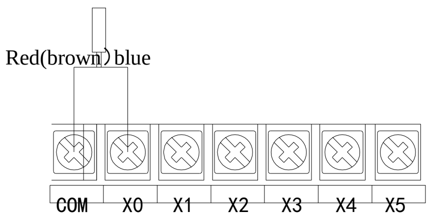
- **NPN mode**

Input signal voltage	DC24V±10%
Input signal current	7mA/DC24V
Input ON current	Below 4.5mA
Input OFF current	Below 1.5mA
Input response time	About 10ms
Input signal mode	Contact input or NPN open collector transistor
Circuit insulation	Photoelectric coupled insulation
Input action display	LED lights when input is ON

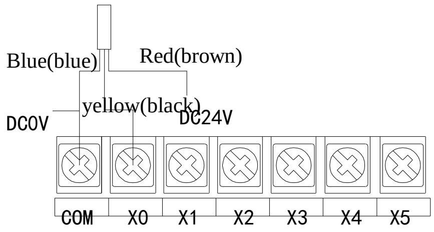


XD series NPN wiring example





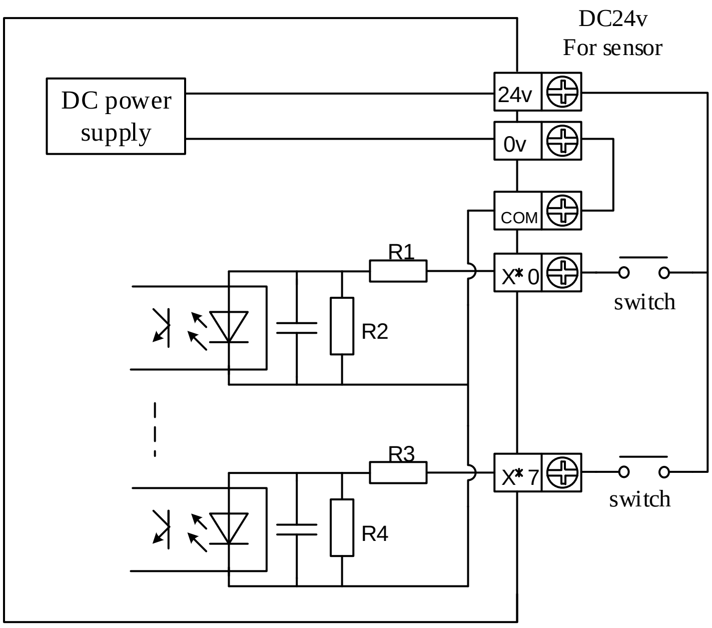
Switch button wiring **two-wire (NO or NC) proximity switch wiring**



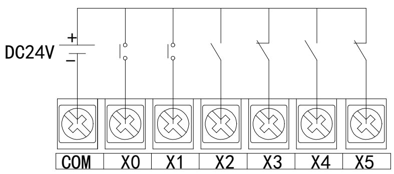
Three-wire (NPN) proximity switch wiring

- **PNP mode**

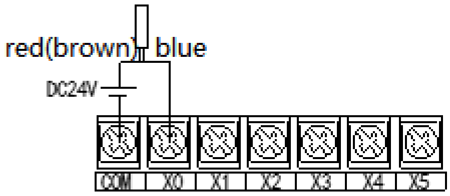
Input signal voltage	DC24V±10%
Input signal current	7mA/DC24V
Input ON current	Below 4.5mA
Input OFF current	Below 1.5mA
Input response time	About 10ms
Input signal mode	Contact input or PNP open collector transistor
Circuit insulation	Photoelectric coupled insulation
Input action display	LED lights when input is ON



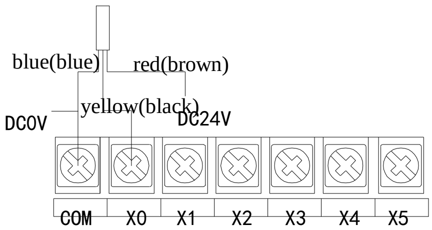
PNP wiring example



Switch button wiring



two-wire(NO or NC) proximity switch wiring



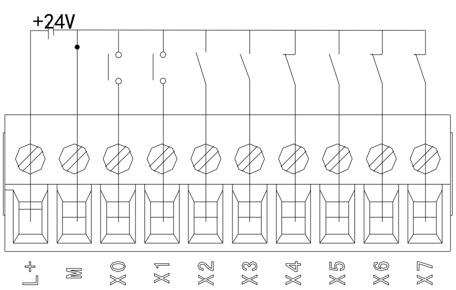
Three-wire (PNP) proximity switch wiring

2-5-1-2. XL series PLC input wiring

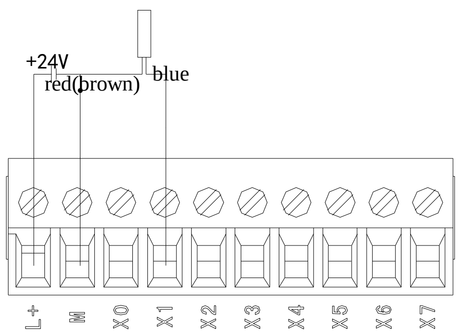
- **Input specifications (NPN mode)**

Input signal voltage	DC24V±10%
Input signal current	7mA/DC24V
Input ON current	Below 4.5mA
Input OFF current	Below 1.5mA
Input response time	About 10ms
Input signal mode	Contact input or NPN open collector transistor
Circuit insulation	Photoelectric coupled insulation
Input action display	LED lights when input is ON

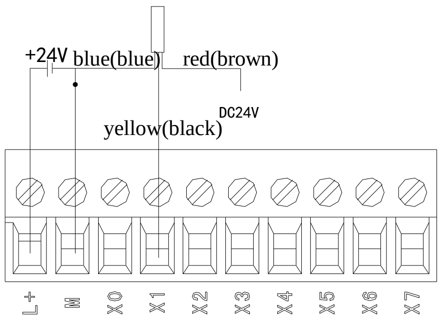
- **XL series PLC NPN input wiring example**



Switch button wiring



two-wire(NO or NC) proximity switch wiring



Three-wire (NPN) proximity switch wiring

2-5-1-3. Attentions for connection of input points

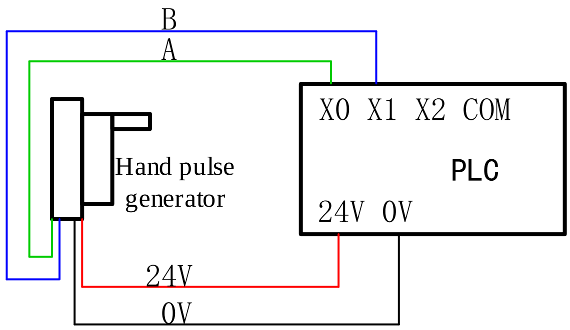
- The input type must be OC signal (collector open circuit signal).
- DC24 does not need to connect DC0V to COM of input point if it uses DC24V provided by PLC body; if it uses external power supply, it must be connected.

2-5-1-4. Hand pulse generator connection

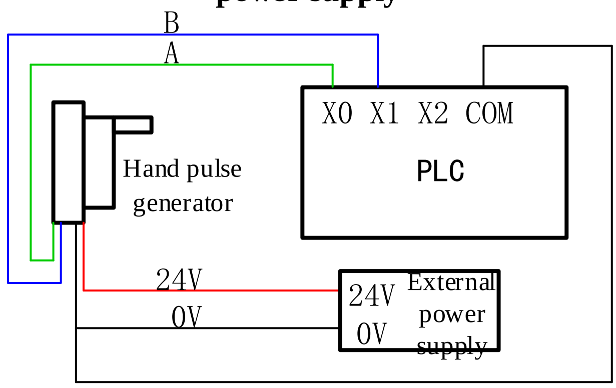
Hand pulse generator is also known as hand artery impulse generator, hand pulse, electronic handwheel and so on. It is used to zero correction and signal segmentation for CNC machine tools, printing machinery, etc. It works like an encoder.

The output signal of the hand pulse generator must be OC (collector open circuit signal) DC24V type. Generally, there will be five wires, three signal wires (A, B, Z), two power wires (24V, 0V), signal wires connected with the corresponding high-speed counting input port of the PLC. The power supply can be supplied by the output 24V of the PLC or by the switching power supply.

Use the 24V from PLC



Use external 24V power supply



Note: When using external switching power supply, the COM of PLC input should be short connected with 0V.

2-5-2. Output wiring

For XD/XL series PLC, the output terminal of motion control command needs high-speed pulse output terminal. Other transistors are ordinary optocouplers. For specifications and introduction, please refer to "XD/XL Series PLC Hardware User Manual".

2-5-1-1. High speed pulse output specification parameters

Model	XDM-24T4/32T4/60T4/60T4L XLME-32T4	XDM-60T10, XDME-60T10
High speed pulse output port	Y0~Y3	Y0~Y11
External power supply	DC5~30V	
Action display	LED light	
Max current	50mA	
Pulse max output frequency	100KHz	

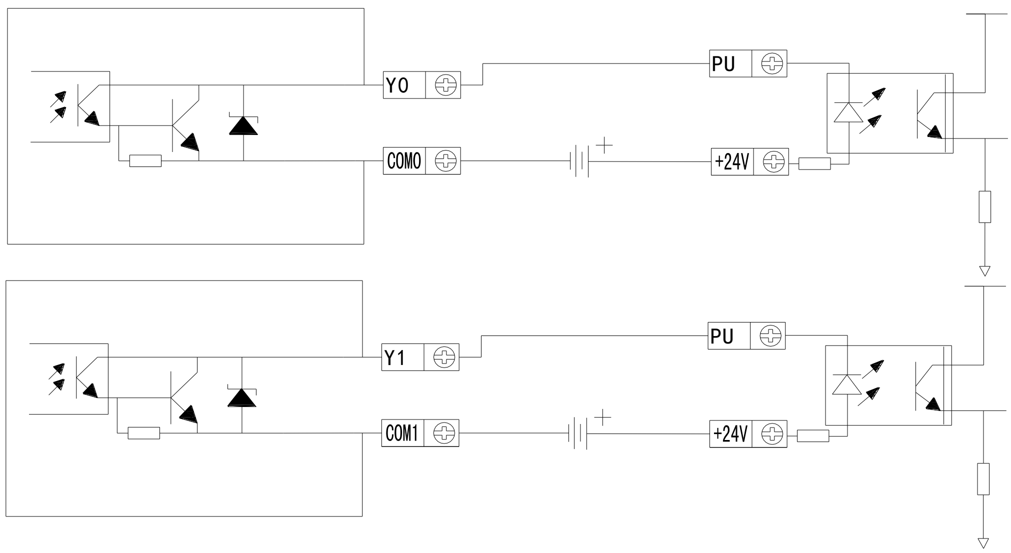
Note: PLC can output 100KHz ~ 200KHz pulses, but it can not guarantee the normal operation of all servos. Please connect about 500Ω resistance between the output and 24V power supply.

2-5-1-2. Cautions for output point connection

If it is XDM-60T10-E or XDME-60T10-E, the output point Y12-Y27 should be used when the output point of the photocoupler is connected with the power load.

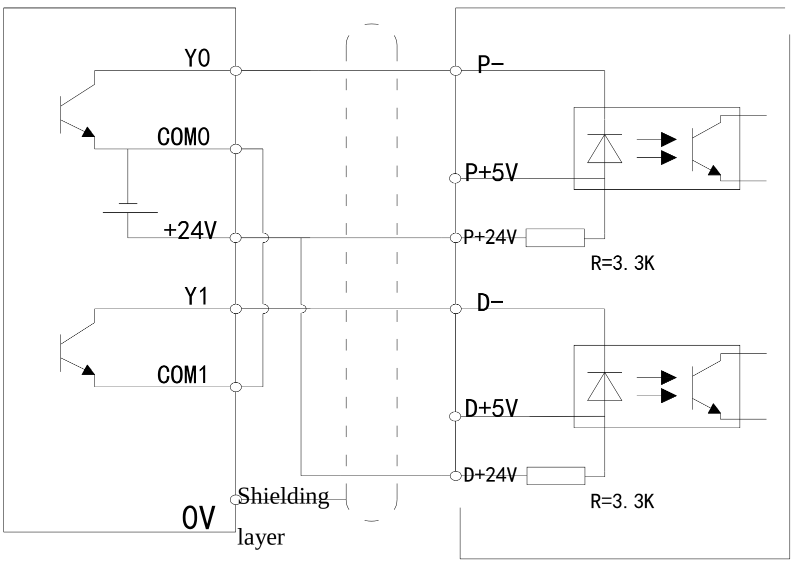
2-5-1-3. Connecting with stepping driver/servo driver

Below is the diagram of the connection between the T-type output terminal and the stepper motor driver.



Note: If the pulse and direction terminals of the stepper motor are driven by DC5V, please connect 2.2KΩ resistance behind the pulse and direction terminals.

Below is the diagram of the connection between the T-type output terminal and XINJE servo motor driver.



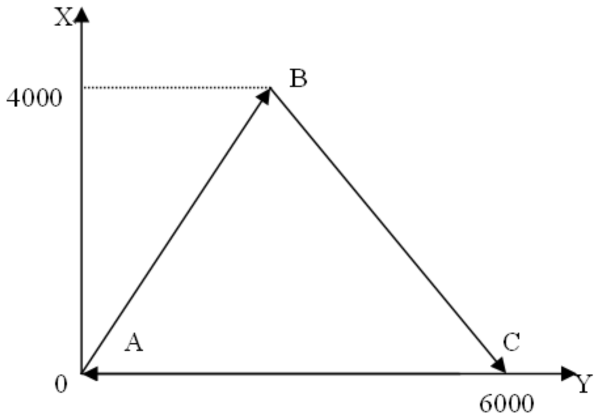
Note: Please suspend P+5V and D+5V.

Detailed hardware wiring diagram refers to "XD/XL Series PLC Hardware User Manual.

2-6．Examples

2-6-1．Isosceles triangle

Step out of an isosceles triangle with a side length of 5000 and a bottom of 6000. The starting point is A (0, 0), from A (0, 0) to B (3000, 4000), then from B (3000, 4000) to C (6000, 0), and finally from C (6000, 0) back to the starting point A (0, 0), as shown in the figure:



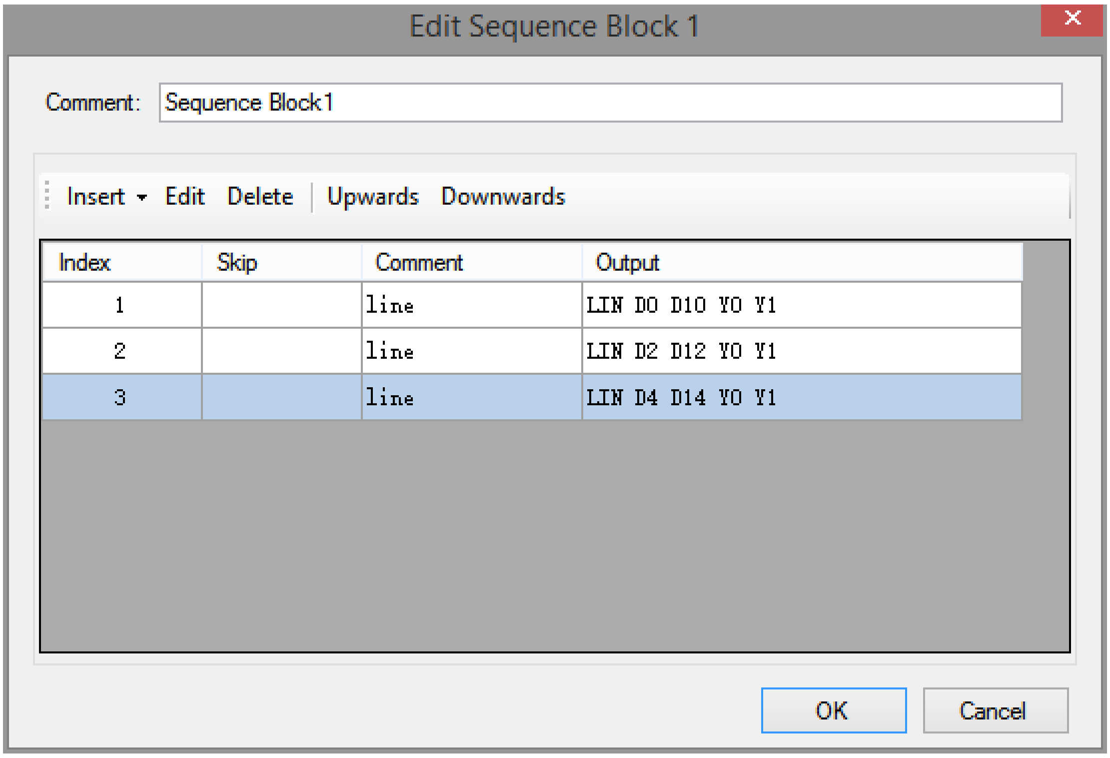
Explain:

The two axes are designated Y0 (Y axis) and Y1 (X axis). The corresponding directional terminals are Y4 and Y5. The coordinates of B point are (D0, D10), C point are (D2, D12), A point is (D4, D14), the speed is 1000Hz, and the acceleration and deceleration time are 50ms. The relevant parameters are set as follows:

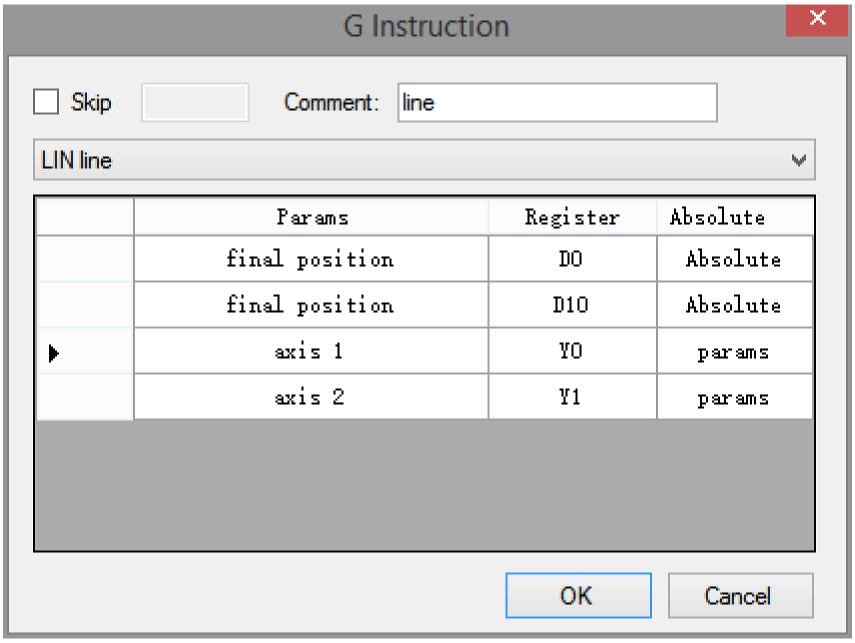
coordinates	X axis address	X axis setting value		Y axis address	Y axis setting value	
		absolute	relative		absolute	relative
B point	D0	3000	3000	D10	4000	4000
C point	D2	6000	3000	D12	0	-4000
A point	D4	0	-6000	D14	0	0
Default speed（Hz）		1000				
Acceleration/deceleration time（ms）		50				
X axis		Y0-pulse；Y4-direction				
Y axis		Y1-pulse；Y5-direction				

Program Ⅰ (absolute mode):

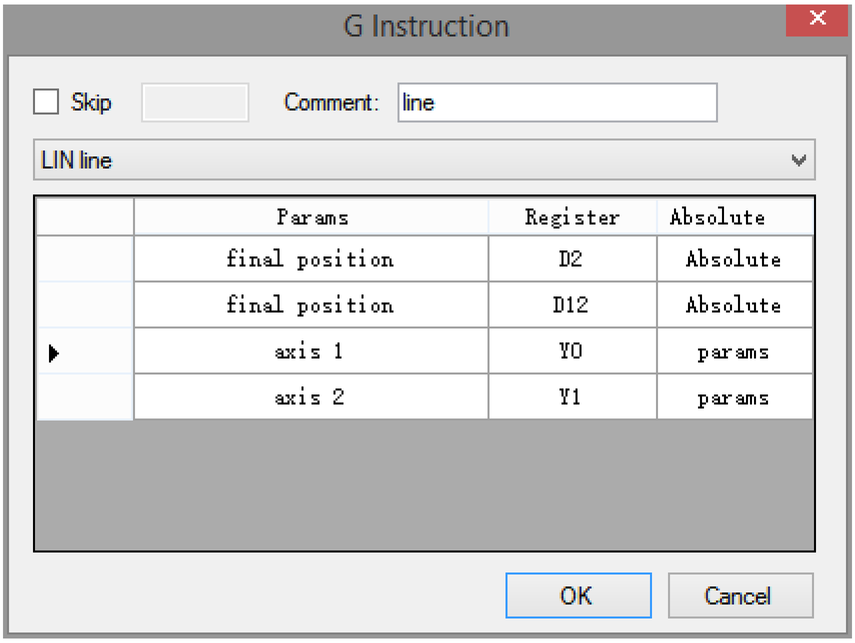
Add the G item in BLOCK, add three LIN instructions in it, as shown below:



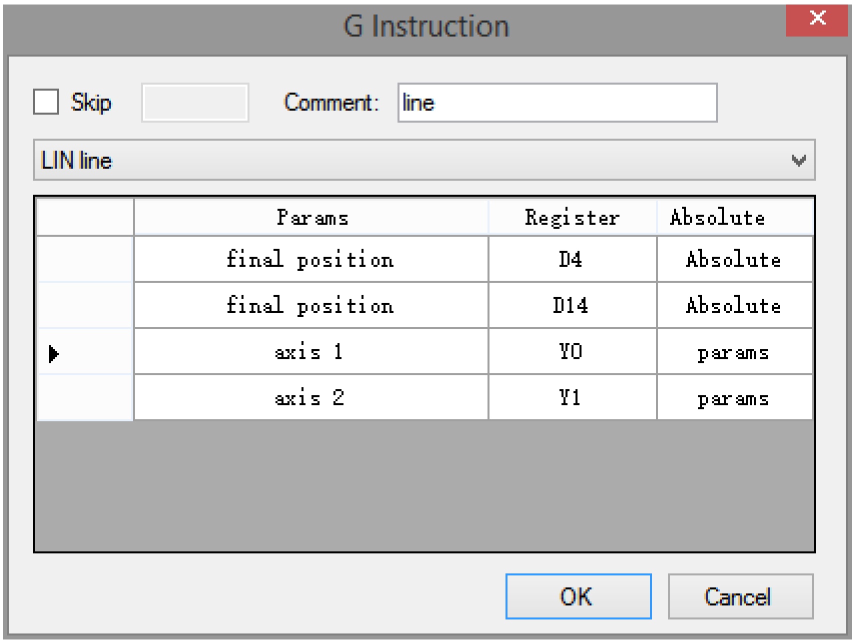
The configuration of the three instructions:



The first one (A→B)

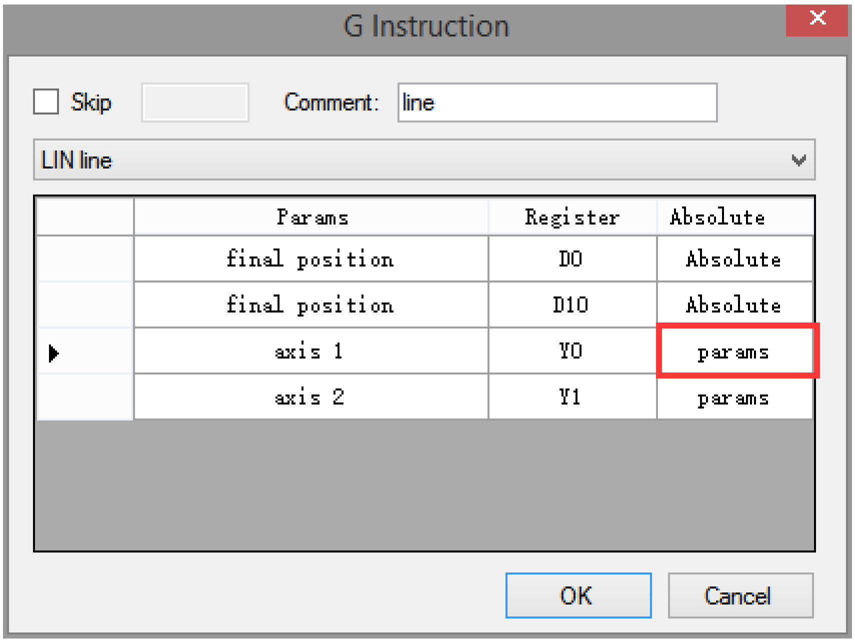


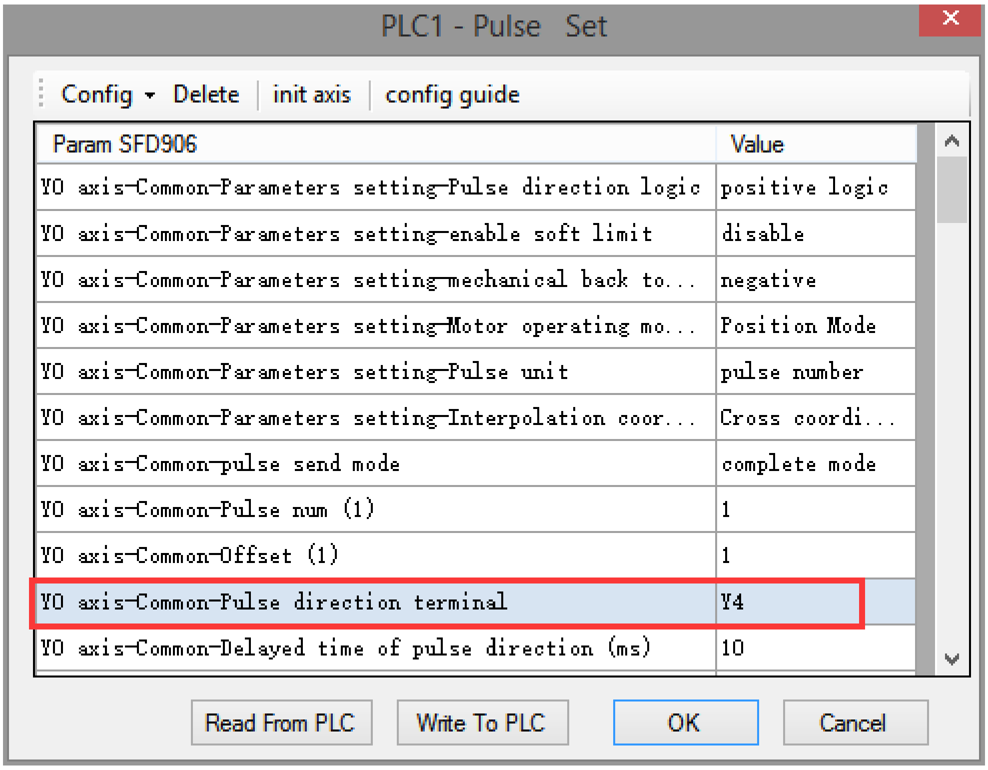
The second one (B→C)



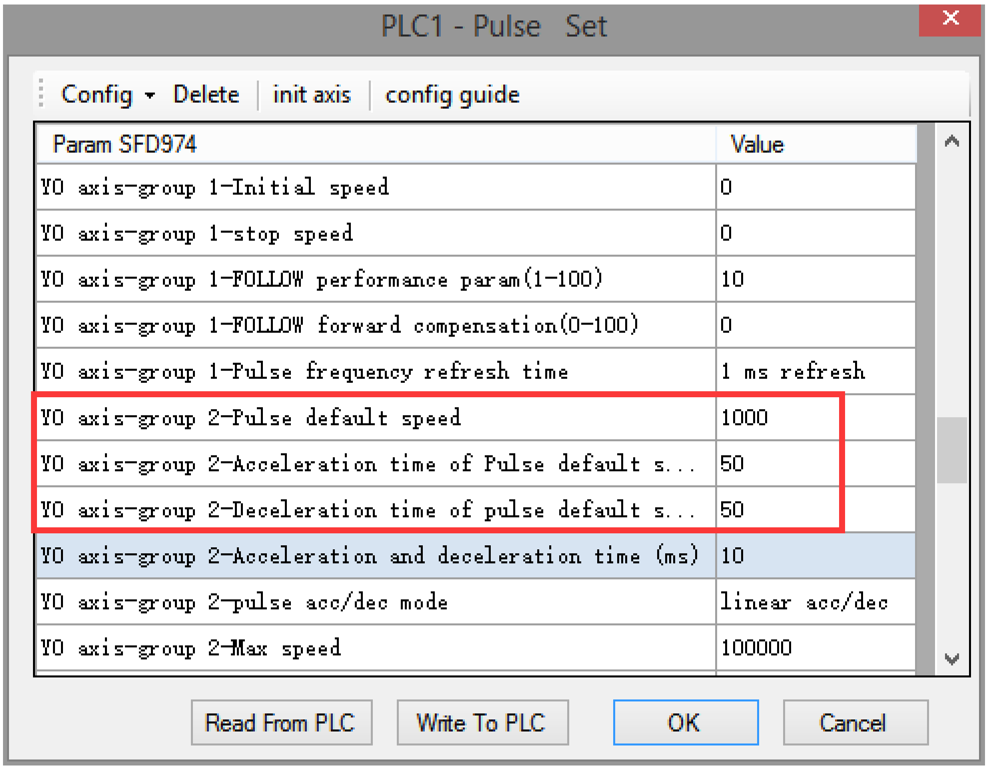
The third one (C→A)

Double click parameters, configure the Y0 axis parameters, as shown below:



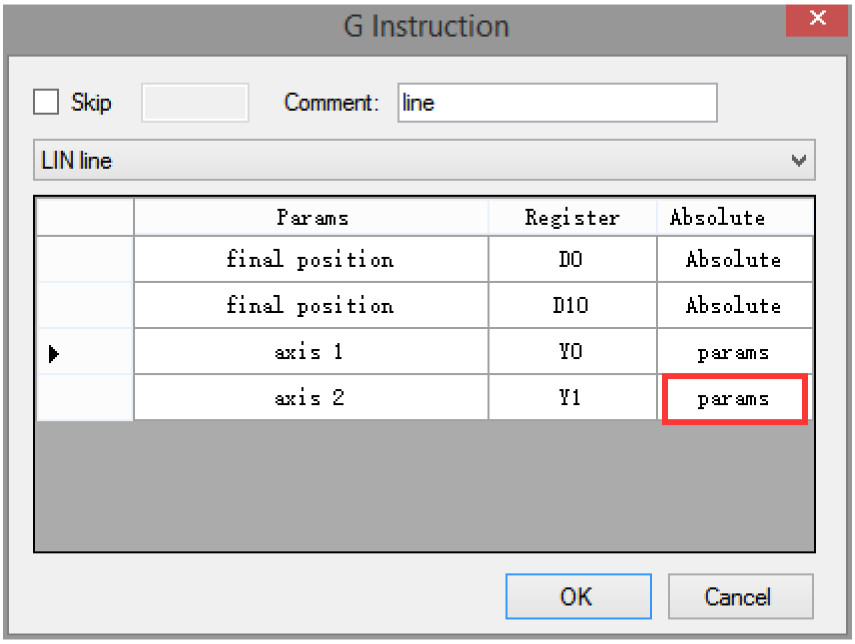


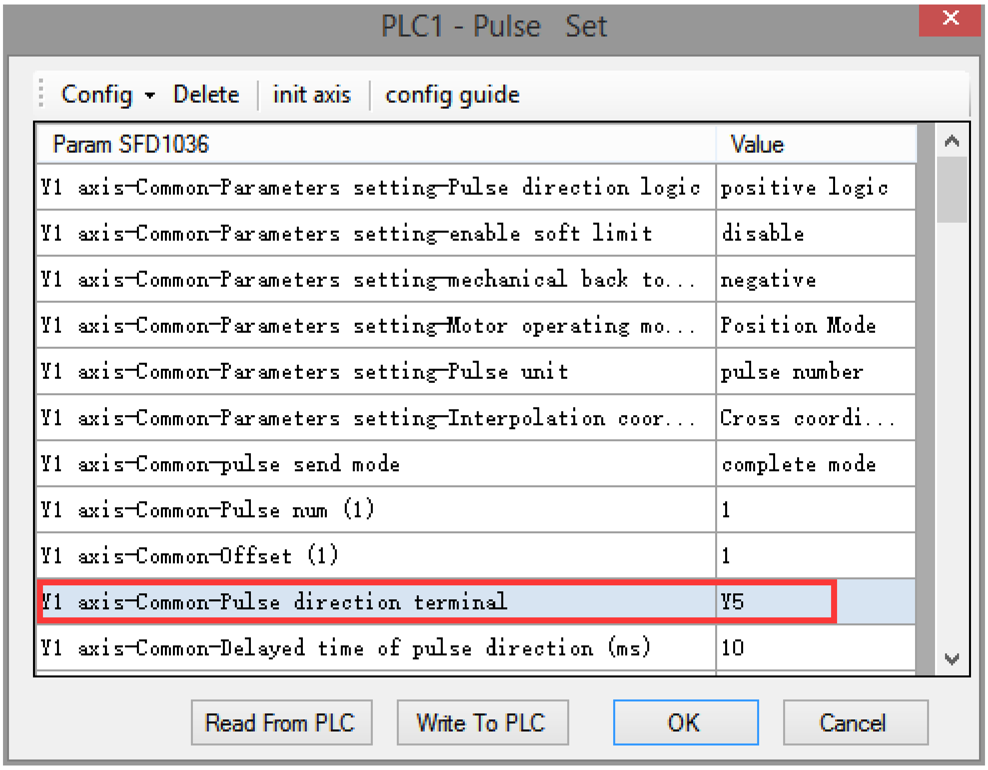
Y0 axis pulse direction terminal is set to Y4



Y0 axis pulse default speed is set to 1000, acc/dec time is 50ms

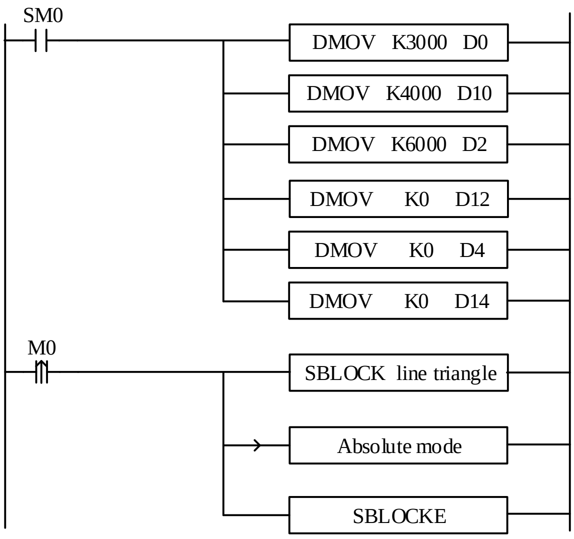
Double click parameters, configure the parameters of Y1 axis, as shown below:





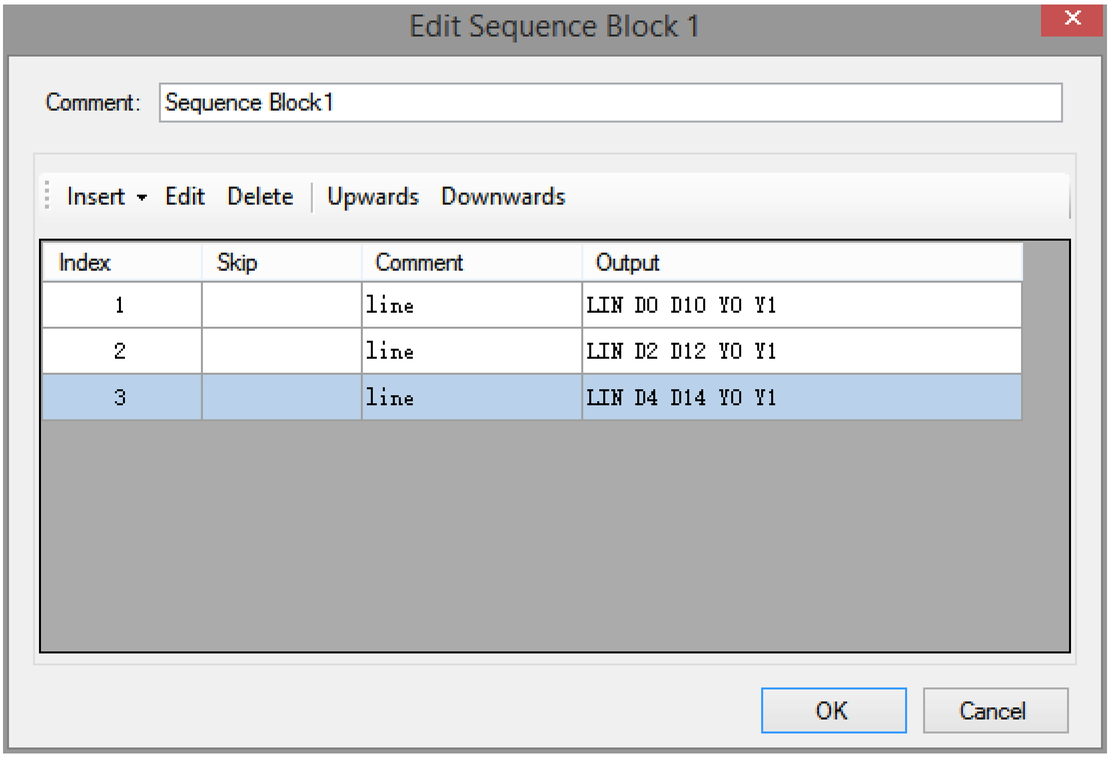
Y1 axis pulse direction terminal is set to Y5

After setting up, click OK to generate the program shown in the following figure in the ladder diagram. Write the set values in D0, D2, D4, D10, D12, D14. When M0 is turned on once, perform BLOCK once, and take a triangular route.

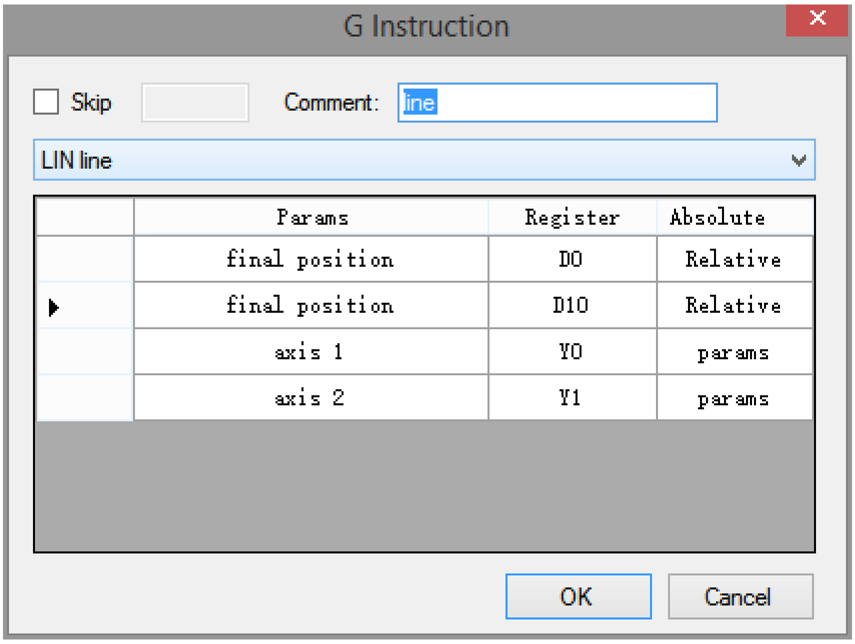


Program Ⅱ (relative mode):

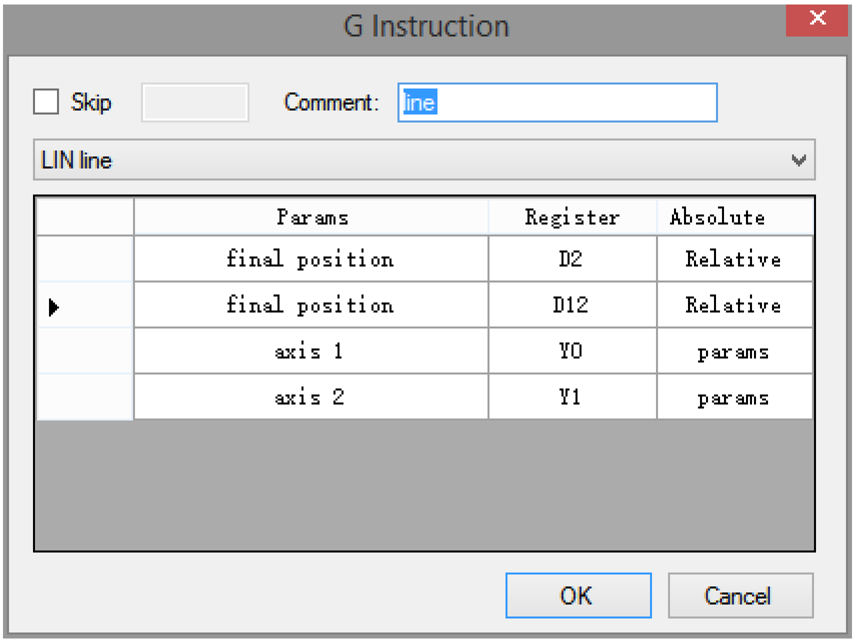
Three linear interpolation instructions [LIN] are added to the BLOCK by using the relative mode, as shown in the following figure:



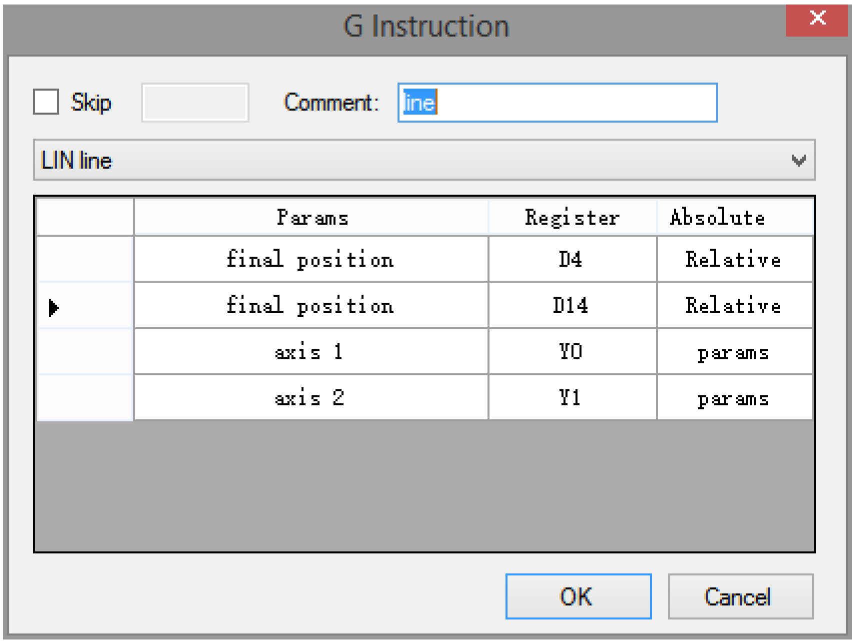
The three instructions are shown as below:



First one（A→B）



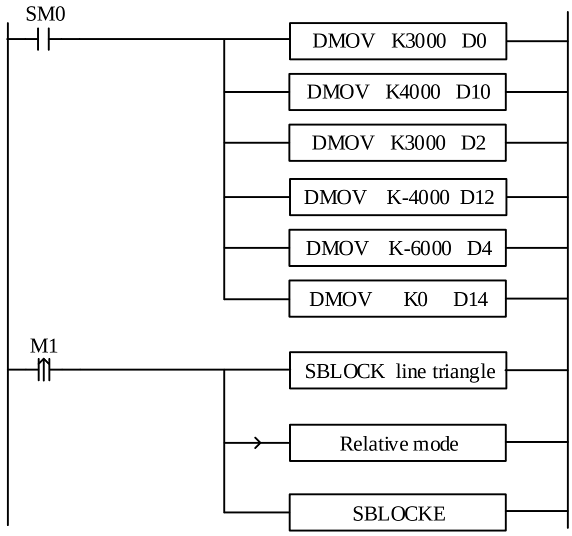
The second one（B→C）



The third one（C→A）

Double-click "parameters" to configure parameters of Y0 and Y1 axis [pulse direction terminal], [group 2 parameters - pulse default speed (Hz)], [group 2 parameters - pulse default speed acceleration time (ms)], [group 2 parameters - pulse default speed deceleration time (ms)] in the same absolute mode, which will not be described here.

After setting up, click OK to generate the program shown in the following figure in the ladder diagram. Assuming that the current values of HSD2 (double word) and HSD6 (double word) are all 0, the set values are written in D0, D2, D4, D10, D12 and D14. When M1 is set ON once, BLOCK is executed once, and a triangular line is taken.



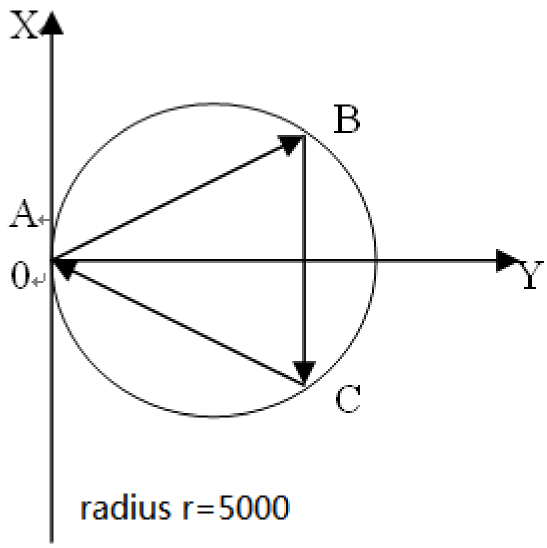
Note:

(1) The current position pulses of the two axes can be monitored by HSD2 (double word) and HSD6 (double word).

(2) The output terminals of the two axes correspond to Y0 and Y1 respectively, while the output terminals of the direction correspond to Y4 and Y5 respectively.

2-6-2. Circle + inscribed triangle

First step out of a circle with radius R = 5000 clockwise, and then follow the pattern of the inner regular triangle of the circle. The starting point is A (0, 0). First, follow the order of A (0, 0) →B (7500, 4285) →C (7500, - 4285) →A (0, 0) to form the circle, then from A(0, 0) to B (7500, 4285), and then from B (7500, 4285) to C(7500, -4285) points, and finally returns from C (7500, -4285) points to the starting point A (0, 0) and completes an inner regular triangle of a circle, as shown in the figure.



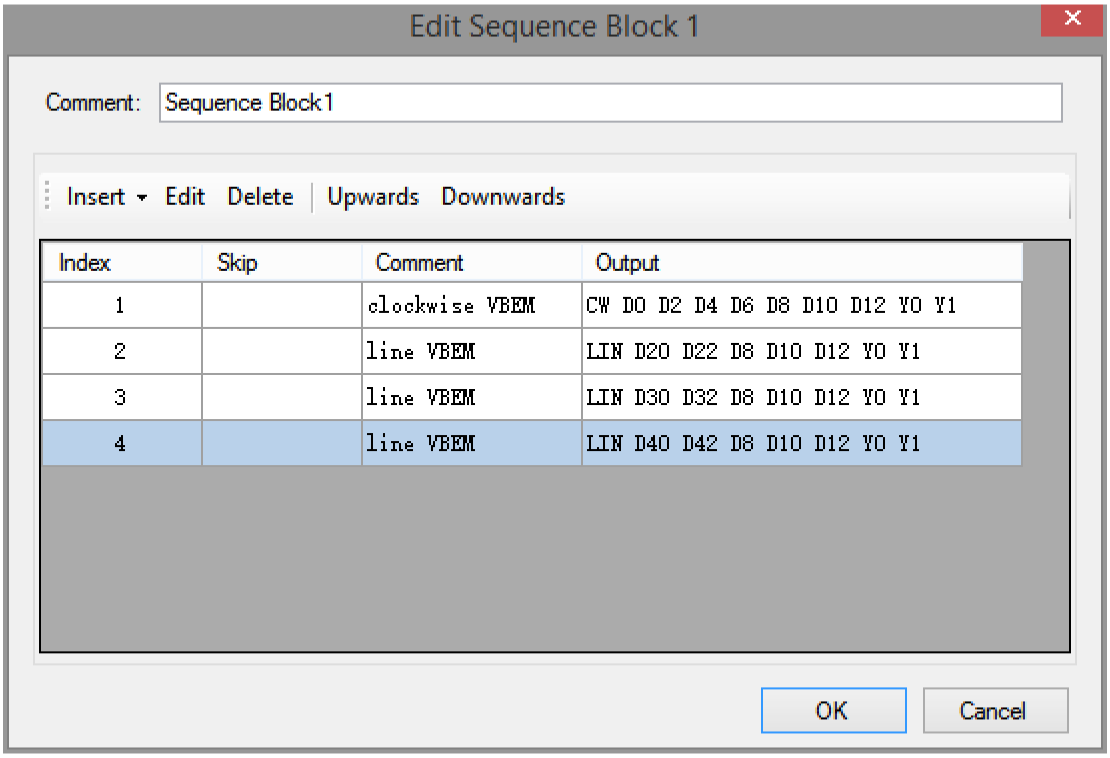
Note:

Two axes are designated as Y0 and Y1 axis, corresponding direction terminals are Y4 and Y5, B point coordinates are (D20, D22), C point coordinates are (D30, D32), A point coordinates are (D40, D42), starting speed is 50 Hz, stop speed is 50 Hz, maximum speed is 2000 Hz, default speed is 1000 Hz, acceleration and deceleration time is 50 ms, the specific parameters are set as follows:

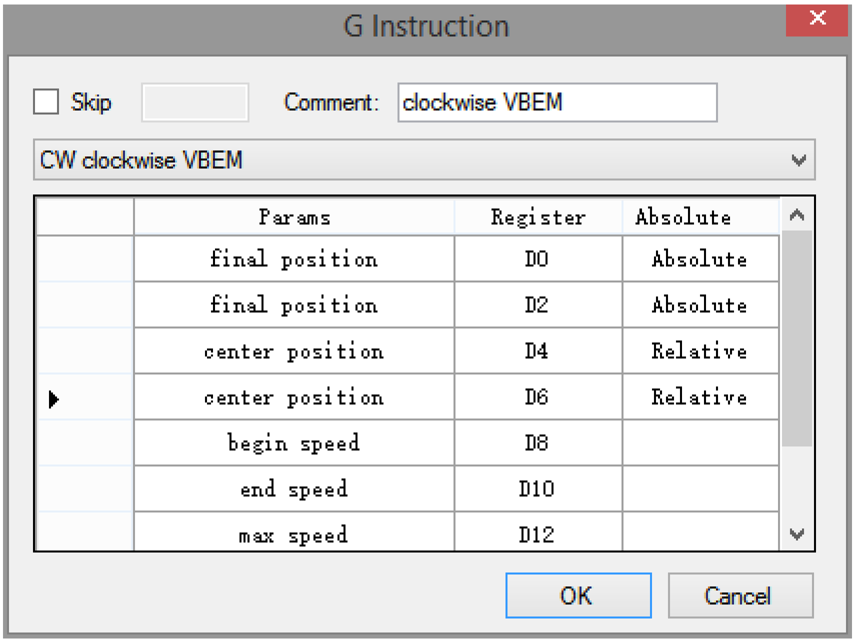
Function	Register or coil address	Value
Endpoint coordinates of circular arcs	D0	0
	D2	0
Center coordinates	D4	5000
	D6	0
B point coordinates	D20	7500
	D22	4285
C point coordinates	D30	7500
	D32	-4285
A point coordinates	D40	0
	D42	0
Starting speed（Hz）	D8	50
Stop speed（Hz）	D10	50
Max speed（Hz）	D12	2000
Default speed（Hz）	-	1000
Acc/dec time（ms）	-	50
X aixs	Y0 pulse, Y4 direction	
Y axis	Y1 pulse, Y5 direction	

Program (absolute mode):

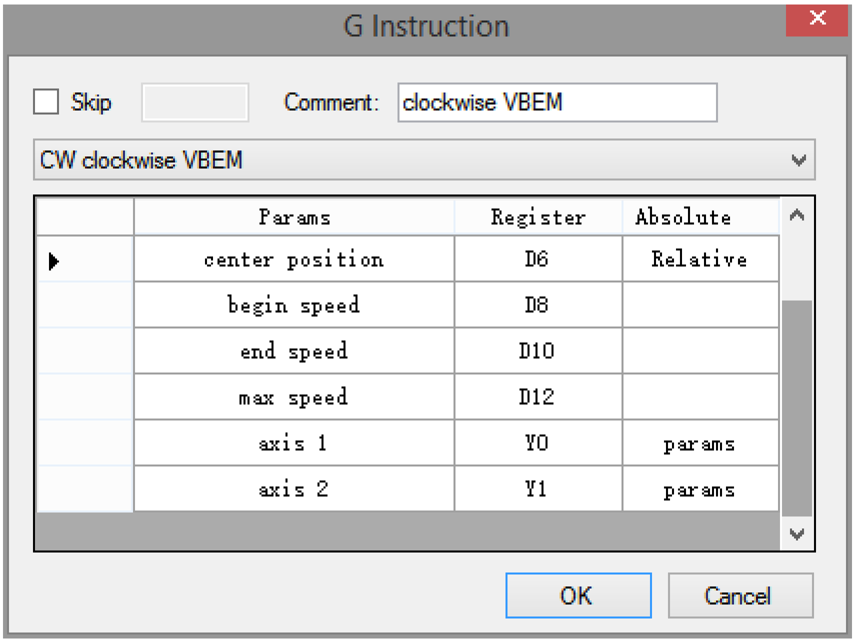
Because of the coincidence of the starting point and the end point, the command "CW clockwise arc VBEM" is chosen here, and the command "LIN line VBEM" is used in the triangle. Insert G instruction into BLOCK and write four interpolation instructions, as shown in the following figure:



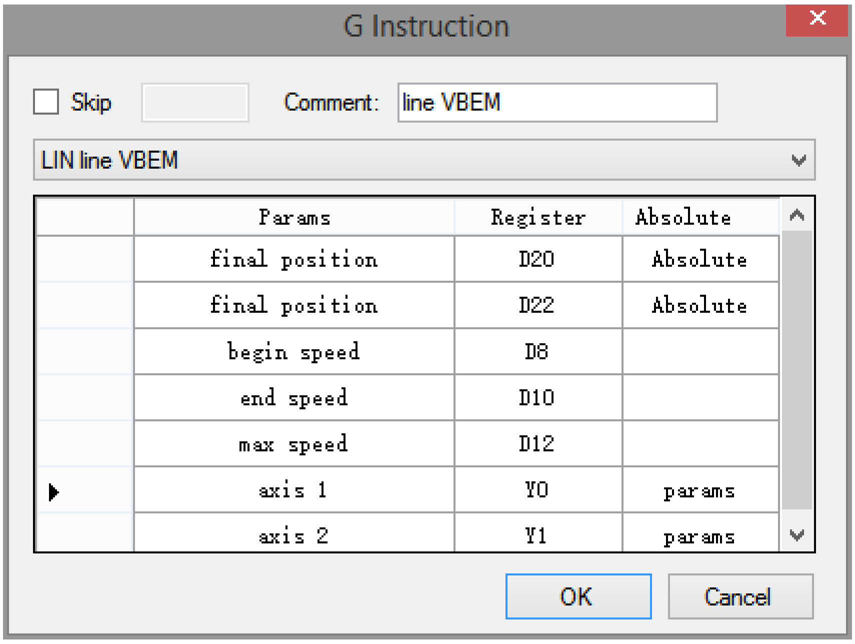
The four instructions are shown as below:



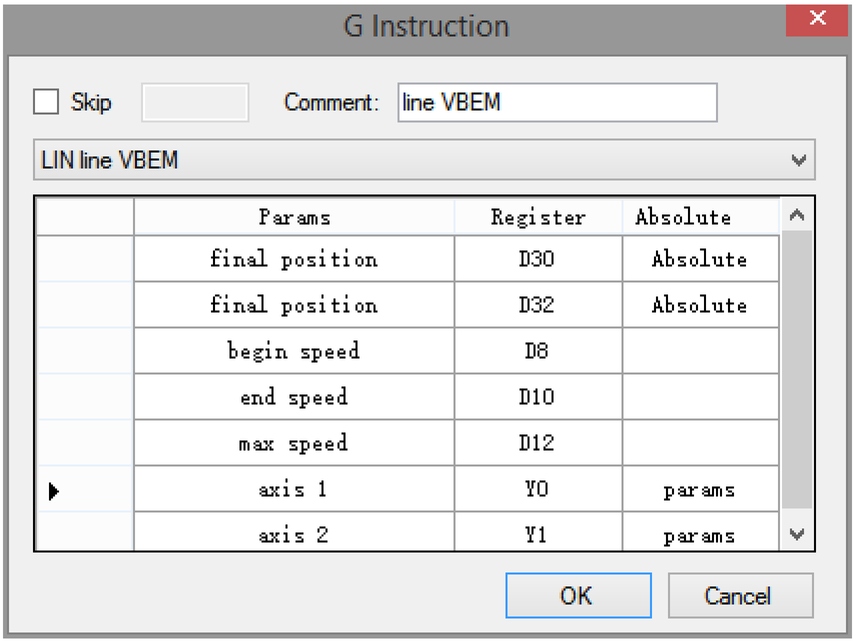
Instruction ① settings（1）



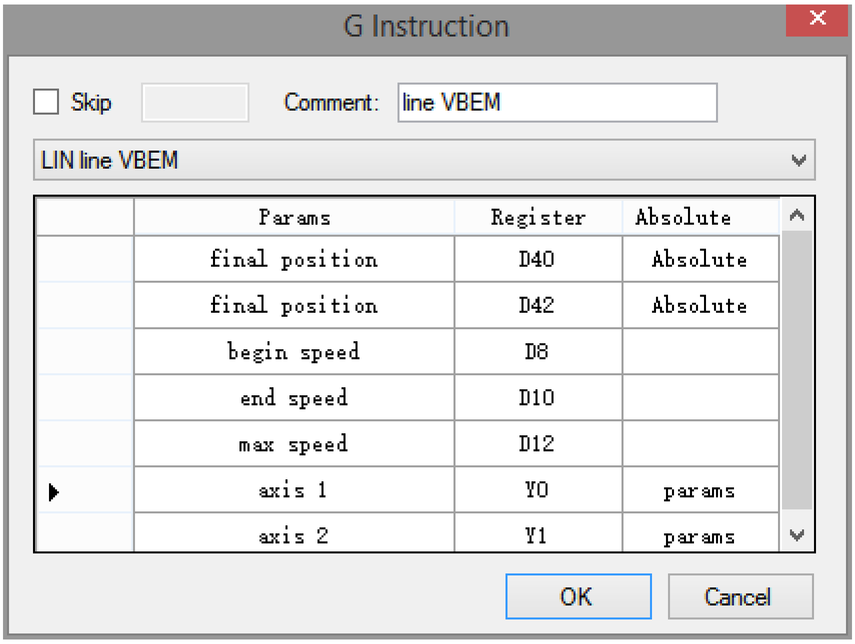
Instruction ① settings（2）



Instruction ② settings

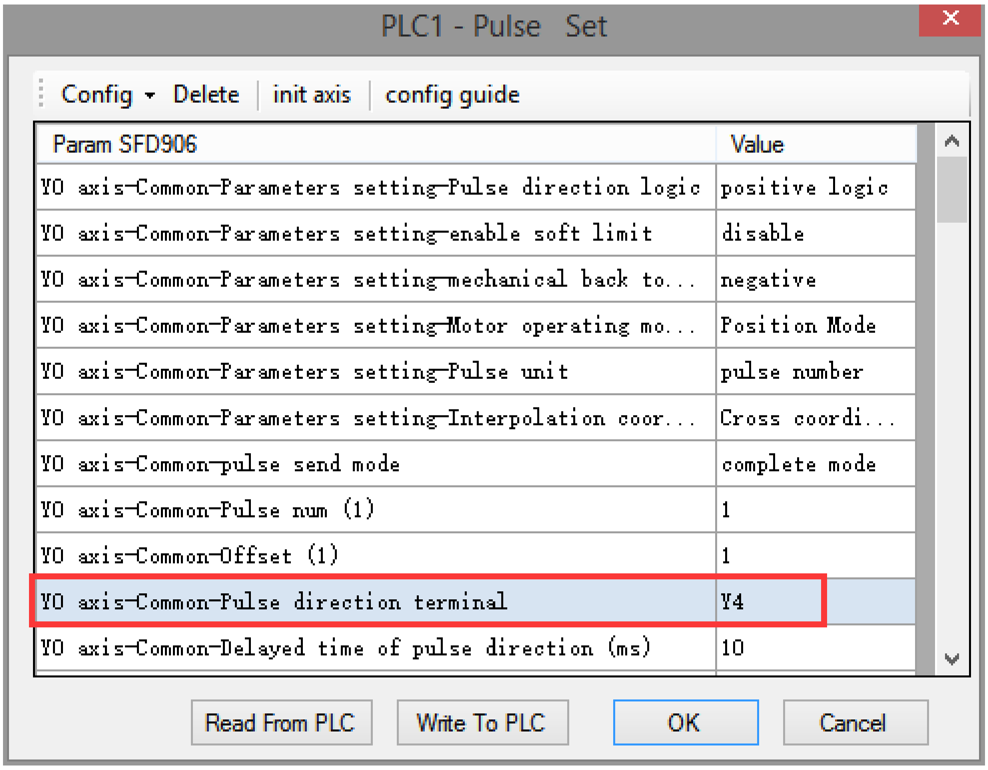


Instruction ③ settings

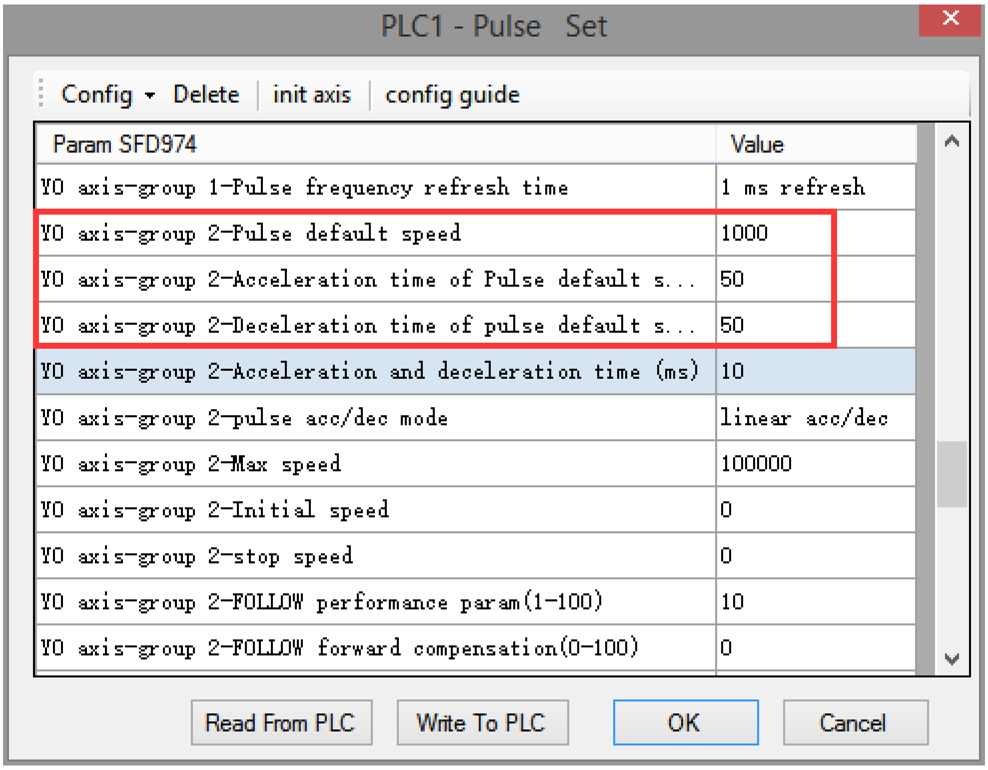


Instruction ④ settings

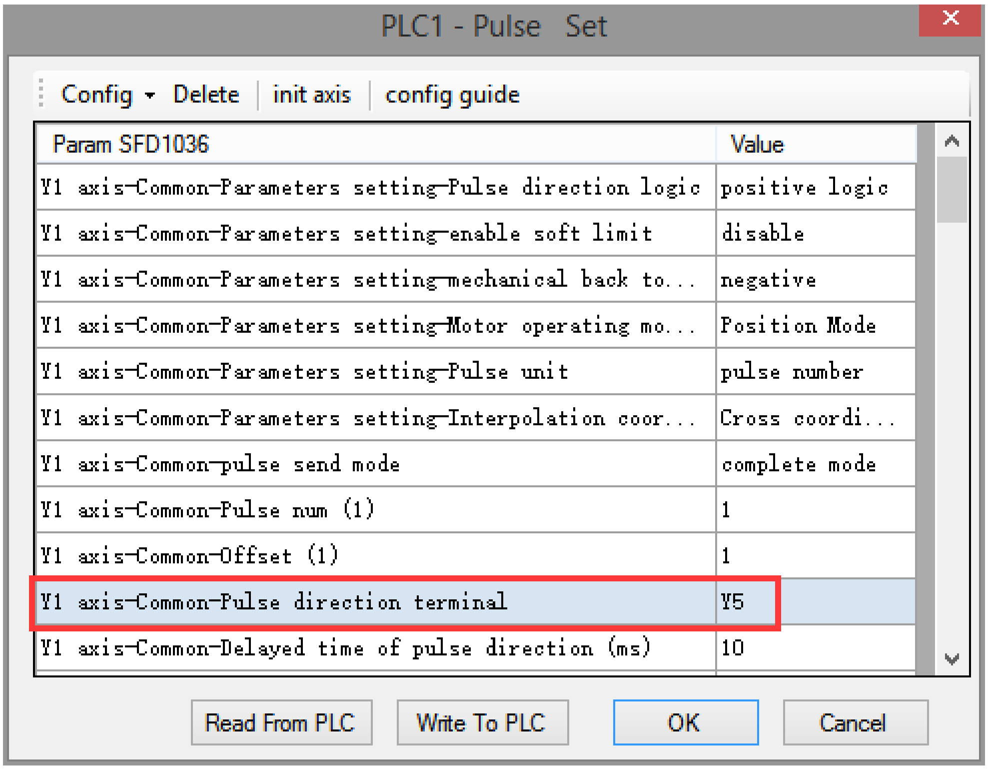
Double-click the "parameters" to configure the parameters of Y0 and Y1 axis [pulse direction terminal], [group 2 parameters - pulse default speed (Hz)], [group 2 parameters - pulse default speed acceleration time (ms)], [group 2 parameters - pulse default speed deceleration time (ms)], as follows:



Y0 axis settings（1）

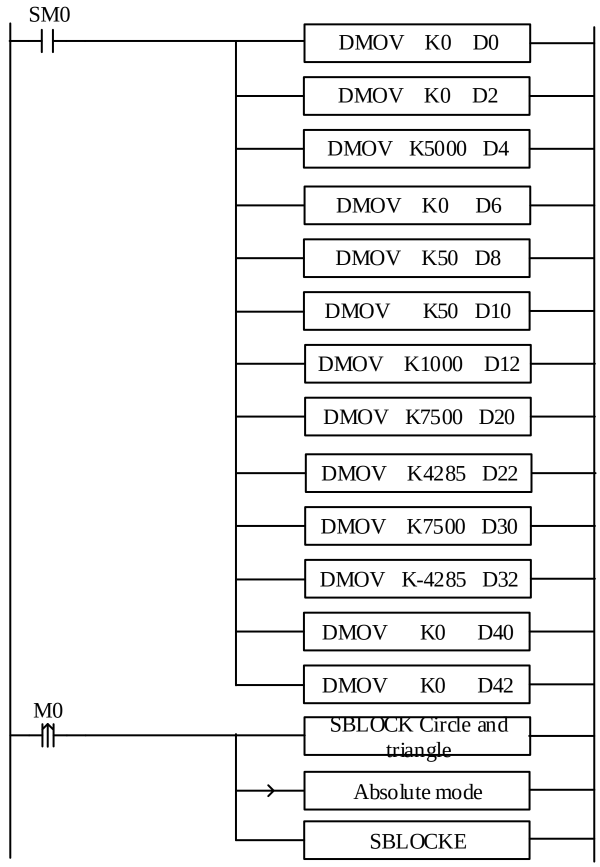


Y0 axis settings（2）



Y1 axis settings（1）

After setting up, click OK to generate the program shown in the following figure in the ladder diagram. Assuming that the current values of HSD2 (double-word) and HSD6 (double-word) are all 0, write the set values in the relevant registers. When M0 is turned on once, perform BLOCK once and take a triangle line once.



Note:

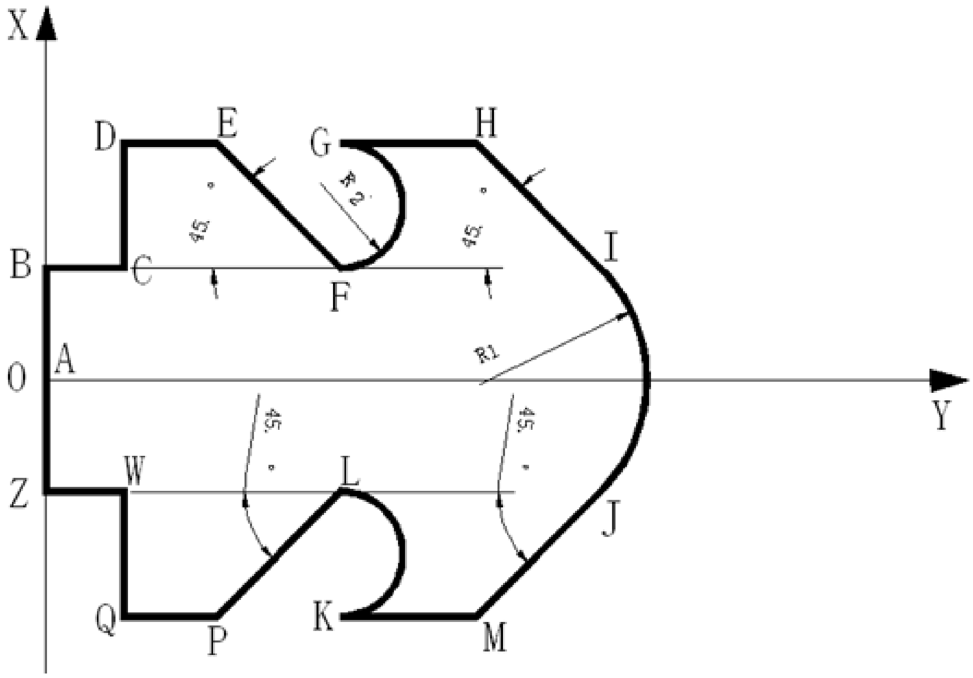
(1) The current position pulses of the two axes can be monitored by HSD2 (double word) and HSD6 (double word).

(2) The output terminals of the two axes correspond to Y0 and Y1 respectively, while the output terminals of the direction correspond to Y4 and Y5 respectively.

(3) When there are many points to go (if there are 1000 points), the ladder chart we write according to the above method will be very long, which is not conducive to the optimization of the program; therefore, we can use HMI to modify the values in the linear interpolation register to execute multiple linear interpolation instructions, in order to improve the readability of the program, optimize and reduce the scanning cycle of the program. The coordinates of each point can be set in the power-off retention register (the setting value of HMI register can be set by recipe function).

2-6-3. Line + Arc symmetric figure

As shown in following figure: starting from origin A (0, 0), and pass point B→C→D→E→F→G→H→I→J→M→K→L→P→Q→W→Z→A, the figure is symmetric with Y axis，AB=5000，BC=3000，CD=6000，DE=4000，R2=3000，GH=6000，R1=7070.



Note:

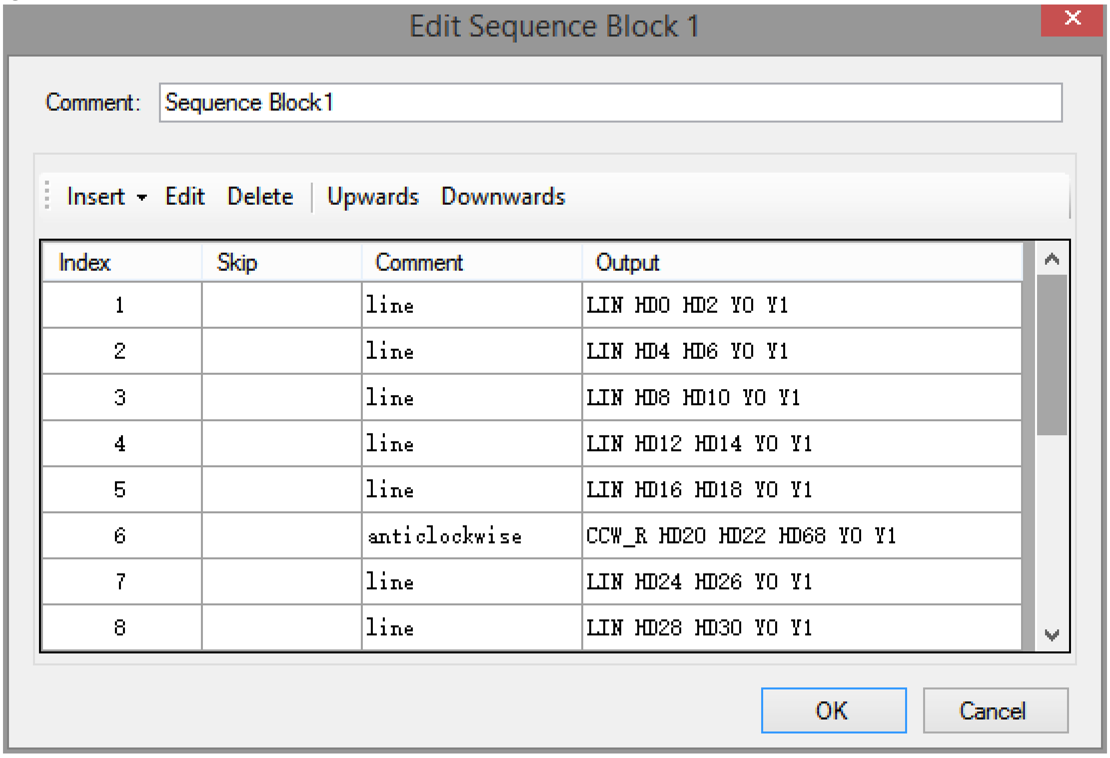
The two axes are designated as Y0 and Y1 axis, the corresponding directional terminals are Y4 and Y5, the default speed is 1000Hz, and the acceleration and deceleration time is 50ms, respectively. It is convenient to select the relative position mode according to the figure, so the specific parameters are set as follows:

Function	Address	Value (relative)	Function	Address	Value (relative)
B point coordinates	HD0	0	C point coordinates	HD4	3000
	HD2	5000		HD6	0
D point coordinates	HD8	0	E point coordinates	HD12	4000
	HD10	6000		HD14	0
F point coordinates	HD16	6000	G point coordinates	HD20	0
	HD18	-6000		HD22	6000
H point coordinates	HD24	6000	I point coordinates	HD28	6000
	HD26	0		HD30	-6000
J point coordinates	HD32	0	M point coordinates	HD36	-6000
	HD34	-10000		HD38	-6000
K point coordinates	HD40	-6000	L point coordinates	HD44	0
	HD42	0		HD46	6000
P point coordinates	HD48	-6000	Q point coordinates	HD52	-4000
	HD50	-6000		HD54	0

W point coordinates	HD56	0	Z point coordinates	HD60	-3000
	HD58	6000		HD62	0
A point coordinates	HD64	0	R2 radius	HD68	3000
	HD66	5000	R1 radius	HD70	7070
Default speed	1000Hz				
Acc/dec time	50ms				
X axis	Y0 pulse, Y4 direction				
Y axis	Y1 pulse, Y5 direction				

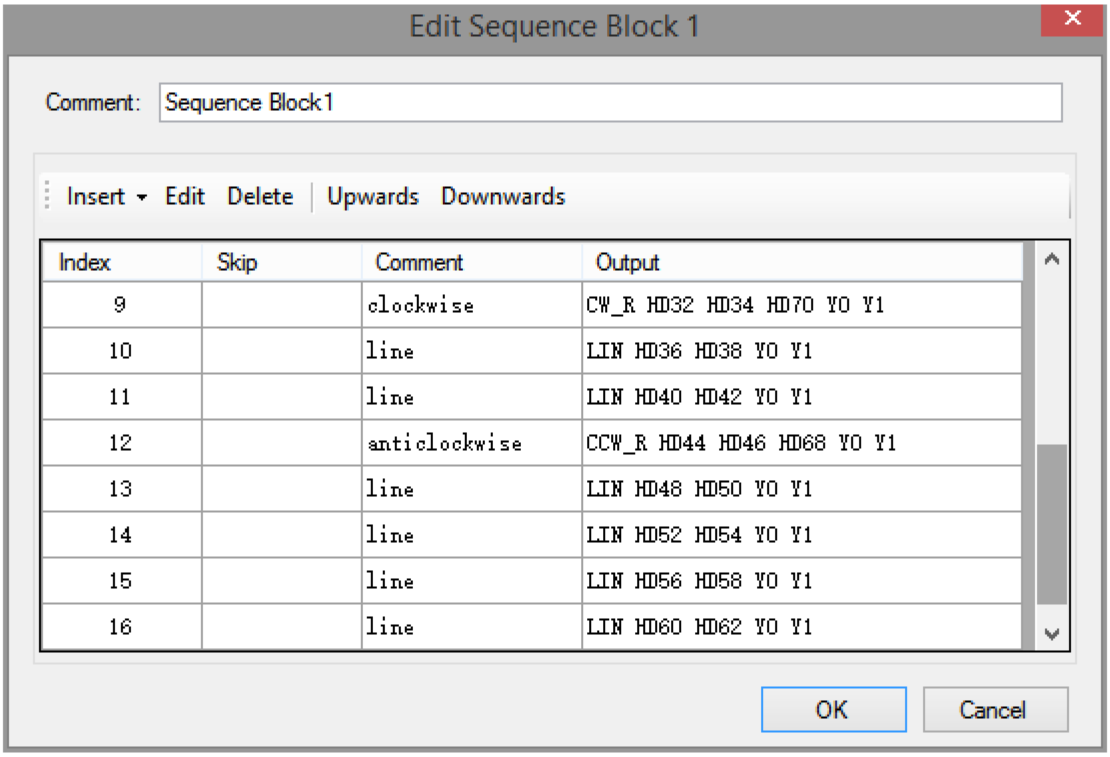
Program (relative mode):

Since the figure is mainly composed of straight lines and arcs, the "LIN line" instruction is chosen here, and the "CCW_R anticlockwise arc" and "CW_R clockwise arc" instruction are used for arcs. Insert G instruction into BLOCK and write 17 interpolation instructions, as shown in the following figure:

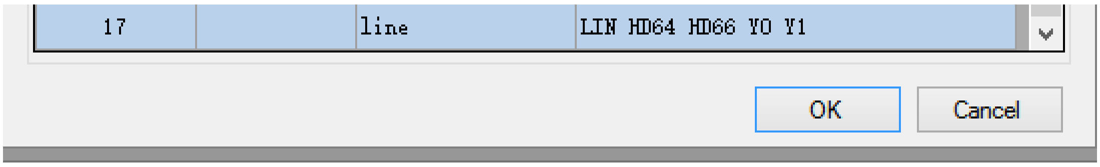


Index	Skip	Comment	Output
1		line	LIN HD0 HD2 Y0 Y1
2		line	LIN HD4 HD6 Y0 Y1
3		line	LIN HD8 HD10 Y0 Y1
4		line	LIN HD12 HD14 Y0 Y1
5		line	LIN HD16 HD18 Y0 Y1
6		anticlockwise	CCW_R HD20 HD22 HD68 Y0 Y1
7		line	LIN HD24 HD26 Y0 Y1
8		line	LIN HD28 HD30 Y0 Y1

instruction（1）~（8）

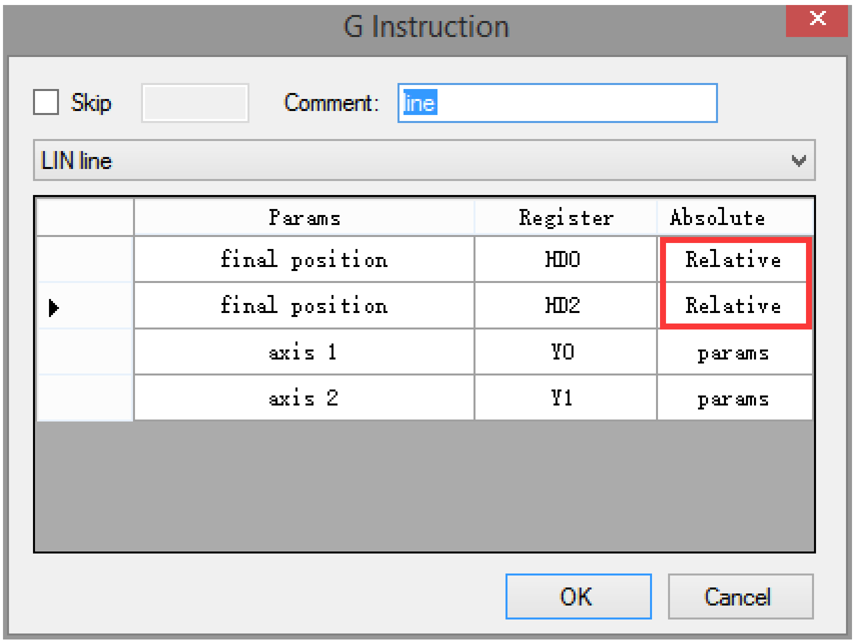


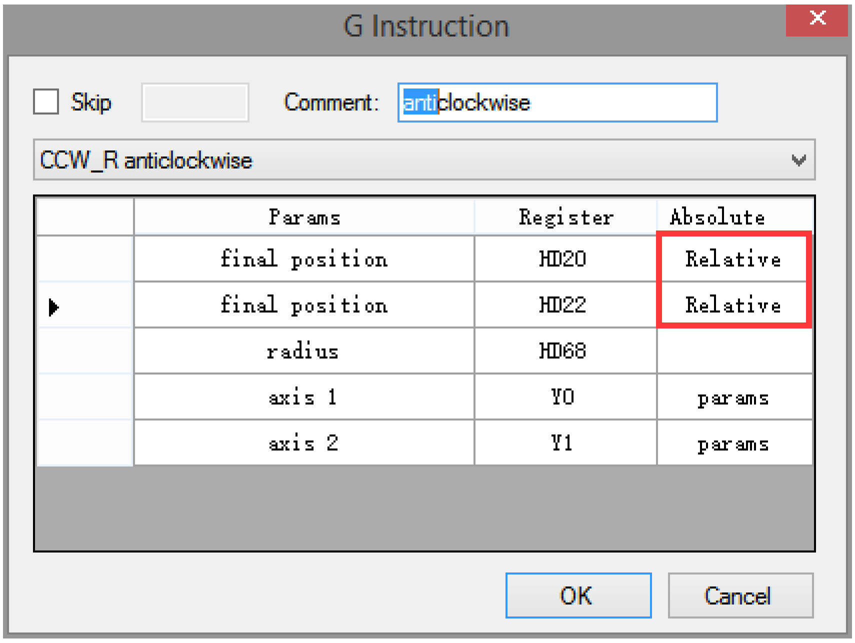
Instruction （9）~（16）

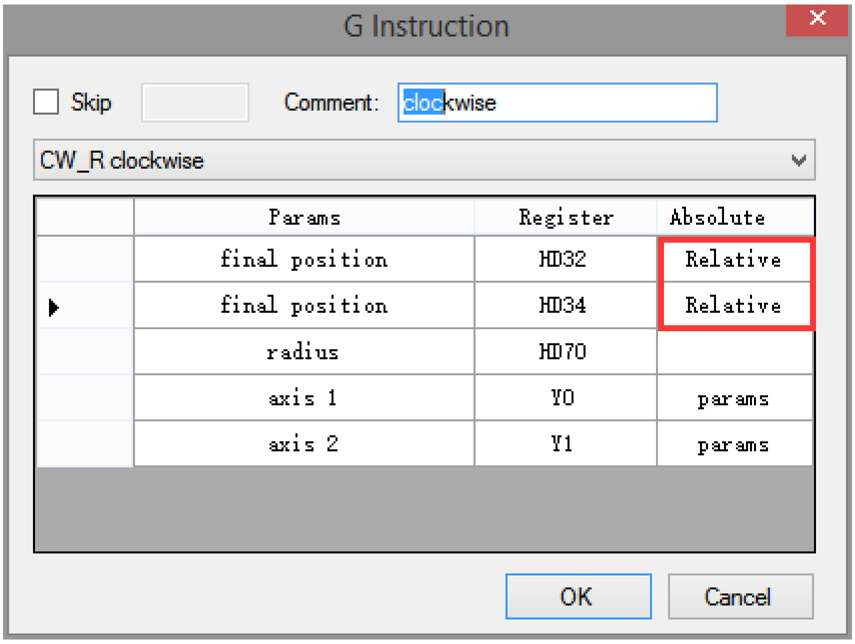


instruction（17）

The endpoint position of all the above instructions must be set to "relative mode", as shown in the following figure:

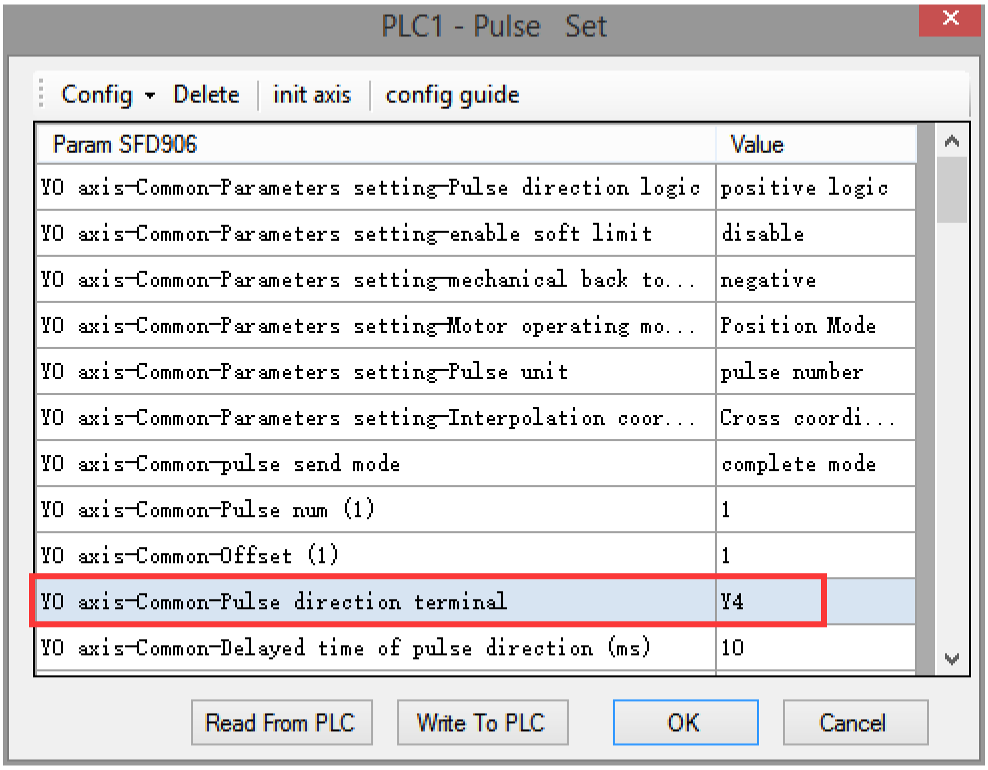




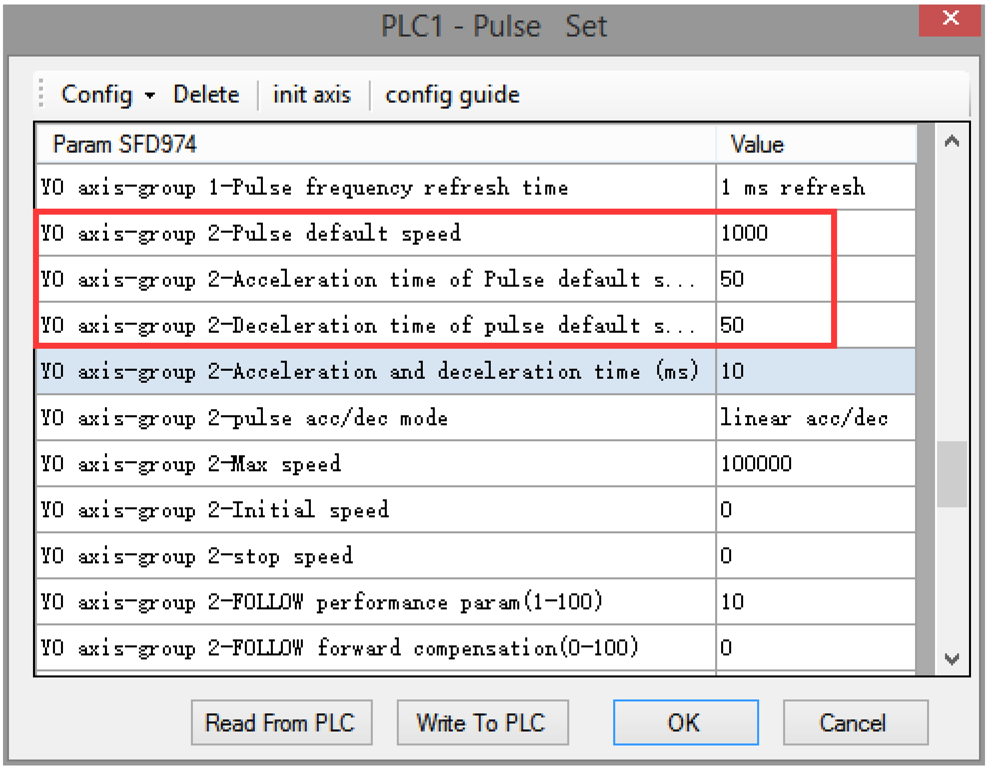


Note: The radius of the clockwise and anticlockwise arcs can only be absolute mode, and can not be modified!

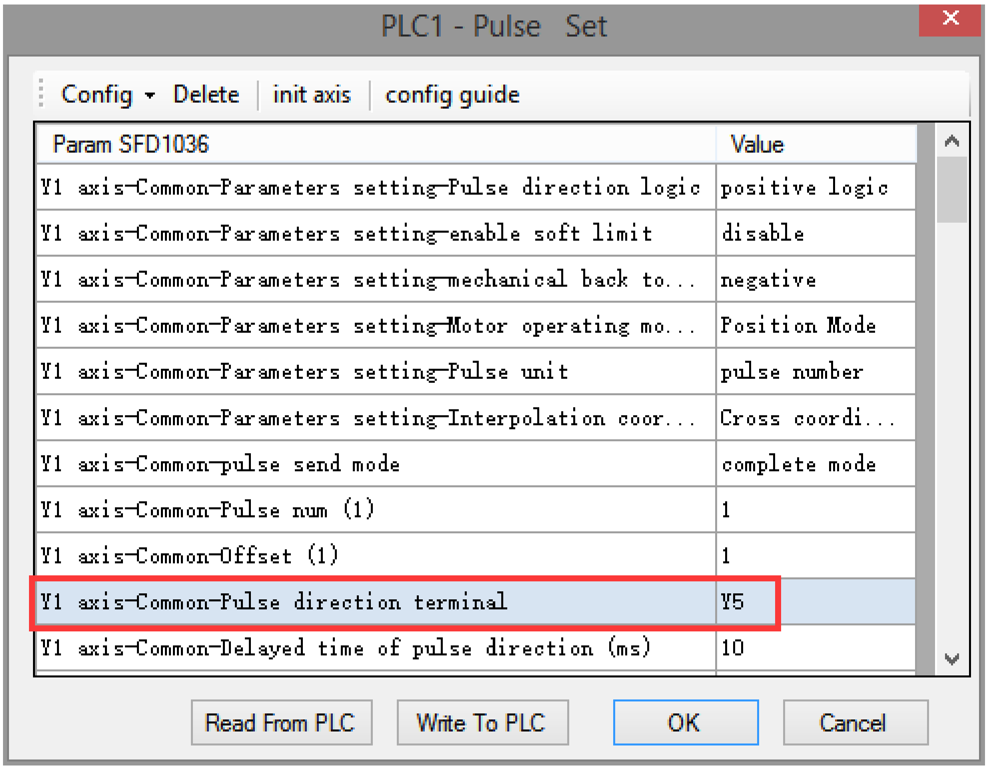
Double-click the "parameters" to configure the parameters of Y0 and Y1 axis [pulse direction terminal], [group 2 parameters - pulse default speed (Hz)], [group 2 parameters - pulse default speed acceleration time (ms)], [group 2 parameters - pulse default speed deceleration time (ms)], as follows:



Y0 axis settings（1）

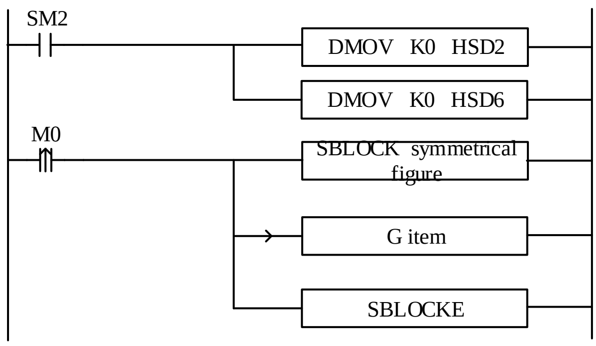


Y0 axis settings（2）



Y1 axis settings（1）

After setting up, click OK and write a complete program in the ladder diagram. As shown in the following figure, write the set value in the relevant register. When M0 is turned on once, execute BLOCK once, and walk the figure in this example once.



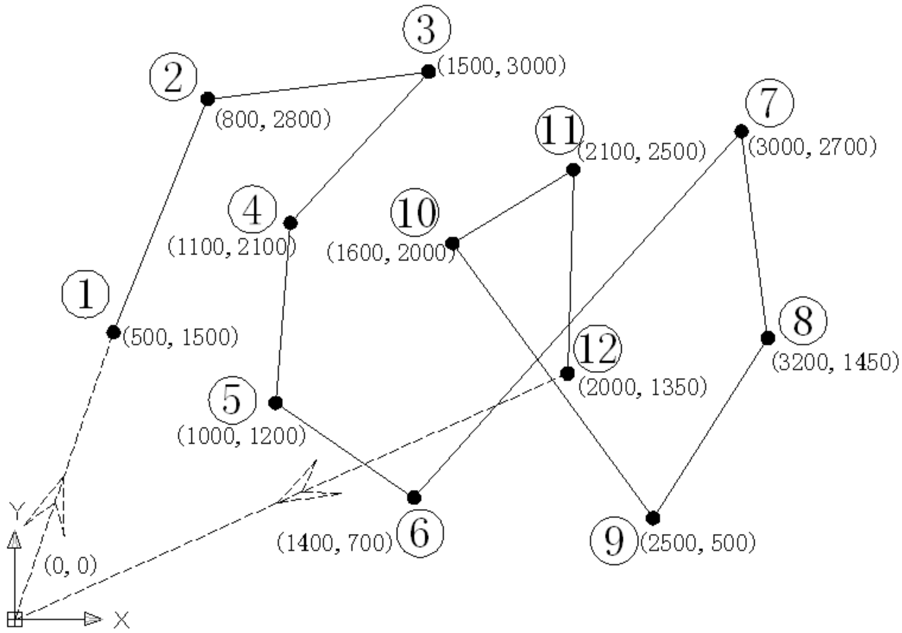
Note:

(1) The current position pulses of the two axes can be monitored by HSD2 (double word) and HSD4 (double word).

(2) The output terminals of the two axes correspond to Y0 and Y1 respectively, while the output terminals of the direction correspond to Y4 and Y5 respectively.

2-6-4. Disorder line segments

As shown in the figure, in the plane consisting of X-axis and Y-axis, the positioning of the equipment starts from the origin (0, 0), moves rapidly in the order of digital labeling (1-12) in the figure, and finally returns to the origin (0, 0) from the position of the 12th point (2000, 1350).



Note:

In this example, as the coordinates of each point are disorderly, so the lines connected sequentially by each point are slopes of arbitrary slope, so they can only be realized by the function of linear interpolation. From the graphics in the example, the coordinates of each point have been determined, so it is easier to choose absolute mode than relative mode.

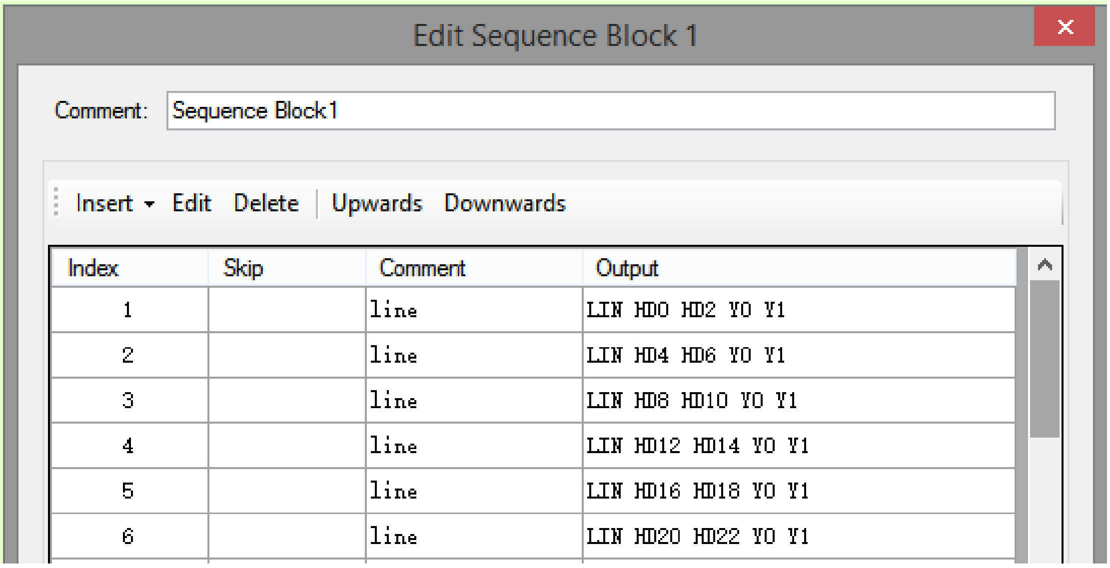
The two axes are designated Y0 (X axis) and Y1 (Y axis), the corresponding direction terminals are Y4 and Y5, the default speed is 1000Hz, the acceleration and deceleration time is 50ms, and all coordinate points are in absolute mode. Therefore, the specific parameters are set as follows:

Point	X axis address	X axis setting value(absolute)	Y axis address	Y axis setting value(absolute)
Point 1	HD0	500	HD2	1500
Point 2	HD4	800	HD6	2800
Point 3	HD8	1500	HD10	3000
Point 4	HD12	1100	HD14	2100
Point 5	HD16	1000	HD18	1200
Point 6	HD20	1400	HD22	700
Point 7	HD24	3000	HD26	2700
Point 8	HD28	3200	HD30	1450
Point 9	HD32	2500	HD34	500
Point 10	HD36	1600	HD38	2000
Point 11	HD40	2100	HD42	2500

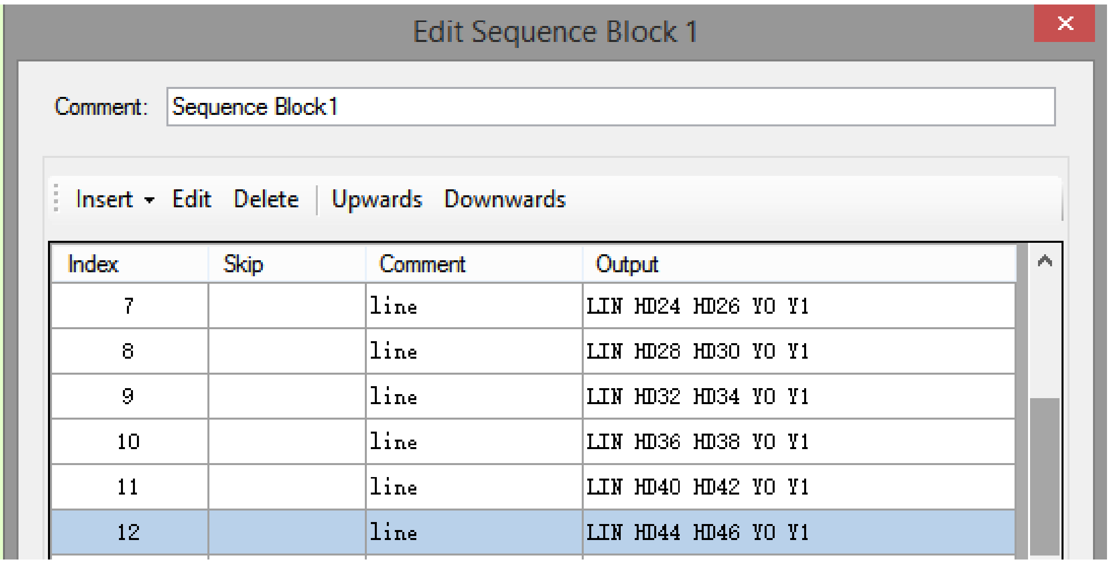
Point 12	HD44	2000	HD46	1350
Default speed（Hz）		1000		
Acc/dec time（ms）		50		
X axis		Y0-pulse；Y4-direction		
Y axis		Y1-pulse；Y5-direction		

Program (absolute mode):

Because the graphics are mainly composed of straight lines, the "LIN line" instruction is chosen here. Insert G instruction into BLOCK and write 12 interpolation instructions, as shown in the following figure:

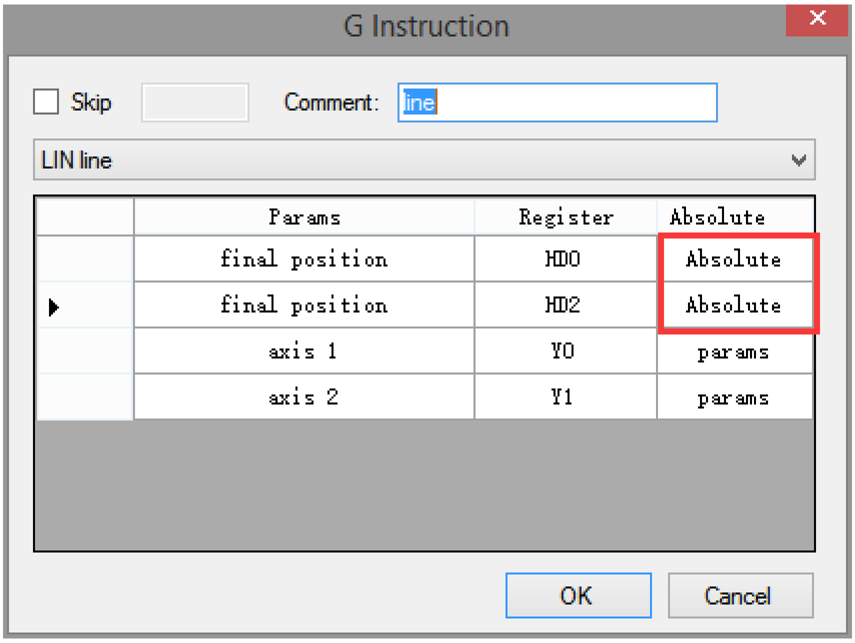


Instruction（1）~（6）

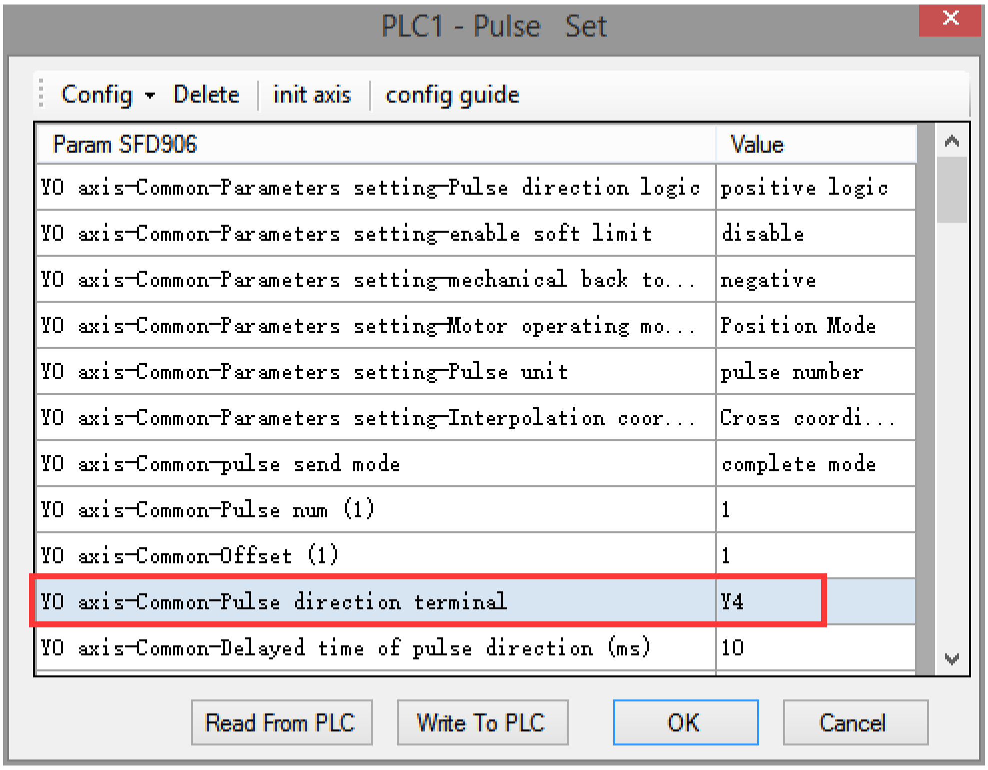


Instruction（7）~（12）

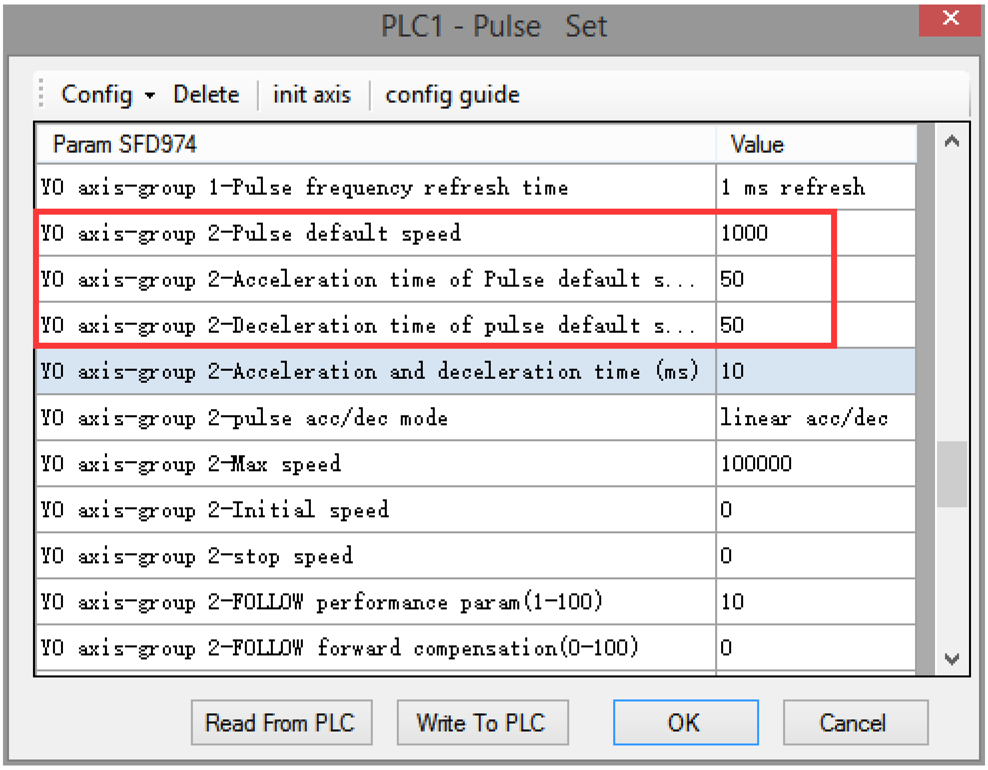
The endpoint position of all the above instructions must be set to "absolute mode", as shown in the following figure:



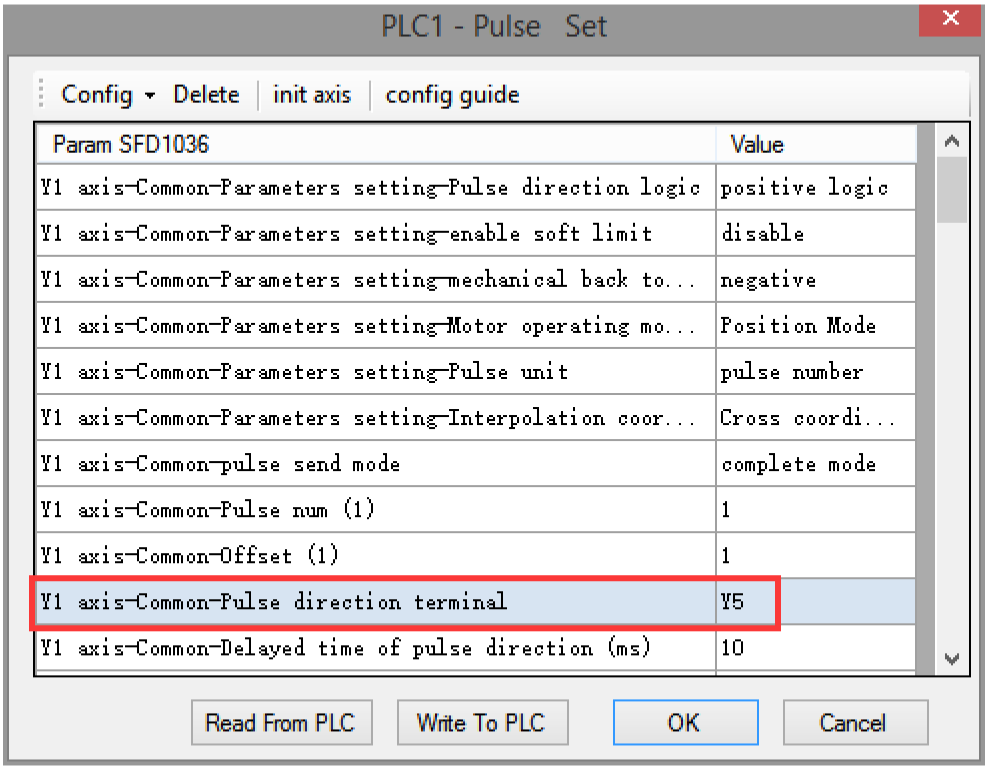
Double-click the "parameters" to configure the parameters of Y0 and Y1 axis [pulse direction terminal], [group 2 parameters - pulse default speed (Hz)], [group 2 parameters - pulse default speed acceleration time (ms)], [group 2 parameters - pulse default speed deceleration time (ms)], as follows:



Y0 axis settings（1）

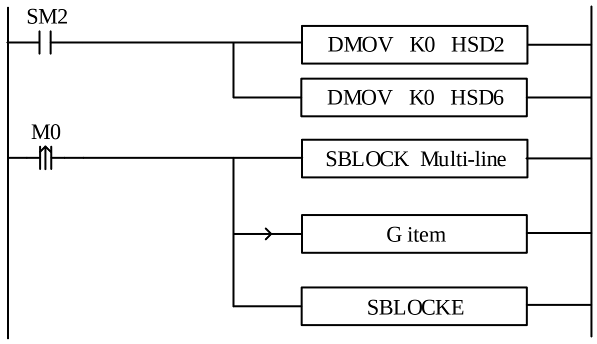


Y0 axis settings（2）



Y1 axis settings（1）

After setting up, click OK and write a complete program in the ladder diagram. As shown in the following figure, write the set value in the relevant register. When M0 is turned on once, execute BLOCK once, and walk the figure in this example once.



Note:

When there are many points to go (if there are 1000 points), the ladder chart we write according to the above method will be very long, which is not conducive to the optimization of the program; therefore, we can implement multiple linear interpolation instructions by modifying the values in the linear interpolation register to improve the readability, optimize and reduce the scanning cycle of the program. For example, the user can set the coordinates of each point in the power-off retentive register through the HMI, as shown in the following table:

Point	X axis register	X axis setting value	Y axis register	Y axis setting value
Point 1	D4000	500	D4100	1500
Point 2	D4002	800	D4102	2800
Point 3	D4004	1500	D4104	300
Point 4	D4006	1100	D4106	2100
Point 5	D4008	1000	D4108	200
Point 6	D4010	1400	D4110	700
Point 7	D4012	3000	D4112	2700
Point 8	D4014	3200	D4114	1450
Point 9	D4016	2500	D4116	500
Point 10	D4018	1600	D4118	2000
Point 11	D4020	2100	D4120	2500
Point 12	D4022	2000	D4122	1350

Note: HMI register setting value (can be set by HMI recipe function).

3 Application examples

In this chapter, some main instructions with more usage are introduced in depth in the form of program examples. These programs focus on pulse output instructions and motion control instructions.

3-1. Application of pulse output

Example: Now we are going to send three consecutive pulses, the pulse terminal is Y0 and the pulse direction terminal is Y2. The pulse frequency, pulse number and acceleration and deceleration of each segment are shown in the table below.

Pulse	Frequency setting value（Hz）	Pulse number setting value
Segment 1	3000	1000
Segment 2	800	2000
Segment 3	6000	8000
Acc/dec time	Frequency changes 1000Hz every 100ms	

Pulse data address assignment is as follows:

Address	Notes	Value
HD0 (double word)	Pulse total segments (1 to 100)	3
HD2 (8 words)	Reserved	0
HD10 (double words)	Pulse frequency (#1)	3000
HD12 (double word)	Pulse number (#1)	1000
HD14	bit15~bit8: waiting condition (#1) H00: pulse sending completion H01: wait time H02: wait signal H03: ACT time H04: EXT signal H05: EXT signal or pulse sending completion bit7~bit0: waiting condition register type H00: constant H01: D	0

	H02: HD H03: FD H04: X H05: M H06: HM	
HD15 (double word)	Constant value/ register no. (for waiting condition)(#1)	0
HD17	bit7~bit0: jump register type H00: constant value H01: D H02: HD H03: FD	0
HD+18 (double word)	Constant value/register no. (for jump register)(#1)	0
HD+20 (double word)	Pulse frequency (#2)	800
HD+22 (double word)	Pulse number (#2)	2000
HD+24	Waiting condition, waiting condition register type (#2)	0
HD+25 (double word)	Constant value or register no. (for waiting condition) (#2)	0
HD+27	Jump type, jump register type (#2)	0
HD+28 (double word)	Constant value or register no. (for jump register) (#2)	0
HD+30 (double word)	Pulse frequency (#3)	6000
HD+32 (double word)	Pulse number (#3)	8000
HD+34	Waiting condition, waiting condition register type (#3)	0
HD+35 (double word)	Constant value or register no. (for waiting condition) (#3)	0
HD+37	Jump type, jump register type (for waiting condition) (#3)	0
HD+38 (double word)	Constant value or register no. (for jump register) (#3)	0

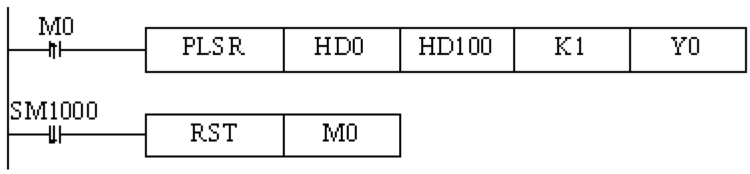
System parameters

SFD900	Pulse parameter setting	Bit 1: pulse direction logic 0: positive logic 1: negative logic, default is 0 Bit 2: use soft limit function 0: not use 1: use default is 0 Bit 3: mechanical return to origin direction 0: negative direction 1: positive direction default is 0 Bit 10~8: pulse unit Bit8: 0: pulse number 1: equivalent 000: pulse number 001: 1 um 011: 0.01mm 101: 0.1mm 111: 1 mm Default is 000 Bit15: interpolation coordinate mode 0: cross coordinate 1: polar coordinate Default is 0	0	Common parameter
SFD901	Pulse sending mode	Bit 0: pulse sending mode 0: complete mode 1: subsequence mode, default is 0	0	
SFD902	Pulse number/1 rotation low 16 bits		0	
SFD903	Pulse number/1 rotation high 16 bits		0	
SFD904	Motion quantity/1 rotation low 16 bits		0	
SFD905	Motion quantity/1 rotation high 16 bits		0	
SFD906	Pulse direction terminal	Y terminal no., 0xFF is no terminal	2	
SFD907	Direction delay time	Default is 20, unit: ms	20	
SFD908	Gear clearance positive compensation		0	
SFD909	Gear clearance negative compensation		0	
SFD910	Electrical origin low 16 bits		0	
SFD911	Electrical origin high 16 bits		0	

SFD912	Signal terminal state setting	Bit0: origin signal switch state Bit1: Z phase switch state Bit2: positive limit switch state Bit3: negative limit switch state 0: normally open(positive logic) 1: normally close(negative logic) default is 0	0	
SFD913	Close point signal	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD914	Z phase terminal setting	Bit0~bit7: set X terminal, 0xFF is no terminal(interruption)	0xFF	
SFD915	Limit terminal setting	Bit7~bit0: X terminal of positive limit, 0xFF is no terminal Bit15~bit8: X terminal of negative limit, 0xFF is no terminal	FFFF	
SFD917	Clear signal CLR output terminal	Bit0~Bit7: Y terminal, 0xFF is no terminal	0xFF	
SFD918	Returning speed VH low 16 bits		0	
SFD919	Returning speed VH high 16 bits		0	
SFD922	Crawling speed VC low 16 bits		0	
SFD923	Crawling speed VC high 16 bits		0	
SFD924	Mechanical origin position low 16 bits		0	
SFD925	Mechanical origin position high 16 bits		0	
SFD926	Z phase numbers		0	
SFD927	CLR signal delay time	Default 20, unit: ms	20	
SFD928	Grinding wheel radius(polar coordinate)	Low 16 bits	0	
SFD929		High 16 bits	0	
SFD930	Soft limit positive limit value	Low 16 bits	0	
SFD931		High 16 bits	0	
SFD932	Soft limit negative limit value	Low 16 bits	0	
SFD933		High 16 bits	0	
…				
SFD950	Pulse default speed low 16 bits	It will send pulse with default speed when the speed is 0.	1000	Group 1
SFD951	Pulse default speed high 16 bits		0	

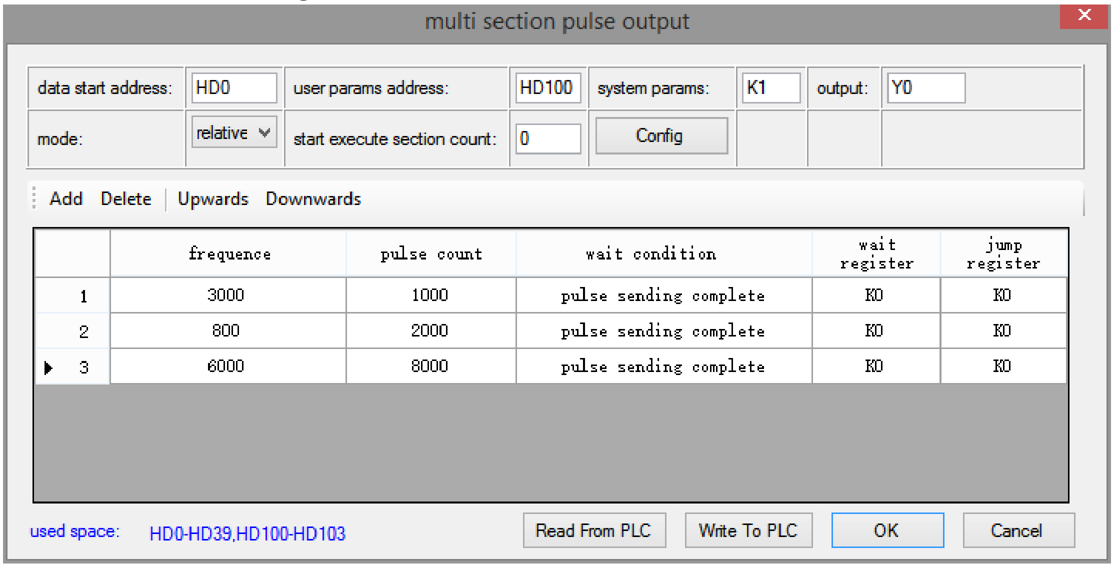
SFD952	Pulse default speed acceleration time		100	
SFD953	Pulse default speed deceleration time		100	
SFD954	Acceleration and deceleration time		0	
SFD955	Pulse acceleration and deceleration mode	Bit 1~0: acc/dec mode 00: line 01: S curve 10: sine curve 11: reserved Bit 15~2: reserved		
SFD956	Max speed limit low 16 bits		3392	
SFD957	Max speed limit high 16 bits		3	
SFD958	Initial speed low 16 bits		0	
SFD959	Initial speed high 16 bits		0	
SFD960	Stop speed low 16 bits		0	
SFD961	Stop speed high 16 bits		0	
SFD962	Follow performance parameters	1~100, 100 means the time constant is one tick, 1 means the time constant is 100 tick.		
SFD963	Follow feedforward compensation	0~100, percentage		
…				

Pulse instruction:

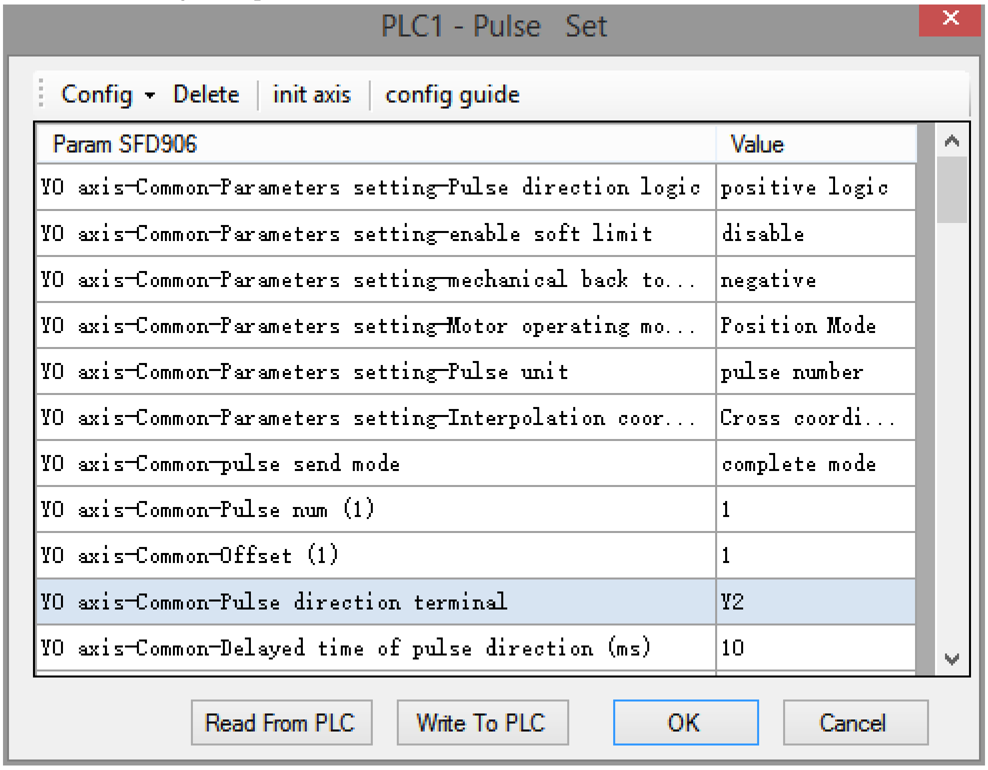


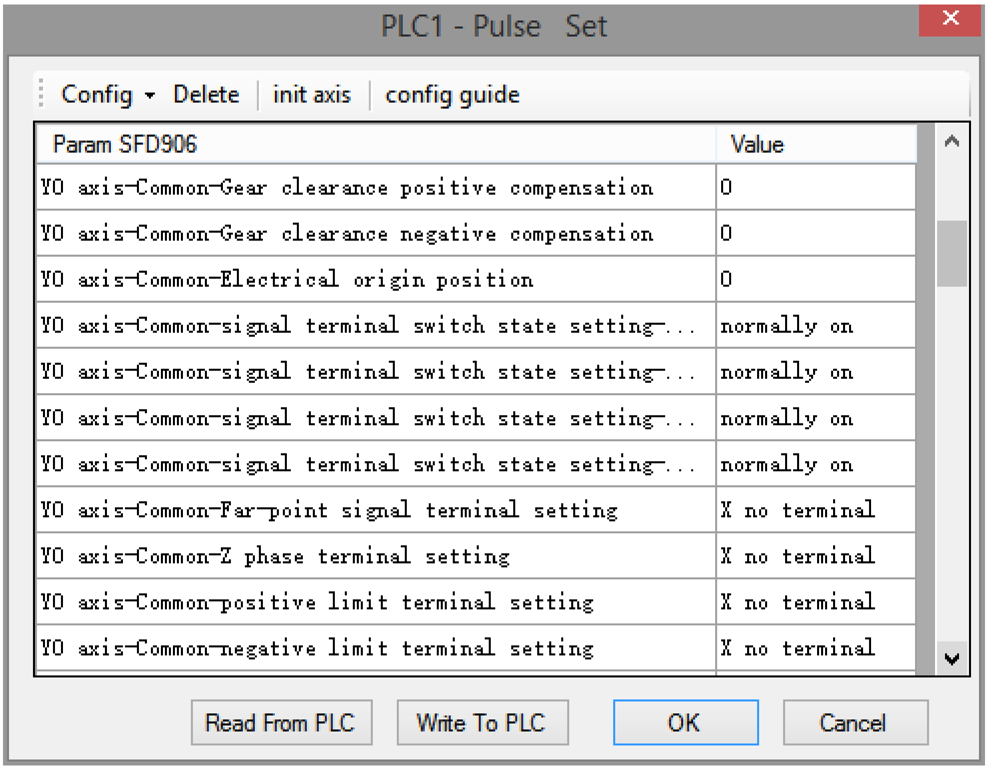
Software configurations:

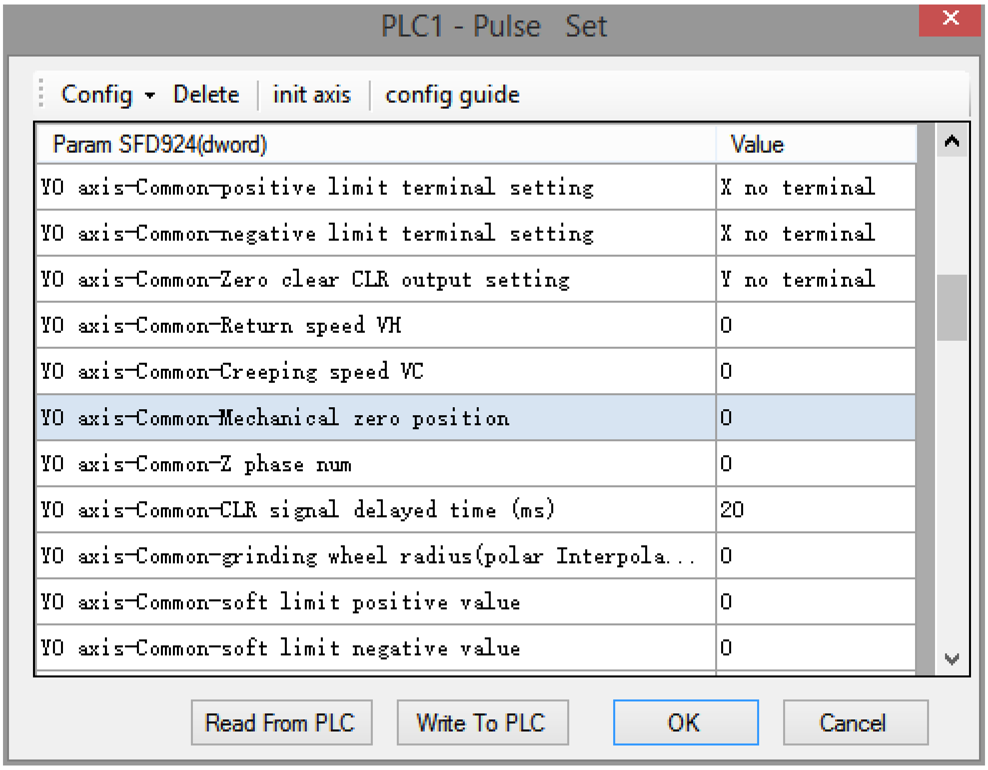
- Pulse configuration

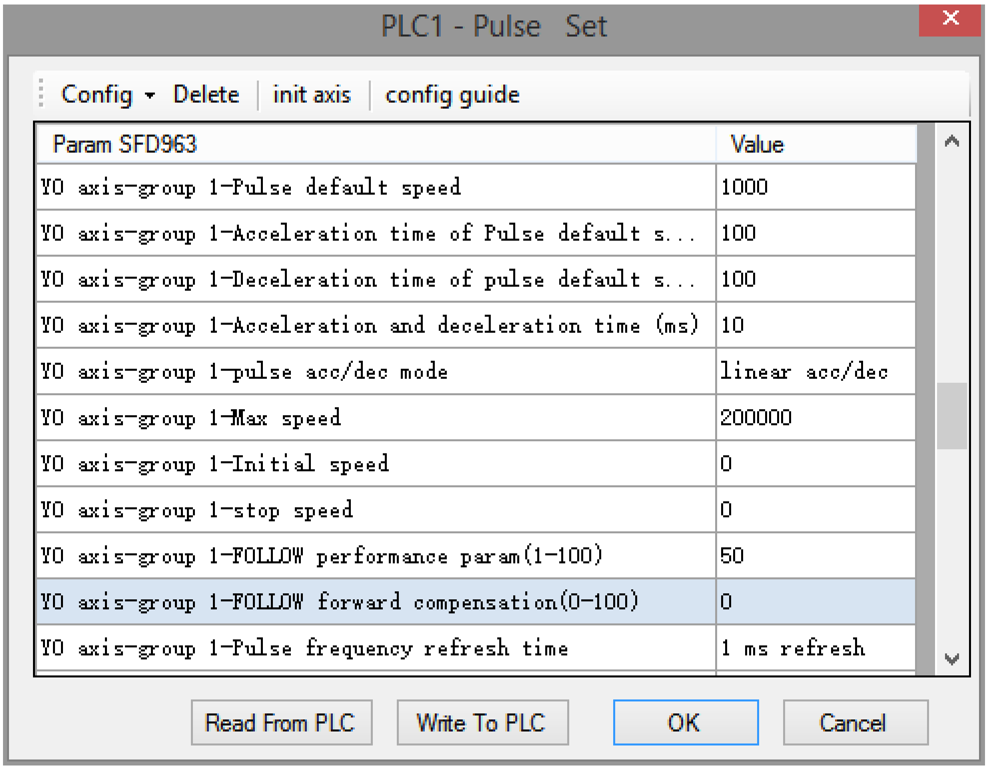


- Pulse system parameters

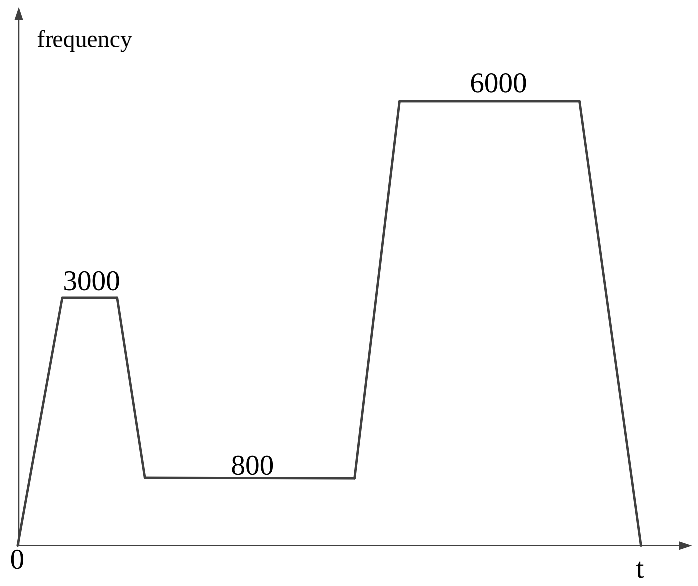



PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD906
Value
Y0 axis-Common-Gear clearance positive compensation
0
Y0 axis-Common-Gear clearance negative compensation
0
Y0 axis-Common-Electrical origin position
0
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-signal terminal switch state setting-...
normally on
Y0 axis-Common-Far-point signal terminal setting
X no terminal
Y0 axis-Common-Z phase terminal setting
X no terminal
Y0 axis-Common-positive limit terminal setting
X no terminal
Y0 axis-Common-negative limit terminal setting
X no terminal
Read From PLC
Write To PLC
OK
Cancel


PLC1 - Pulse Set
Config
Delete
init axis
config guide
Param SFD924(dword)
Value
Y0 axis-Common-positive limit terminal setting
X no terminal
Y0 axis-Common-negative limit terminal setting
X no terminal
Y0 axis-Common-Zero clear CLR output setting
Y no terminal
Y0 axis-Common-Return speed VH
0
Y0 axis-Common-Creeping speed VC
0
Y0 axis-Common-Mechanical zero position
0
Y0 axis-Common-Z phase num
0
Y0 axis-Common-CLR signal delayed time (ms)
20
Y0 axis-Common-grinding wheel radius(polar Interpola...
0
Y0 axis-Common-soft limit positive value
0
Y0 axis-Common-soft limit negative value
0
Read From PLC
Write To PLC
OK
Cancel



➢ Pulse sending oscillogram



3-2. Application of motion control in arc saw machining system

1. Introduction of arc saw technology

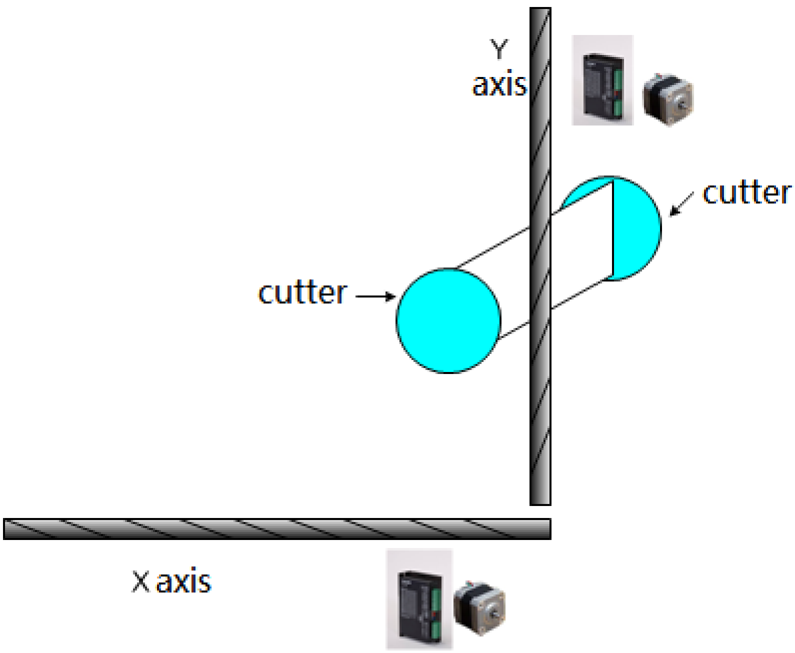
The arc saw is a machine used to cut arc boards. The mechanical characteristics are that the arc radius is large and the motor load is large.

2. Products applied in this system

Product name	Model	Number
PLC	XDM-32T4-E	1
HMI	OP320-A	1
Stepper driver	DP-21P5	2

3. Composition of control system

(1) The composition of system hardware



As shown in the figure, two stepper motors control X and Y axis respectively, and use the arc interpolation instruction of XINJE XDM PLC to make X and Y axis coordinate and get out of the circular arc track. The relative distance of the cutter installed on the workbench determines the width of the plate cut by the cutter.

(2) Technical difficulties

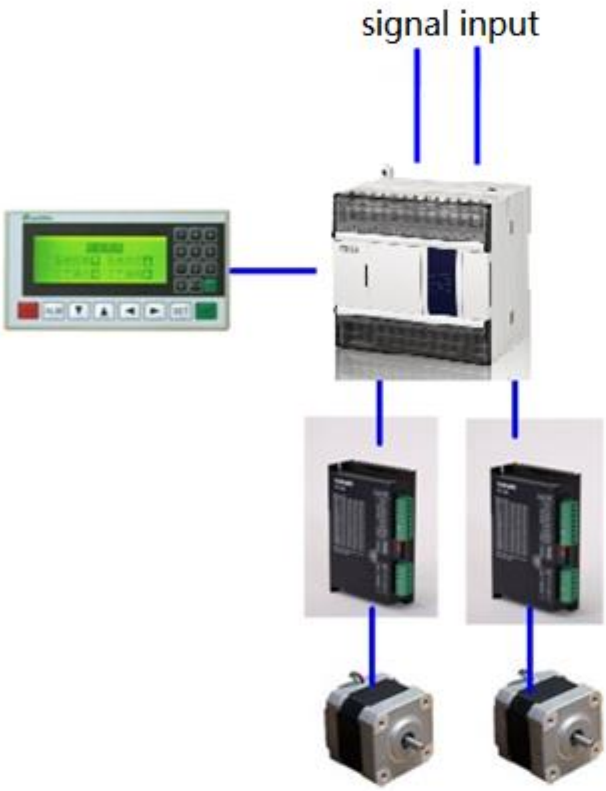
- The processing arc radius is large, the pitch of the XY axis screw is large, the number of pulse and the amount of movement are difficult to configure, if the setting is not appropriate, the data calculation is easy to overflow.
- Due to the heavy load of the motor, it is easy to lose step or overshoot.

- The speed of returning to the mechanical origin should not be too fast.
- Owing to the ellipse of the processed arc board, the ellipse can not be cut directly by arc interpolation, otherwise the board can not be sawn through.

(3) Control scheme

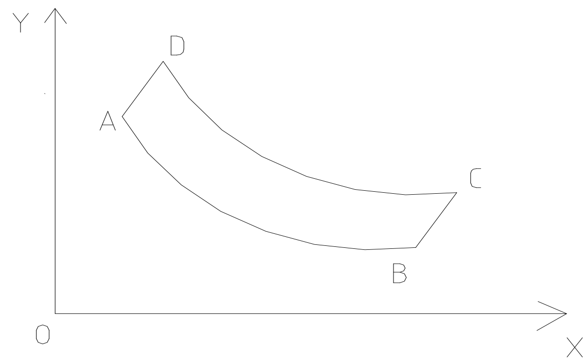
This scheme adopts the motion-controlled PLC XDM, which has high-speed command operation, built-in four 100KHz high-speed pulse output, support motion control command arc interpolation, RS232, RS485 serial ports, convenient for various upper computer monitoring, powerful external interrupt function, greatly saves the electrical cost for customers.

In view of the above difficulties, we adopt the method of reducing the ratio of the number of pulses and the amount of movement to reduce the calculation value and prevent the calculation overflow. (For example, the number of pulses is 2400 and the amount of movement is 10000. When setting parameters, the amount of movement is reduced by 10 times to 1000, so the number of pulses per unit is increased by 10 times. When setting physical quantities, we will reduce by 10 times accordingly. For example, when setting 1000 millimeters, we only need to set 100 in the corresponding registers.) In order to ensure that the motor is not out of step or overshoot, it is necessary to set the acceleration and deceleration time a little longer and increase the driver current (note that the motor is easy to heat if the current is too large). Before the arc interpolation, the straight line cutting is carried out, and then the arc cutting is carried out, which solves the problem that the direct arc cutting can not be cut through.



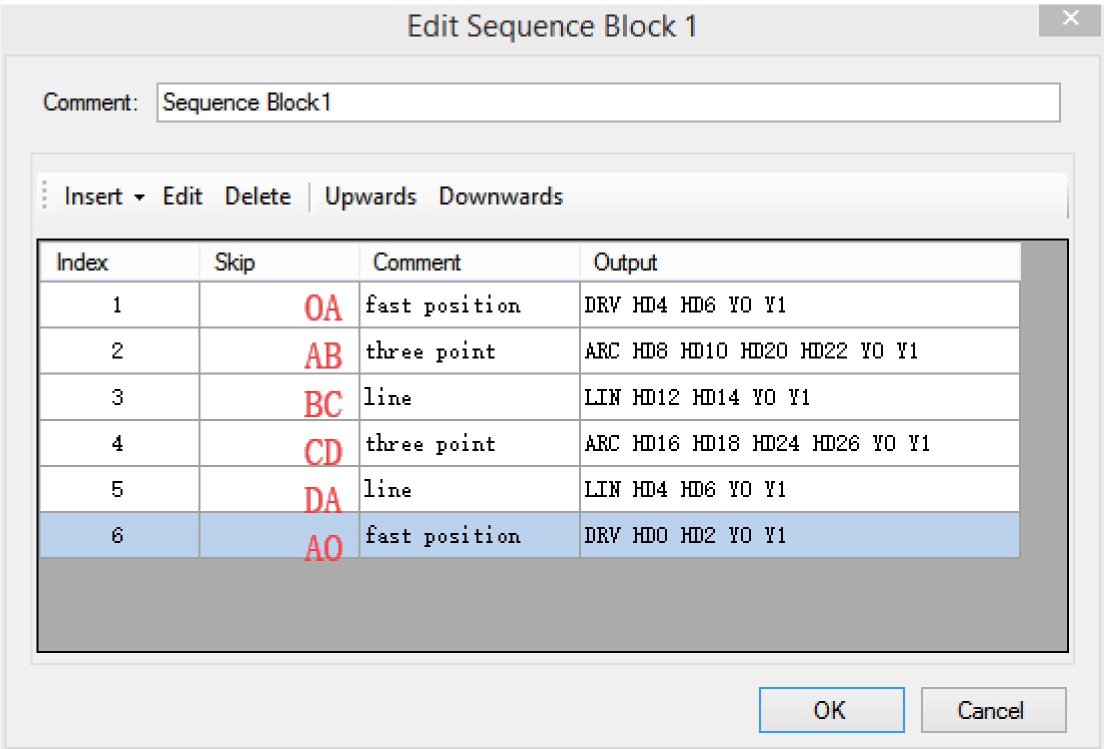
In positioning motion control, returning to mechanical origin is very important for control accuracy. However, some mechanical motors have a large load and only one origin signal. The control object is a stepper motor. There is no Z-phase signal output, and the requirement of returning to the origin is fast. In this case, we use the ZRN instruction in XD to configure the internal acceleration and deceleration time settings. The problem has solved.

(4) The operation diagram of the interpolation instructions in the system is as follows:



The coordinates of the points in the figure are as follows: O(HD0, HD2), A(HD4, HD6), B(HD8, HD10), C(HD12, HD14), C(HD16, HD18), the midpoint coordinates of the AB arc are (HD20, HD22), the midpoint coordinates of the CD arc are (HD24, HD26).
Motion path: O→A→B→C→D→A→O.

5. **The interpolation instructions in the system are as follows:**



Index	Skip	Comment	Output
1	OA	fast position	DRV HD4 HD6 Y0 Y1
2	AB	three point	ARC HD8 HD10 HD20 HD22 Y0 Y1
3	BC	line	LIN HD12 HD14 Y0 Y1
4	CD	three point	ARC HD16 HD18 HD24 HD26 Y0 Y1
5	DA	line	LIN HD4 HD6 Y0 Y1
6	AO	fast position	DRV HD0 HD2 Y0 Y1

3-3. Application of motion control in hair planting machine

1. Process introduction

At present, the electric control system structure of hair planting machine is mainly divided into single chip computer control system or CNC numerical control system. Among them, the single-chip computer control system is based on the integrated service of automation system manufacturer, supplemented by the independent research and development of toothbrush equipment manufacturer.

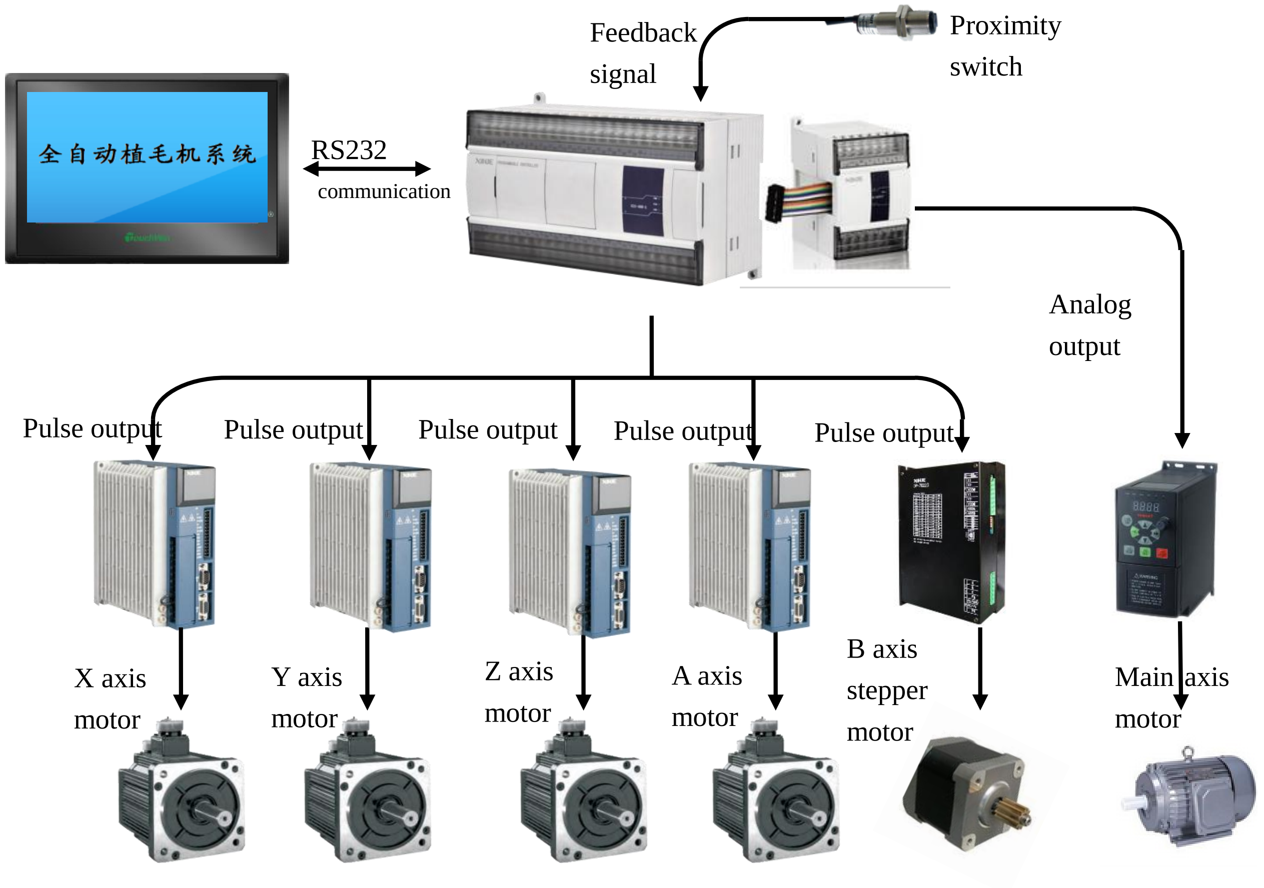
The drive structure of high-speed toothbrush hair planter is composed of main drive shaft and four servo drive shaft systems. The four servo axes are horizontal X-axis, vertical Y-axis, hair changing Z-axis and rotary A-axis. The position of the toothbrush hole is determined by the coordinates of the XY two axes. The A axis play the role of replacing the next toothbrush and the Z axis play the role of replacing the brush color. When the main shaft motor (frequency converter control) runs, the four electronically controlled servo shafts will run, while the other four shafts will stop when the main shaft stops. The speed of the main axis determines the speed of hair planting. The response of the four servo shafts need coordinated driving, otherwise, hair removal or hair irregularity will occur.

2. the products required in the application

Product name	Model	Quantity
PLC	XDM-60T4-E	1
Extension module	XD-E2DA	1
HMI	TG865-MT（U）	1
Servo drive	DS3-20P7-PQA	3
Servo drive	DS3-20P4-PQA	1

3. Composition of Control System

(1) The Composition of System Hardware



(2) Finished toothbrush products

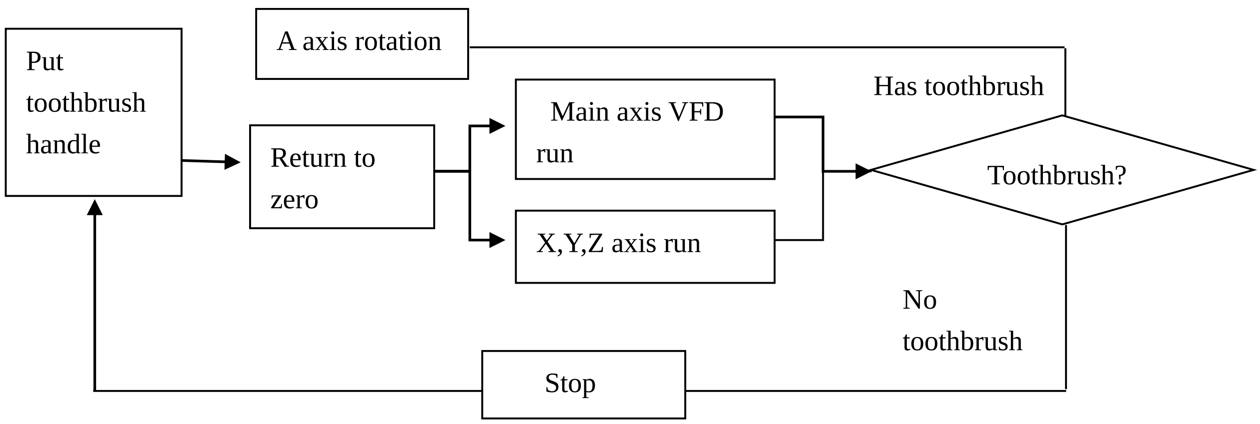
(3) Technological difficulties

The difficulty of developing servo solution is the joint debugging of electromechanical system, in which the adjustment of servo gain and the cooperation of PLC triangular function curve are the main problems. Among the four servo shafts, the mechanical inertia of X-axis and Y-axis is relatively stable due to the screw drive structure, and it is easy to debug, so it is possible to modify the speed gain. The Z-axis of the turning plate is a rotating axis. There is centrifugal force in high-speed rotation. If the gain of the turning plate is set very high, the motor will vibrate when it starts and stops. At this time, the position filtering time parameters can be modified to eliminate the vibration. Comparatively speaking, the structure of cam mechanism for changing hair U-axis makes debugging more difficult. In addition, the mechanical rigidity of U-axis is not good. When the motor runs, the inertia ratio varies greatly, the output current of the motor varies greatly, and the parameters can not be adjusted properly. When the motor runs around, the shaft either vibrates or screams, or reacts slowly. When the parameters are adjusted, the gain of the speed loop and the filtering time parameters and position loop gain need to be adjusted accordingly.

(4) Control solution

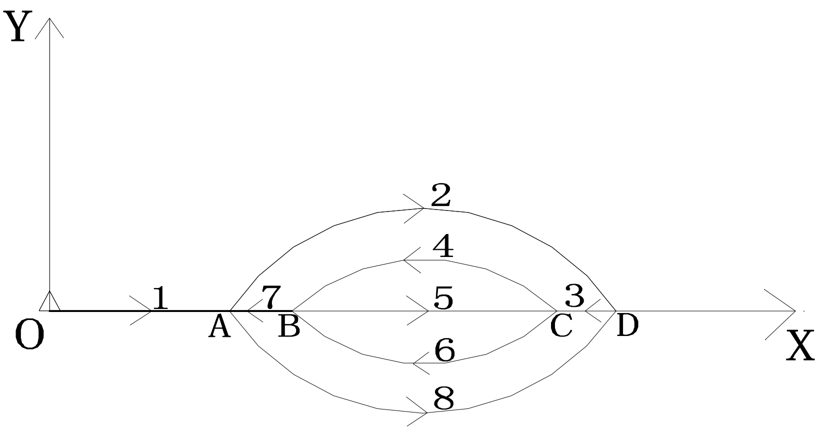
Mainly control axis pulse command signal to achieve servo drive, usually four-axis control output. The motion control type of PLC XDM-60T4-E is chosen. It has a response speed of 0.1ms and four high-speed pulses, which can realize the two-axis interpolation operation required by the toothbrush hair planter. The four sets of servo drivers are DS3 series AC servo system with power of 400W~750W. The driver has many functions, such as strong overload ability, strong anti-load disturbance ability, large starting moment, high dynamic response speed and short positioning time. The main axis motor frequency converter model is Xinje VB5N series, the power is 400 W.

(5) action order



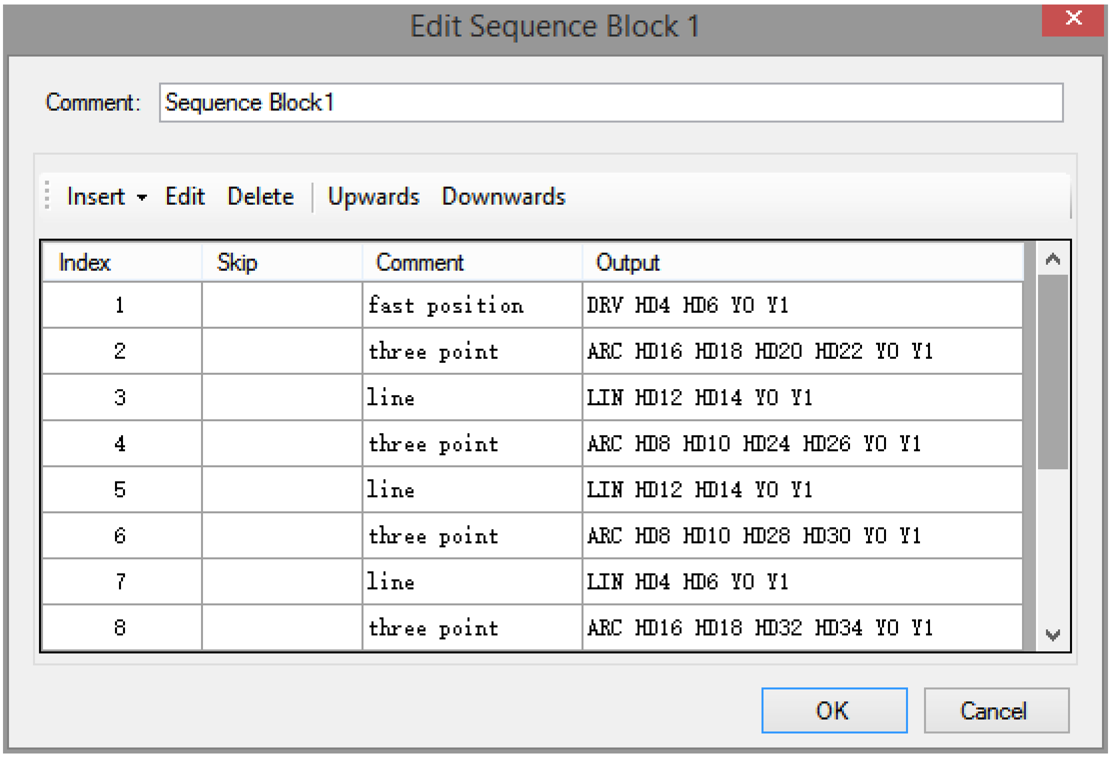
Action process: The clip holds the toothbrush handle from Y axis direction → 90 degrees positioning to Z axis direction → platform drives the clip to do X Y axis movement enables the brush hair to be hit into the hole of the toothbrush head → hair planting completes, the clip rotates downward 90 degrees → the clip loosens, and a toothbrush is produced. The application of Xinje XDM series PLC and DS5 servo system can achieve 900 times/minute hair planting speed. And at the same time of high-speed start and stop, the stability and softness of the overall movement is particularly prominent. Through the application of self-made pulse S curve in PLC, we can achieve hole skipping hair planting. When skipping, the machine is almost as smooth as usual without obvious jitter while ensuring the accuracy of skipping.

4. The operation diagram of the interpolation instructions in the system is as follows.

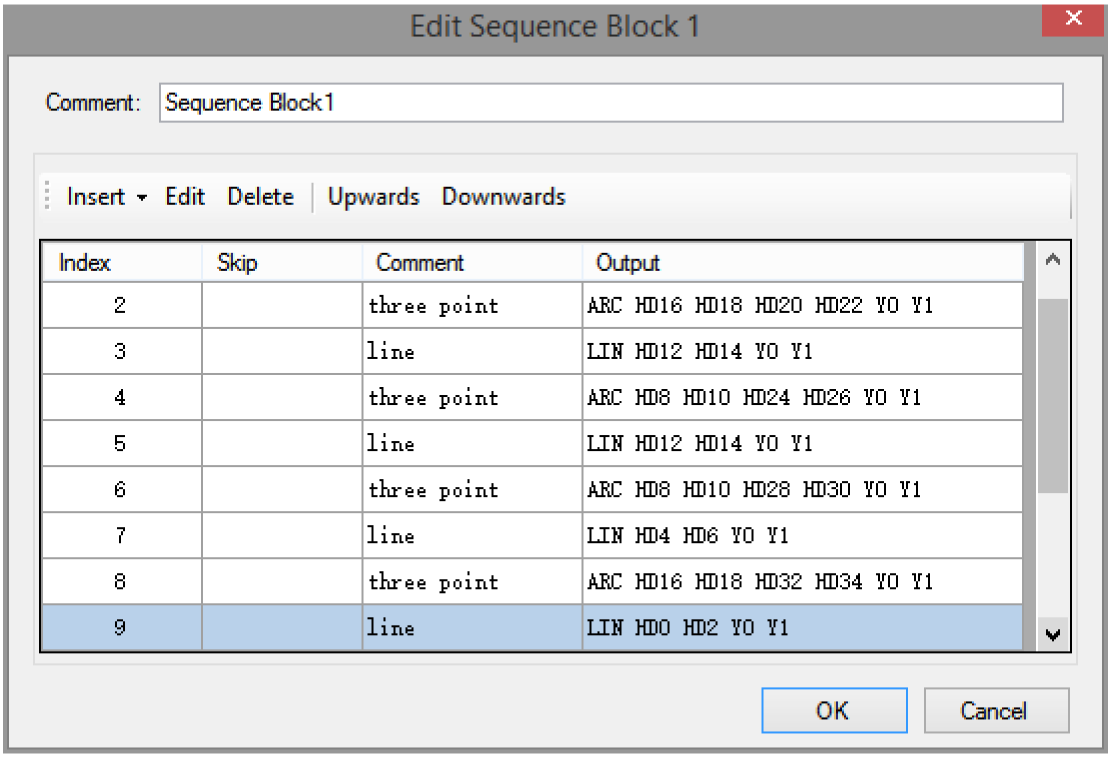


The coordinates of the points in the figure are as follows: O (HD0, HD2), A (HD4, HD6), B (HD8, HD10), C (HD12, HD14), D (HD16, HD18), the midpoint coordinates of the clockwise arc of AD segment (HD20, HD22), the midpoint coordinates of the anticlosewise arc of AD segment (HD32, HD34), the midpoint coordinates of the clockwise arc of BC segment (HD28, HD30), and the midpoint coordinates of the anticlockwise arc of BC segment (HD24, HD26). Path of particle: O→A→D→C→B→C→B→A→D→O.

5. The interpolation instructions in the system.



Index	Skip	Comment	Output
1		fast position	DRV HD4 HD6 Y0 Y1
2		three point	ARC HD16 HD18 HD20 HD22 Y0 Y1
3		line	LIN HD12 HD14 Y0 Y1
4		three point	ARC HD8 HD10 HD24 HD26 Y0 Y1
5		line	LIN HD12 HD14 Y0 Y1
6		three point	ARC HD8 HD10 HD28 HD30 Y0 Y1
7		line	LIN HD4 HD6 Y0 Y1
8		three point	ARC HD16 HD18 HD32 HD34 Y0 Y1

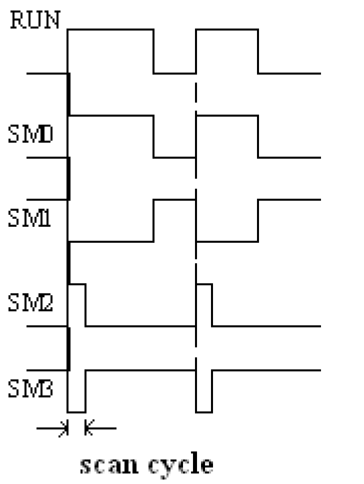

Edit Sequence Block 1
Comment:
Sequence Block1
Insert
Edit
Delete
Upwards
Downwards
Index
Skip
Comment
Output
2 three point ARC HD16 HD18 HD20 HD22 Y0 Y1
3 line LIN HD12 HD14 Y0 Y1
4 three point ARC HD8 HD10 HD24 HD26 Y0 Y1
5 line LIN HD12 HD14 Y0 Y1
6 three point ARC HD8 HD10 HD28 HD30 Y0 Y1
7 line LIN HD4 HD6 Y0 Y1
8 three point ARC HD16 HD18 HD32 HD34 Y0 Y1
9 line LIN HD0 HD2 Y0 Y1
OK
Cancel

Appendix Special soft element list

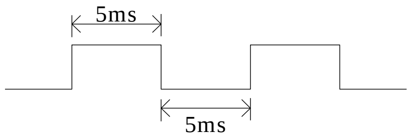
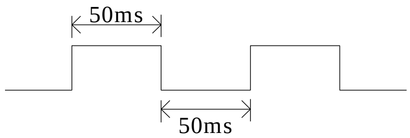
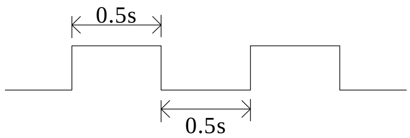
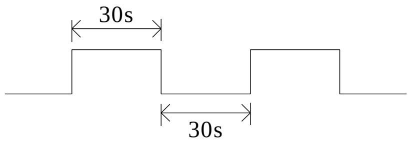
Appendix mainly introduces the functions of XD/XL series PLC special soft element, data register, FlashROM and the address distribution of expansions for users to search.

Appendix 1. Special auxiliary relay

Initial Status（SM0-SM7）

ID	Function	Description	
SM000	Coil ON when running		SM000 keeps ON when PLC running
SM001	Coil OFF when running		SM001 keeps OFF when PLC running
SM002	Initial positive pulse coil		SM002 is ON in first scan cycle
SM003	Initial negative pulse coil		SM003 is OFF in first scan cycle
SM004	PLC running error	When SM4 sets ON, it indicates that there is an error in the operation of PLC. (Firmware version V3.4.5 and above supports this function by PLC)	
SM005	Battery low alarm coil	When the battery voltage is less than 2.5V, SM5 will put ON (at this time, please replace the battery as soon as possible, otherwise the data will not be maintained)	
SM007	Power-off memory data error		

Clock（SM11-SM14）

ID	Function	Description
SM011	10ms frequency cycle	
SM012	100ms frequency cycle	
SM013	1s frequency cycle	
SM014	1min frequency cycle	

Mark（SM20-SM22）

ID	**Function**	**Description**
SM020	Zero bit	SM020 is ON when plus/minus operation result is 0
SM021	Borrow bit	SM021 is ON when minus operation overflows
SM022	Carry bit	SM022 is ON when plus operation overflows

PC Mode（SM32-SM34）

ID	**Function**	**Description**
SM032	Retentive register reset	When SM032 is ON, ON/OFF mapping memory of HM、HS and current values of HT、HC、HD will be reset.
SM033	Clear user’s program	When SM033 is ON, all PLC user’s program will be cleared.
SM034	All output forbidden	When SM034 is ON, all PLC external contacts will be set

		OFF.

Stepping Ladder

ID	Function	Description
SM040	The process is running	Set ON when the process is running

Interruption ban（SM50-SM90）

ID	Address	Function	Description
SM050	I0000/I0001	Forbid input interruption 0	After executing EI instruction, the input interruption couldn’t act independently when M acts, even if the interruption is allowed. E.g.： when SM050 is ON, I0000/I0001 is forbidden.
SM051	I0100/I0101	Forbid input interruption 1	
SM052	I0200/I0201	Forbid input interruption 2	
SM053	I0300/I0301	Forbid input interruption 3	
SM054	I0400/I0401	Forbid input interruption 4	
……	……	……	
SM069	I1900/I1901	Forbid input interruption 19	
SM070	I40**	Forbid timing interruption 0	After executing EI instruction, the timing interruption couldn’t act independently when M acts, even if the interruption is allowed.
SM071	I41**	Forbid timing interruption 1	
SM072	I42**	Forbid timing interruption 2	
SM073	I43**	Forbid timing interruption 3	
SM074	I44**	Forbid timing interruption 4	
……	……	……	
SM089	I59**	Forbid timing interruption 19	
SM090		Forbid all interruptions	Forbid all interruptions

High Speed Ring Counter（SM99）

address	Function	Note
SM099	High Speed Ring Counting enable	SM99 set ON，SD99 add one per 0.1ms, cycle between 0 and 32767

High speed count complete（SM100-SM109）

Address	Function	Note
SM100	HSC0 count complete flag（100 segments）	
SM101	HSC2 count complete flag（100 segments）	
SM102	HSC4 count complete flag（100 segments）	
SM103	HSC6 count complete flag（100 segments）	
SM104	HSC8 count complete flag（100 segments）	
SM105	HSC10 count complete flag（100 segments）	
SM106	HSC12 count complete flag（100 segments）	
SM107	HSC14 count complete flag（100 segments）	
SM108	HSC16 count complete flag（100 segments）	
SM109	HSC18 count complete flag（100 segments）	

High speed counter direction（SM110-SM119）

Address	Function	Note
SM110	HSC0 direction flag	
SM111	HSC2 direction flag	
SM112	HSC4 direction flag	
SM113	HSC6 direction flag	
SM114	HSC8 direction flag	
SM115	HSC10 direction flag	
SM116	HSC12 direction flag	
SM117	HSC14 direction flag	
SM118	HSC16 direction flag	
SM119	HSC18 direction flag	

High speed counter error（SM120-SM129）

address	Function	Note
SM120	HSC0 error flag	
SM121	HSC2 error flag	
SM122	HSC4 error flag	
SM123	HSC6 error flag	
SM124	HSC8 error flag	
SM125	HSC10 error flag	
SM126	HSC12 error flag	
SM127	HSC14 error flag	
SM128	HSC16 error flag	
SM129	HSC18 error flag	

Communication（SM140-SM193）

	Address	Function	Note
Serial port 0	SM140	Modbus instruction execution flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM141	X-NET instruction execution flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM142	Free format communication sending flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM143	Free format communication receive complete flag	When receiving a frame of data or receiving data timeout, set ON. Require user program to set OFF
Serial port 1	SM150	Modbus instruction execution flag	Same to SM140
	SM151	X-NET instruction execution flag	Same to SM141
	SM152	Free format communication sending flag	Same to SM142
	SM153	Free format communication receive complete flag	Same to SM143
Serial port 2	SM160	Modbus instruction execution flag	Same to SM140
	SM161	X-NET instruction execution flag	Same to SM141
	SM162	Free format communication sending flag	Same to SM142
	SM163	Free format communication receive complete flag	Same to SM143
Serial port 3	SM170	Modbus instruction execution flag	Same to SM140
	SM171	X-NET instruction execution flag	Same to SM141
	SM172	Free format communication sending flag	Same to SM142
	SM173	Free format communication receive complete flag	Same to SM143
Serial port 4	SM180	Modbus instruction execution flag	Same to SM140
	SM181	X-NET instruction execution flag	Same to SM141
	SM182	Free format communication sending flag	Same to SM142
	SM183	Free format communication receive complete flag	Same to SM143
Serial	SM190	Modbus instruction execution flag	Same to SM140

port 5	SM191	X-NET instruction execution flag	Same to SM141
	SM192	Free format communication sending flag	Same to SM142
	SM193	Free format communication receive complete flag	Same to SM143

Sequence Function BLOCK（SM240-SM349）

ID	Function	Description
SM300	BLOCK1 running flag	SM300 will be ON when block1 is running
SM301	BLOCK2 running flag	SM301 will be ON when block2 is running
SM302	BLOCK3 running flag	SM302 will be ON when block3 is running
SM303	BLOCK4 running flag	SM303 will be ON when block4 is running
SM304	BLOCK5 running flag	SM304 will be ON when block5 is running
SM305	BLOCK6 running flag	SM305 will be ON when block6 is running
……	……	
SM346	BLOCK47 running flag	SM346 will be ON when block47is running
SM347	BLOCK48 running flag	SM347 will be ON when block48 is running
SM348	BLOCK49 running flag	SM348 will be ON when block49 is running
SM349	BLOCK50 running flag	SM349 will be ON when block50 is running

Error check（SM400-SM413）

ID	Function	Description
SM400	I/O error	ERR LED keeps ON, PLC don not run and output, check when power on
SM401	Expansion module communication error	
SM402	BD communication error	
……		
SM405	No user program	Internal code check wrong
SM406	User program error	Implement code or configuration table check wrong
SM407	SSFD check error	ERR LED keeps ON, PLC don not run and output, check when power on
SM408	Memory error	Can not erase or write Flash
SM409	Calculation error	
SM410	Offset overflow	Offset exceeds soft element range
SM411	FOR-NEXT	Reset when power on or users can also reset by hand.

	overflow	
SM412	Invalid data fill	When offset of register overflows, the return value will be SM372 value

Error Message（SM450-SM452）

ID	Function	Description
SM450	System error check	
SM451	Hardfault interrupt flag	
SM452		
SM453	SD card error	
SM454	Power supply is cut off	
……		
SM460	Extension module ID not match	
SM461	BD/ED module ID not match	
SM462	Extension module communication overtime	
SM463	BD/ED module communication overtime	

Expansion Modules, BD Status（SM500）

ID	Function	Description
SM500	Module status read is finished	

High speed pulse（SM1000-SM1190）

ID	Function	Explanation	Output point
SM1000	Pulse sending flag	ON: Pulse is sending	Y0
SM1001	Direction flag	1 is positive direction, related direction signal is ON	
SM1002	Accumulated pulse number overflow flag	1 is overflow	
SM1003	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1004			
SM1005			
SM1006			
SM1007			
SM1008			
SM1009			

SM1010	Pulse error flag	ON: error	
SM1020	Pulse sending flag	ON: Pulse is sending	Y1
SM1021	Direction flag	1 is positive direction, related direction signal is ON	
SM1022	Accumulated pulse number overflow flag	1 is overflow	
SM1023	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1024			
SM1025			
SM1026			
SM1027			
SM1028			
SM1029			
SM1030	Pulse error flag	ON: error	
SM1040	Pulse sending flag	ON: Pulse is sending	Y2
SM1041	Direction flag	1 is positive direction, related direction signal is ON	
SM1042	Accumulated pulse number overflow flag	1 is overflow	
SM1043	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1044			
SM1045			
SM1046			
SM1047			
SM1048			
SM1049			
SM1050	Pulse error flag	ON: error	
SM1060	Pulse sending flag	ON: Pulse is sending	Y3
SM1061	Direction flag	1 is positive direction, related direction signal is ON	
SM1062	Accumulated pulse number overflow flag	1 is overflow	
SM1063	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1064			
SM1065			
SM1066			
SM1067			
SM1068			
SM1069			
SM1070	Pulse error flag	ON: error	

SM1080	Pulse sending flag	ON: Pulse is sending	Y4
SM1081	Direction flag	1 is positive direction, related direction signal is ON	
SM1082	Accumulated pulse number overflow flag	1 is overflow	
SM1083	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1084			
SM1085			
SM1086			
SM1087			
SM1088			
SM1089			
SM1090	Pulse error flag	ON: error	
SM1100	Pulse sending flag	ON: Pulse is sending	Y5
SM1101	Direction flag	1 is positive direction, related direction signal is ON	
SM1102	Accumulated pulse number overflow flag	1 is overflow	
SM1103	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1104			
SM1105			
SM1106			
SM1107			
SM1108			
SM1109			
M1110	Pulse error flag	ON: error	
SM1120	Pulse sending flag	ON: Pulse is sending	Y6
SM1121	Direction flag	1 is positive direction, related direction signal is ON	
SM1122	Accumulated pulse number overflow flag	1 is overflow	
SM1123	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1124			
SM1125			
SM1126			
SM1127			
SM1128			
SM1129			
SM1130	Pulse error flag	ON: error	
SM1140	Pulse sending flag	ON: Pulse is sending	Y7

SM1141	Direction flag	1 is positive direction, related direction signal is ON	
SM1142	Accumulated pulse number overflow flag	1 is overflow	
SM1143	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1144			
SM1145			
SM1146			
SM1147			
SM1148			
SM1149			
SM1150	Pulse error flag	ON: error	
SM1160	Pulse sending flag	ON: Pulse is sending	Y10
SM1161	Direction flag	1 is positive direction, related direction signal is ON	
SM1162	Accumulated pulse number overflow flag	1 is overflow	
SM1163	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1164			
SM1165			
SM1166			
SM1167			
SM1168			
SM1169			
SM1170	Pulse error flag	ON: error	
SM1180	Pulse sending flag	ON: Pulse is sending	Y11
SM1181	Direction flag	1 is positive direction, related direction signal is ON	
SM1182	Accumulated pulse number overflow flag	1 is overflow	
SM1183	Accumulated pulse equivalent overflow flag	1 is overflow	
SM1184			
SM1185			
SM1186			
SM1187			
SM1188			
SM1189			
SM1190	Pulse error flag	ON: error	

Appendix 2. Special data reigster list

Battery （SD5~SD7）

ID	Function	Description
SD005	Battery register	It will display 100 when the battery voltage is 3V, if the battery voltaeg is lower than 2.5V, it will display 0, it means please change new battery at once, otherwise the data will lose when PLC power off.
SD007	Power-off memory data error type	

Clock（SD10-SD019）

ID	Function	Description
SD010	Current scan cycle	100us，us is the unit
SD011	Min scan time	100us，us is the unit
SD012	Max scan time	100us，us is the unit
SD013	Second (clock)	0~59 (BCD code)
SD014	Minute (clock)	0~59 (BCD code)
SD015	Hour (clock)	0~23 (BCD code)
SD016	Day (clock)	0~31 (BCD code)
SD017	Month (clock)	0~12 (BCD code)
SD018	Year (clock)	2000~2099 (BCD code)
SD019	Week (clock)	0(Sunday)~6(Saturday)(BCD code)

Flag （SD020-SD031）

ID	Function	Note
SD020	Model type	
SD021	model（low-8）series（high-8）	
SD022	Compatiable system version（low）system version（high）	
SD023	Compatiable model version（low）model version（high）	
SD024	Model info	
SD025	Model info	
SD026	Model info	
SD027	Model info	
SD028	Suitable software version	
SD029	Suitable software version	
SD030	Suitable software version	
SD031	Suitable software version	

Step ladder（SD040）

ID	Function	Description
SD40	Flag of the executing process S	

High Speed Counting（SD100-SD109）

ID	Function	Description	
SD100	Current segment (No. n segment)		HSC00
SD101	Current segment (No. n segment)		HSC02
SD102	Current segment (No. n segment)		HSC04
SD103	Current segment (No. n segment)		HSC06
SD104	Current segment (No. n segment)		HSC08
SD105	Current segment (No. n segment)		HSC10
SD106	Current segment (No. n segment)		HSC12
SD107	Current segment (No. n segment)		HSC14
SD108	Current segment (No. n segment)		HSC16
SD109	Current segment (No. n segment)		HSC18

High speed counter error（SD120-SD129）

ID	Function	Note
SD120	HSC0 error info	
SD121	HSC2 error info	
SD122	HSC4 error info	
SD123	HSC6 error info	
SD124	HSC8 error info	
SD125	HSC10 error info	
SD126	HSC12 error info	
SD127	HSC14 error info	
SD128	HSC16 error info	
SD129	HSC18 error info	

communication（SD140~SD199）

	ID	Function	Note
	SD140	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime

Serial port 0			180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD141	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD142	Free format communication send result	0: correct 410: free format send buffer overflow
	SD143	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD144	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD149		
Serial port 1	SD150	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD151	X-Net communication result	0: correct 1: communication overtime 2: memory error

			3: receive CRC error
	SD152	Free format communication send result	0: correct 410: free format send buffer overflow
	SD153	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD154	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD159		
Serial port 2	SD160	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD161	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD162	Free format communication send result	0: correct 410: free format send buffer overflow
	SD163	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD164	Free format communication receive data numbers	In bytes, there are no start and stop characters

	……		
	SD169		
Serial port 3	SD170~SD179		
Serial port 4	SD180~SD189		
Serial port 5	SD190~SD199		

Sequence Function Block（SD300-SD399）

ID	Function	Description
SD300	Executing instruction of BLOCK1	The value will be used when BLOCK monitors
SD301	Executing instruction of BLOCK2	The value will be used when BLOCK monitors
SD302	Executing instruction of BLOCK3	The value will be used when BLOCK monitors
SD303	Executing instruction of BLOCK4	The value will be used when BLOCK monitors
SD304	Executing instruction of BLOCK5	The value will be used when BLOCK monitors
SD305	Executing instruction of BLOCK6	The value will be used when BLOCK monitors
……	……	……
SD396	Executing instruction of BLOCK97	The value will be used when BLOCK monitors
SD397	Executing instruction of BLOCK98	The value will be used when BLOCK monitors
SD398	Executing instruction of BLOCK99	The value will be used when BLOCK monitors
SD399	Executing instruction of BLOCK100	The value will be used when BLOCK monitors

Error Check（SD400-SD413）

ID	Function	Note
SD400		
SD401	Extension module no. of communication error	Means module no.n is error
SD402	BD/ED module no. of communication error	
SD403	FROM/TO error type	
SD404	PID error type	
……		
SD409	Calculation error code	1: divide by 0 error 2: MRST, MSET front operand address less than back operand

		3: ENCO, DECO data bits of encoding and decoding instructions exceed the limit. 4: BDC code error 7: Radical sign error
SD410	The number of offset register D when offset crosses the boundary	
SD411		
SD412	Invalid data fill value (low 16 bits)	
SD413	Invalid data fill value (high 16 bits)	

Error Check（SD450-SD452）

ID	Function	Description
SD450	1：Watchdog act (Default 200ms) 2：Control block application fail 3：Visit illegal address	
SD451	Hardware error type: 1：Register error 2：Bus error 3：Usage error	
SD452	Hardware error	
SD453	SD card error	
SD454	Power-off time	
SD460	Extension module ID not match	
SD461	BD/ED module ID not match	
SD462	Extension module communication overtime	
SD463	BD/ED module communication overtime	

Expansion Modules, BD Status（SD500-SD516）

ID	Function	Description	
SD500	Module number Expansion modules：#10000～10015 BD：#20000～20001 ED: #30000		
SD501~516	Expansion module, BD /ED status		16 registers

Module info（SD520-SD823）

ID	Function	Explanation	Note
SD520~SD535	Extension module info	Extension module 1	Each extension module, BD, ED occupies 16 registers
……	……	……	
SD760~SD775	Extension module info	Extension module 16	
SD776~SD791	BD module info	BD module 1	
SD792~SD807	BD module info	BD module 2	
SD808~SD823	ED module info	ED module 1	

Expansion Module Error Information

ID	Function	Description	
SD860	Error times of module read		Expansion module 1
SD861	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD862	Error times of module write		
SD863	Error types of module write		
SD864	Error times of module read		Expansion module 2
SD865	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD866	Error times of module write		
SD867	Error types of module write		
……			
SD920	Error times of module read		Expansion module 16
SD921	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD922	Error times of module write		

SD923	Error types of module write		
SD924	Error times of module read		BD module 1
SD925	Error types of module read		
SD926	Error times of module write		
SD927	Error types of module write		
SD928	Error times of module read		BD module 2
SD929	Error types of module read		
SD930	Error times of module write		
SD931	Error types of module write		
SD932	Error times of module read		ED module 1
SD933	Error types of module read		
SD934	Error times of module write		
SD935	Error types of module write		

Version info（SD990~SD993）

ID	Function	Explanation	Note
SD990	Firmware version date	Low 16-bit	
SD991	Firmware version compilation date	High 16-bit	
SD992	FPGA version compilation date	Low 16-bit	
SD993	FPGA version compilation date	High 16-bit	

High speed pulse（SD1000-SD1099）

ID	Function	Explanation	Output point
SD1000	Present segment (segment n)		Y0
SD1001			
SD1002	Present pulse number low 16-bit	(the unit is pulse number)	
SD1003	Present pulse number high 16-bit	(the unit is pulse number)	
SD1004	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1005	Present pulse number high	(the unit is pulse equivalent)	

	16-bit		
SD1006	Present output frequency low 16-bit	(the unit is pulse number)	
SD1007	Present output frequency high 16-bit	(the unit is pulse number)	
SD1008	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1009	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1010	Pulse error information	1: pulse data segment configuration error 2: In equivalent mode, the number of pulses per turn and the movement per 1 turn is 0. 3: System parameter block number error 4: Pulse parameter block number exceeding maximum limit 5: Stop after encountering positive limit signal 6: Stop after meeting the negative limit signal 10: No origin signal is set for origin regression 11:Velocity of origin regression VH is 0 12: Origin regression crawling speed VC is 0 or VC ≥ VH) 13: Origin regression signal error 15:Follow Performance Parameters ≤ 0 or >100 16:Follow Feedforward Compensation < 0 or >100 17:Follow Multiplication Coefficient and Division Coefficient Ratio ≤ 0 or >100 20: Interpolation Direction Terminal Not Set or Set Error 21: The default maximum interpolation speed is 0 22: Arc interpolation data error 23: Arc radius data error 24:Three-point Arc Data Error 25: In polar coordinate mode, the current position is (0, 0) 26: Control block allocation failed	
SD1011	error pulse data block number		
SD1020	Present segment (segment n)		Y1
SD1021			
SD1022	Present pulse number low 16-bit	(the unit is pulse number)	

SD1023	Present pulse number high 16-bit	(the unit is pulse number)	
SD1024	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1025	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1026	Present output frequency low 16-bit	(the unit is pulse number)	
SD1027	Present output frequency high 16-bit	(the unit is pulse number)	
SD1028	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1029	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1030	Pulse error information	Same to SD1010	
SD1031	error pulse data block number		
SD1040	Present segment (segment n)		Y2
SD1041			
SD1042	Present pulse number low 16-bit	(the unit is pulse number)	
SD1043	Present pulse number high 16-bit	(the unit is pulse number)	
SD1044	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1045	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1046	Present output frequency low 16-bit	(the unit is pulse number)	
SD1047	Present output frequency high 16-bit	(the unit is pulse number)	
SD1048	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1049	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1050	Pulse error information	Same to SD1010	
SD1051	error pulse data block number		
SD1060	Present segment		Y3

	(segment n)		
SD1061			
SD1062	Present pulse number low 16-bit	(the unit is pulse number)	
SD1063	Present pulse number high 16-bit	(the unit is pulse number)	
SD1064	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1065	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1066	Present output frequency low 16-bit	(the unit is pulse number)	
SD1067	Present output frequency high 16-bit	(the unit is pulse number)	
SD1068	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1069	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1070	Pulse error information	Same to SD1010	
SD1071	error pulse data block number		
SD1080	Present segment (segment n)		Y4
SD1082	Present pulse number low 16-bit	(the unit is pulse number)	
SD1083	Present pulse number high 16-bit	(the unit is pulse number)	
SD1084	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1085	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1086	Present output frequency low 16-bit	(the unit is pulse number)	
SD1087	Present output frequency high 16-bit	(the unit is pulse number)	
SD1088	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1089	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1090	Pulse error information	Same to SD1010	

SD1091	error pulse data block number		
SD1100	Present segment (segment n)		Y5
SD1102	Present pulse number low 16-bit	(the unit is pulse number)	
SD1103	Present pulse number high 16-bit	(the unit is pulse number)	
SD1104	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1105	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1106	Present output frequency low 16-bit	(the unit is pulse number)	
SD1107	Present output frequency high 16-bit	(the unit is pulse number)	
SD1108	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1109	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1110	Pulse error information	Same to SD1010	
SD1111	error pulse data block number		
SD1120	Present segment (segment n)		Y6
SD1122	Present pulse number low 16-bit	(the unit is pulse number)	
SD1123	Present pulse number high 16-bit	(the unit is pulse number)	
SD1124	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1125	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1126	Present output frequency low 16-bit	(the unit is pulse number)	
SD1127	Present output frequency high 16-bit	(the unit is pulse number)	
SD1128	Present output frequency	(the unit is pulse equivalent)	

	low 16-bit		
SD1129	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1130	Pulse error information	Same to SD1010	
SD1131	error pulse data block number		
SD1140	Present segment (segment n)		Y7
SD1142	Present pulse number low 16-bit	(the unit is pulse number)	
SD1143	Present pulse number high 16-bit	(the unit is pulse number)	
SD1144	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1145	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1146	Present output frequency low 16-bit	(the unit is pulse number)	
SD1147	Present output frequency high 16-bit	(the unit is pulse number)	
SD1148	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1149	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1150	Pulse error information	Same to SD1010	
SD1151	error pulse data block number		
SD1160	Present segment (segment n)		Y10
SD1162	Present pulse number low 16-bit	(the unit is pulse number)	
SD1163	Present pulse number high 16-bit	(the unit is pulse number)	
SD1164	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1165	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1166	Present output frequency	(the unit is pulse number)	

	low 16-bit		
SD1167	Present output frequency high 16-bit	(the unit is pulse number)	
SD1168	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1169	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1170	Pulse error information	Same to SD1010	
SD1171	error pulse data block number		
SD1180	Present segment (segment n)		Y11
SD1182	Present pulse number low 16-bit	(the unit is pulse number)	
SD1183	Present pulse number high 16-bit	(the unit is pulse number)	
SD1184	Present pulse number low 16-bit	(the unit is pulse equivalent)	
SD1185	Present pulse number high 16-bit	(the unit is pulse equivalent)	
SD1186	Present output frequency low 16-bit	(the unit is pulse number)	
SD1187	Present output frequency high 16-bit	(the unit is pulse number)	
SD1188	Present output frequency low 16-bit	(the unit is pulse equivalent)	
SD1189	Present output frequency high 16-bit	(the unit is pulse equivalent)	
SD1190	Pulse error information	Same to SD1010	
SD1191	error pulse data block number		

Special data register HSD (power-off retentive)

High speed pulse

ID	Function	Explanation	Output point
HSD0	Accumulated pulse number low 16-bit (the unit is pulse number)		Y0
HSD1	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD2	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD3	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD4	Accumulated pulse number low 16-bit (the unit is pulse number)		Y1
HSD5	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD6	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD7	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD8	Accumulated pulse number low 16-bit (the unit is pulse number)		Y2
HSD9	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD10	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD11	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD12	Accumulated pulse number low 16-bit (the unit is pulse number)		Y3
HSD13	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD14	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD15	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD16	Accumulated pulse number low 16-bit (the unit is pulse number)		Y4
HSD17	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD18	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		

HSD19	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD20	Accumulated pulse number low 16-bit (the unit is pulse number)		Y5
HSD21	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD22	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD23	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD24	Accumulated pulse number low 16-bit (the unit is pulse number)		Y6
HSD25	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD26	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD27	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD28	Accumulated pulse number low 16-bit (the unit is pulse number)		Y7
HSD29	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD30	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD31	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD32	Accumulated pulse number low 16-bit (the unit is pulse number)		Y10
HSD33	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD34	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD35	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		
HSD36	Accumulated pulse number low 16-bit (the unit is pulse number)		Y11
HSD37	Accumulated pulse number high 16-bit (the unit is pulse number)		
HSD38	Accumulated pulse number low 16-bit (the unit is pulse equivalent)		
HSD39	Accumulated pulse number high 16-bit (the unit is pulse equivalent)		

Appendix 3. Special FLASH register list

Special FLASH data register SFD

*** means it works only after repower on the PLC**

I filtering

ID	Function	Description
SFD0*	Input filter time	
SFD2*	Watchdog run-up time, default value is 200ms	

I Mapping

ID	Function	Description	
SFD10*	I00 corresponds to X**	Input terminal 0 corresponds to X** number	0xFF means terminal bad，0xFE means terminal idle
SFD11*	I01 corresponds to X**		
SFD12*	I02 corresponds to X**		
……	……		
SFD73*	I77 corresponds to X**	Default value is 77（Octonary）	

O Mapping

ID	Function	Description	
SFD74*	O00 corresponds to Y**	Output terminal 0 correspond to Y** number	0xFF means terminal bad，0xFE means terminal idle
		Default value is 0	
……	……		
SFD134*	O77 corresponds to Y**	Default value is 77（Octonary）	

I Attribute

ID	Function	Description	
SFD138*	I00 attribute	Attribute of input terminal 0	0：positive logic others：negative logic
SFD139*	I01 attribute		
……	……		
SFD201*	I77 attribute		

High Speed Counting

ID	Function	Description
SFD320	HSC0 frequency times	2：2 times frequency; 4: 4 times frequency(effective at AB phase counting mode)
SFD321	HSC2 frequency times	Ditto
SFD322	HSC4 frequency times	Ditto
SFD323	HSC6 frequency times	Ditto
SFD324	HSC8 frequency times	Ditto
SFD325	HSC10 frequency times	Ditto
SFD326	HSC12 frequency times	Ditto
SFD327	HSC14 frequency times	Ditto
SFD328	HSC16 frequency times	Ditto
SFD329	HSC18 frequency times	Ditto
SFD330	Bit selection of HSC absolute and relative（24 segment）	bit0 corresponds to HSC0， bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：relative 1：absolute
SFD331	Interrupt circulating of 24 segments high speed counting	bit0 corresponds to HSC0， bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：single 1：loop
SFD332	CAM function	bit0 corresponds to HSC0， bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：do not support CAM function 1：support CAM function

Expansion Module Configuration

ID	Function	Explanation
SFD340	Extension module configuration status（#1#2）	Configuration Status of Extension Modules 1 and 2
SFD341	Extension module configuration status（#3#4）	Configuration Status of Extension Modules 3 and 4
……	……	……
SFD347	Extension module configuration status（#15#16）	Configuration Status of Extension Modules 15 and 16
SFD348	BD module configuration status（#1#2）	Configuration Status of BD Modules 1 and 2
SFD349	ED module configuration status（#1）	Configuration Status of ED Module 1
SFD350 ： SFD359	Extension module configuration	Configuration of Extension Module 1
SFD360 ：	Extension module configuration	Configuration of Extension Module 2

SFD369		
:	:	
SFD500 : SFD509	Extension module configuration	Configuration of Extension Module 16
SFD510 : SFD519	BD module configuration	Configuration of BD Module 1
SFD520 : SFD529	BD module configuration	Configuration of BD Module 2
SFD530 : SFD539	ED module configuration	Configuration of ED Module 1

Communication

ID	Function	Note
SFD600	COM1 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD610	COM2 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD620	COM3 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD630	COM4 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD640	COM5 free format communication buffer bit numbers	0: 8-bit 1: 16-bit

Motion control

ID	function	Explanation
Y0 (common parameters)		
SFD900	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD901	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD902	Pulse number/1 rotation low 16-bit	
SFD903	Pulse number/1 rotation high 16-bit	
SFD904	Moving amount/1 rotation low 16-bit	
SFD905	Moving amount/1 rotation high 16-bit	
SFD906	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD907	Direction delay time	Default is 20, unit: ms
SFD908	Gear clearance positive compensation	
SFD909	Gear clearance negative compensation	
SFD910	Electrical origin position low 16-bit	
SFD911	Electrical origin position high 16-bit	
SFD912	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD913	Near-point signal terminal setting	

SFD914	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal
SFD915	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD917	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD918	Return speed VH low 16-bit	
SFD919	Return speed VH high 16-bit	
SFD922	Creeping speed VC low 16-bit	
SFD923	Creeping speed VC high 16-bit	
SFD924	Mechanical origin position low 16-bit	
SFD925	Mechanical origin position high 16-bit	
SFD926	Z phase number	
SFD927	CLR signal delay time	Default is 20, unit: ms
SFD928	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD929		High 16-bit
SFD930	Soft limit positive value	Low 16-bit
SFD931		High 16-bit
SFD932	Soft limit negative value	Low 16-bit
SFD933		High 16-bit
…		
Y0 (group 1 parameters)		
SFD950	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD951	Pulse default speed high 16-bit	
SFD952	Acceleration time of pulse default speed	
SFD953	deceleration time of pulse default speed	
SFD954	Accerlation and deceleration time	
SFD955	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD956	Max speed low 16-bit	
SFD957	Max speed high 16-bit	
SFD958	Initial speed low 16-bit	
SFD959	Initial speed high 16-bit	

SFD960	Stop speed low 16-bit	
SFD961	Stop speed high 16-bit	
SFD962	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD963	Follow feedforward compensation	0~100, %
…		
Y0 (group 2 parameters)		
SFD970	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD971	Pulse default speed high 16-bit	
SFD972	Acceleration time of pulse default speed	
SFD973	deceleration time of pulse default speed	
SFD974	Accerlation and deceleration time	
SFD975	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD976	Max speed low 16-bit	
SFD977	Max speed high 16-bit	
SFD978	Initial speed low 16-bit	
SFD979	Initial speed high 16-bit	
SFD980	Stop speed low 16-bit	
SFD981	Stop speed high 16-bit	
SFD982	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD983	Follow feedforward compensation	0~100, %
…		
Y0 (group 3 parameters)		
SFD990	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD991	Pulse default speed high 16-bit	
SFD992	Acceleration time of pulse default speed	
SFD993	deceleration time of pulse default speed	
SFD994	Accerlation and deceleration time	

SFD995	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD996	Max speed low 16-bit	
SFD997	Max speed high 16-bit	
SFD998	Initial speed low 16-bit	
SFD999	Initial speed high 16-bit	
SFD1000	Stop speed low 16-bit	
SFD1001	Stop speed high 16-bit	
SFD1002	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1003	Follow feedforward compensation	0~100, %
…		
Y0 (group 4 parameters)		
SFD1010	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1011	Pulse default speed high 16-bit	
SFD1012	Acceleration time of pulse default speed	
SFD1013	deceleration time of pulse default speed	
SFD1014	Accerlation and deceleration time	
SFD1015	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1016	Max speed low 16-bit	
SFD1017	Max speed high 16-bit	
SFD1018	Initial speed low 16-bit	
SFD1019	Initial speed high 16-bit	
SFD1020	Stop speed low 16-bit	
SFD1021	Stop speed high 16-bit	
SFD1022	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1023	Follow feedforward compensation	0~100, %
…		
Y1 (common parameters)		

SFD1030	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1031	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD1032	Pulse number/1 rotation low 16-bit	
SFD1033	Pulse number/1 rotation high 16-bit	
SFD1034	Moving amount/1 rotation low 16-bit	
SFD1035	Moving amount/1 rotation high 16-bit	
SFD1036	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1037	Direction delay time	Default is 20, unit: ms
SFD1038	Gear clearance positive compensation	
SFD1039	Gear clearance negative compensation	
SFD1040	Electrical origin position low 16-bit	
SFD1041	Electrical origin position high 16-bit	
SFD1042	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1044	Near-point signal terminal setting	
SFD1045	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD1047	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1048	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1049	Return speed VH low 16-bit	
SFD1052	Return speed VH high 16-bit	
SFD1053	Creeping speed VC low 16-bit	
SFD1054	Creeping speed VC high 16-bit	
SFD1055	Mechanical origin position low 16-bit	
SFD1056	Mechanical origin position high 16-bit	
SFD1057	Z phase number	
SFD1058	CLR signal delay time	Default is 20, unit: ms
SFD1059	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1060		High 16-bit
SFD1061	Soft limit positive value	Low 16-bit
SFD1062		High 16-bit
SFD1063	Soft limit negative value	Low 16-bit
…		
Y1 (group 1 parameters)		
SFD1080	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1081	Pulse default speed high 16-bit	
SFD1082	Acceleration time of pulse default speed	
SFD1083	deceleration time of pulse default speed	
SFD1084	Accerlation and deceleration time	
SFD1085	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1086	Max speed low 16-bit	
SFD1087	Max speed high 16-bit	
SFD1088	Initial speed low 16-bit	
SFD1089	Initial speed high 16-bit	
SFD1090	Stop speed low 16-bit	
SFD1091	Stop speed high 16-bit	

SFD1092	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1093	Follow feedforward compensation	0~100, %
…		
Y1 (group 2 parameters)		
SFD1100	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1101	Pulse default speed high 16-bit	
SFD1102	Acceleration time of pulse default speed	
SFD1103	deceleration time of pulse default speed	
SFD1104	Accerlation and deceleration time	
SFD1105	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1106	Max speed low 16-bit	
SFD1107	Max speed high 16-bit	
SFD1108	Initial speed low 16-bit	
SFD1109	Initial speed high 16-bit	
SFD1110	Stop speed low 16-bit	
SFD1111	Stop speed high 16-bit	
SFD1112	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1113	Follow feedforward compensation	0~100, %
…		
Y1 (group 3 parameters)		
SFD1120	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1121	Pulse default speed high 16-bit	
SFD1122	Acceleration time of pulse default speed	
SFD1123	deceleration time of pulse default speed	
SFD1124	Accerlation and deceleration time	
SFD1125	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1126	Max speed low 16-bit	

SFD1127	Max speed high 16-bit	
SFD1128	Initial speed low 16-bit	
SFD1129	Initial speed high 16-bit	
SFD1130	Stop speed low 16-bit	
SFD1131	Stop speed high 16-bit	
SFD1132	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1133	Follow feedforward compensation	0~100, %
…		
Y1 (group 4 parameters)		
SFD1140	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1141	Pulse default speed high 16-bit	
SFD1142	Acceleration time of pulse default speed	
SFD1143	deceleration time of pulse default speed	
SFD1144	Accerlation and deceleration time	
SFD1145	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1146	Max speed low 16-bit	
SFD1147	Max speed high 16-bit	
SFD1148	Initial speed low 16-bit	
SFD1149	Initial speed high 16-bit	
SFD1150	Stop speed low 16-bit	
SFD1151	Stop speed high 16-bit	
SFD1152	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1153	Follow feedforward compensation	0~100, %
…		
Y2 (common parameters)		

SFD1160	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1161	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD1162	Pulse number/1 rotation low 16-bit	
SFD1163	Pulse number/1 rotation high 16-bit	
SFD1164	Moving amount/1 rotation low 16-bit	
SFD1165	Moving amount/1 rotation high 16-bit	
SFD1166	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1167	Direction delay time	Default is 20, unit: ms
SFD1168	Gear clearance positive compensation	
SFD1169	Gear clearance negative compensation	
SFD1170	Electrical origin position low 16-bit	
SFD1171	Electrical origin position high 16-bit	
SFD1172	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1174	Near-point signal terminal setting	
SFD1175	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD1177	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1178	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1179	Return speed VH low 16-bit	
SFD1182	Return speed VH high 16-bit	
SFD1183	Creeping speed VC low 16-bit	
SFD1184	Creeping speed VC high 16-bit	
SFD1185	Mechanical origin position low 16-bit	
SFD1186	Mechanical origin position high 16-bit	
SFD1187	Z phase number	
SFD1188	CLR signal delay time	Default is 20, unit: ms
SFD1189	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1190	Soft limit positive value	High 16-bit
SFD1191		Low 16-bit
SFD1192	Soft limit negative value	High 16-bit
SFD1193		Low 16-bit
…		
Y2 (group 1 parameters)		
SFD1210	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1211	Pulse default speed high 16-bit	
SFD1212	Acceleration time of pulse default speed	
SFD1213	deceleration time of pulse default speed	
SFD1214	Accerlation and deceleration time	
SFD1215	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1216	Max speed low 16-bit	
SFD1217	Max speed high 16-bit	
SFD1218	Initial speed low 16-bit	
SFD1219	Initial speed high 16-bit	
SFD1220	Stop speed low 16-bit	
SFD1221	Stop speed high 16-bit	

SFD1222	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1223	Follow feedforward compensation	0~100, %
…		
Y2 (group 2 parameters)		
SFD1230	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1231	Pulse default speed high 16-bit	
SFD1232	Acceleration time of pulse default speed	
SFD1233	deceleration time of pulse default speed	
SFD1234	Accerlation and deceleration time	
SFD1235	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1236	Max speed low 16-bit	
SFD1237	Max speed high 16-bit	
SFD1238	Initial speed low 16-bit	
SFD1239	Initial speed high 16-bit	
SFD1240	Stop speed low 16-bit	
SFD1241	Stop speed high 16-bit	
SFD1242	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1243	Follow feedforward compensation	0~100, %
…		
Y2 (group 3 parameters)		
SFD1250	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1251	Pulse default speed high 16-bit	
SFD1252	Acceleration time of pulse default speed	
SFD1253	deceleration time of pulse default speed	
SFD1254	Accerlation and deceleration time	
SFD1255	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1256	Max speed low 16-bit	

SFD1257	Max speed high 16-bit	
SFD1258	Initial speed low 16-bit	
SFD1259	Initial speed high 16-bit	
SFD1260	Stop speed low 16-bit	
SFD1261	Stop speed high 16-bit	
SFD1262	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1263	Follow feedforward compensation	0~100, %
…		
Y2 (group 4 parameters)		
SFD1270	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1271	Pulse default speed high 16-bit	
SFD1272	Acceleration time of pulse default speed	
SFD1273	deceleration time of pulse default speed	
SFD1274	Accerlation and deceleration time	
SFD1275	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1276	Max speed low 16-bit	
SFD1277	Max speed high 16-bit	
SFD1278	Initial speed low 16-bit	
SFD1279	Initial speed high 16-bit	
SFD1280	Stop speed low 16-bit	
SFD1281	Stop speed high 16-bit	
SFD1282	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1283	Follow feedforward compensation	0~100, %
…		
Y3 (common parameters)		

SFD1290	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1291	Pulse sending mode	Bit 0： pulse sending mode 0： complete mode； 1： continue mode Default is 0
SFD1292	Pulse number/1 rotation low 16-bit	
SFD1293	Pulse number/1 rotation high 16-bit	
SFD1294	Moving amount/1 rotation low 16-bit	
SFD1295	Moving amount/1 rotation high 16-bit	
SFD1296	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1297	Direction delay time	Default is 20, unit: ms
SFD1298	Gear clearance positive compensation	
SFD1299	Gear clearance negative compensation	
SFD1300	Electrical origin position low 16-bit	
SFD1301	Electrical origin position high 16-bit	
SFD1302	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1304	Near-point signal terminal setting	
SFD1305	Z phase terminal setting	Bit0~Bit7： Specify the number of the X terminal, 0xFF is no terminal

SFD1307	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1308	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1309	Return speed VH low 16-bit	
SFD1312	Return speed VH high 16-bit	
SFD1313	Creeping speed VC low 16-bit	
SFD1314	Creeping speed VC high 16-bit	
SFD1315	Mechanical origin position low 16-bit	
SFD1316	Mechanical origin position high 16-bit	
SFD1317	Z phase number	
SFD1318	CLR signal delay time	Default is 20, unit: ms
SFD1319	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1320		High 16-bit
SFD1321	Soft limit positive value	Low 16-bit
SFD1322		High 16-bit
SFD1323	Soft limit negative value	Low 16-bit
…		
Y3 (group 1 parameters)		
SFD1340	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1341	Pulse default speed high 16-bit	
SFD1342	Acceleration time of pulse default speed	
SFD1343	deceleration time of pulse default speed	
SFD1344	Accerlation and deceleration time	
SFD1345	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1346	Max speed low 16-bit	
SFD1347	Max speed high 16-bit	
SFD1348	Initial speed low 16-bit	
SFD1349	Initial speed high 16-bit	
SFD1350	Stop speed low 16-bit	
SFD1351	Stop speed high 16-bit	

SFD1352	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1353	Follow feedforward compensation	0~100, %
…		
Y3 (group 2 parameters)		
SFD1360	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1361	Pulse default speed high 16-bit	
SFD1362	Acceleration time of pulse default speed	
SFD1363	deceleration time of pulse default speed	
SFD1364	Accerlation and deceleration time	
SFD1365	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1366	Max speed low 16-bit	
SFD1367	Max speed high 16-bit	
SFD1368	Initial speed low 16-bit	
SFD1369	Initial speed high 16-bit	
SFD1370	Stop speed low 16-bit	
SFD1371	Stop speed high 16-bit	
SFD1372	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1373	Follow feedforward compensation	0~100, %
…		
Y3 (group 3 parameters)		
SFD1380	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1381	Pulse default speed high 16-bit	
SFD1382	Acceleration time of pulse default speed	
SFD1383	deceleration time of pulse default speed	
SFD1384	Accerlation and deceleration time	
SFD1385	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1386	Max speed low 16-bit	

SFD1387	Max speed high 16-bit	
SFD1388	Initial speed low 16-bit	
SFD1389	Initial speed high 16-bit	
SFD1390	Stop speed low 16-bit	
SFD1391	Stop speed high 16-bit	
SFD1392	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1393	Follow feedforward compensation	0~100, %
…		
Y3 (group 4 parameters)		
SFD1400	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1401	Pulse default speed high 16-bit	
SFD1402	Acceleration time of pulse default speed	
SFD1403	deceleration time of pulse default speed	
SFD1404	Accerlation and deceleration time	
SFD1405	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1406	Max speed low 16-bit	
SFD1407	Max speed high 16-bit	
SFD1408	Initial speed low 16-bit	
SFD1409	Initial speed high 16-bit	
SFD1410	Stop speed low 16-bit	
SFD1411	Stop speed high 16-bit	
SFD1412	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1413	Follow feedforward compensation	0~100, %
…		
Y4 (common parameters)		

SFD1420	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1421	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD1422	Pulse number/1 rotation low 16-bit	
SFD1423	Pulse number/1 rotation high 16-bit	
SFD1424	Moving amount/1 rotation low 16-bit	
SFD1425	Moving amount/1 rotation high 16-bit	
SFD1426	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1427	Direction delay time	Default is 20, unit: ms
SFD1428	Gear clearance positive compensation	
SFD1429	Gear clearance negative compensation	
SFD1430	Electrical origin position low 16-bit	
SFD1431	Electrical origin position high 16-bit	
SFD1432	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1434	Near-point signal terminal setting	
SFD1435	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD1437	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1438	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1439	Return speed VH low 16-bit	
SFD1442	Return speed VH high 16-bit	
SFD1443	Creeping speed VC low 16-bit	
SFD1444	Creeping speed VC high 16-bit	
SFD1445	Mechanical origin position low 16-bit	
SFD1446	Mechanical origin position high 16-bit	
SFD1447	Z phase number	
SFD1448	CLR signal delay time	Default is 20, unit: ms
SFD1449	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1450	Soft limit positive value	High 16-bit
SFD1451		Low 16-bit
SFD1452	Soft limit negative value	High 16-bit
SFD1453		Low 16-bit
…		
Y4 (group 1 parameters)		
SFD1470	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1471	Pulse default speed high 16-bit	
SFD1472	Acceleration time of pulse default speed	
SFD1473	deceleration time of pulse default speed	
SFD1474	Accerlation and deceleration time	
SFD1475	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1476	Max speed low 16-bit	
SFD1477	Max speed high 16-bit	
SFD1478	Initial speed low 16-bit	
SFD1479	Initial speed high 16-bit	
SFD1480	Stop speed low 16-bit	
SFD1481	Stop speed high 16-bit	

SFD1482	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1483	Follow feedforward compensation	0~100, %
…		
Y4 (group 2 parameters)		
SFD1490	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1491	Pulse default speed high 16-bit	
SFD1492	Acceleration time of pulse default speed	
SFD1493	deceleration time of pulse default speed	
SFD1494	Accerlation and deceleration time	
SFD1495	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1496	Max speed low 16-bit	
SFD1497	Max speed high 16-bit	
SFD1498	Initial speed low 16-bit	
SFD1499	Initial speed high 16-bit	
SFD1500	Stop speed low 16-bit	
SFD1501	Stop speed high 16-bit	
SFD1502	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1503	Follow feedforward compensation	0~100, %
…		
Y4 (group 3 parameters)		
SFD1510	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1511	Pulse default speed high 16-bit	
SFD1512	Acceleration time of pulse default speed	
SFD1513	deceleration time of pulse default speed	
SFD1514	Accerlation and deceleration time	
SFD1515	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1516	Max speed low 16-bit	

SFD1517	Max speed high 16-bit	
SFD1518	Initial speed low 16-bit	
SFD1519	Initial speed high 16-bit	
SFD1520	Stop speed low 16-bit	
SFD1521	Stop speed high 16-bit	
SFD1522	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1523	Follow feedforward compensation	0~100, %
…		
Y4 (group 4 parameters)		
SFD1530	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1531	Pulse default speed high 16-bit	
SFD1532	Acceleration time of pulse default speed	
SFD1533	deceleration time of pulse default speed	
SFD1534	Accerlation and deceleration time	
SFD1535	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1536	Max speed low 16-bit	
SFD1537	Max speed high 16-bit	
SFD1538	Initial speed low 16-bit	
SFD1539	Initial speed high 16-bit	
SFD1540	Stop speed low 16-bit	
SFD1541	Stop speed high 16-bit	
SFD1542	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1543	Follow feedforward compensation	0~100, %
…		
Y5 (common parameters)		

SFD1550	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1551	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD1552	Pulse number/1 rotation low 16-bit	
SFD1553	Pulse number/1 rotation high 16-bit	
SFD1554	Moving amount/1 rotation low 16-bit	
SFD1555	Moving amount/1 rotation high 16-bit	
SFD1556	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1557	Direction delay time	Default is 20, unit: ms
SFD1558	Gear clearance positive compensation	
SFD1559	Gear clearance negative compensation	
SFD1560	Electrical origin position low 16-bit	
SFD1561	Electrical origin position high 16-bit	
SFD1562	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1564	Near-point signal terminal setting	
SFD1565	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD1567	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1568	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1569	Return speed VH low 16-bit	
SFD1572	Return speed VH high 16-bit	
SFD1573	Creeping speed VC low 16-bit	
SFD1574	Creeping speed VC high 16-bit	
SFD1575	Mechanical origin position low 16-bit	
SFD1576	Mechanical origin position high 16-bit	
SFD1577	Z phase number	
SFD1578	CLR signal delay time	Default is 20, unit: ms
SFD1579	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1580		High 16-bit
SFD1581	Soft limit positive value	Low 16-bit
SFD1582		High 16-bit
SFD1583	Soft limit negative value	Low 16-bit
…		
Y5 (group 1 parameters)		
SFD1600	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1601	Pulse default speed high 16-bit	
SFD1602	Acceleration time of pulse default speed	
SFD1603	deceleration time of pulse default speed	
SFD1604	Accerlation and deceleration time	
SFD1605	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1606	Max speed low 16-bit	
SFD1607	Max speed high 16-bit	
SFD1608	Initial speed low 16-bit	
SFD1609	Initial speed high 16-bit	
SFD1610	Stop speed low 16-bit	
SFD1611	Stop speed high 16-bit	

SFD1612	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1613	Follow feedforward compensation	0~100, %
…		
Y5 (group 2 parameters)		
SFD1620	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1621	Pulse default speed high 16-bit	
SFD1622	Acceleration time of pulse default speed	
SFD1623	deceleration time of pulse default speed	
SFD1624	Accerlation and deceleration time	
SFD1625	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1626	Max speed low 16-bit	
SFD1627	Max speed high 16-bit	
SFD1628	Initial speed low 16-bit	
SFD1629	Initial speed high 16-bit	
SFD1630	Stop speed low 16-bit	
SFD1631	Stop speed high 16-bit	
SFD1632	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1633	Follow feedforward compensation	0~100, %
…		
Y5 (group 3 parameters)		
SFD1640	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1641	Pulse default speed high 16-bit	
SFD1642	Acceleration time of pulse default speed	
SFD1643	deceleration time of pulse default speed	
SFD1644	Accerlation and deceleration time	
SFD1645	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1646	Max speed low 16-bit	

SFD1647	Max speed high 16-bit	
SFD1648	Initial speed low 16-bit	
SFD1649	Initial speed high 16-bit	
SFD1650	Stop speed low 16-bit	
SFD1651	Stop speed high 16-bit	
SFD1652	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1653	Follow feedforward compensation	0~100, %
…		
Y5 (group 4 parameters)		
SFD1660	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1661	Pulse default speed high 16-bit	
SFD1662	Acceleration time of pulse default speed	
SFD1663	deceleration time of pulse default speed	
SFD1664	Accerlation and deceleration time	
SFD1665	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1666	Max speed low 16-bit	
SFD1667	Max speed high 16-bit	
SFD1668	Initial speed low 16-bit	
SFD1669	Initial speed high 16-bit	
SFD1670	Stop speed low 16-bit	
SFD1671	Stop speed high 16-bit	
SFD1672	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1673	Follow feedforward compensation	0~100, %
…		
Y6 (common parameters)		

SFD1680	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1681	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD1682	Pulse number/1 rotation low 16-bit	
SFD1683	Pulse number/1 rotation high 16-bit	
SFD1684	Moving amount/1 rotation low 16-bit	
SFD1685	Moving amount/1 rotation high 16-bit	
SFD1686	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1687	Direction delay time	Default is 20, unit: ms
SFD1688	Gear clearance positive compensation	
SFD1689	Gear clearance negative compensation	
SFD1690	Electrical origin position low 16-bit	
SFD1691	Electrical origin position high 16-bit	
SFD1692	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1694	Near-point signal terminal setting	
SFD1695	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD1697	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1698	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1699	Return speed VH low 16-bit	
SFD1702	Return speed VH high 16-bit	
SFD1703	Creeping speed VC low 16-bit	
SFD1704	Creeping speed VC high 16-bit	
SFD1705	Mechanical origin position low 16-bit	
SFD1706	Mechanical origin position high 16-bit	
SFD1707	Z phase number	
SFD1708	CLR signal delay time	Default is 20, unit: ms
SFD1709	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1710		High 16-bit
SFD1711	Soft limit positive value	Low 16-bit
SFD1712		High 16-bit
SFD1713	Soft limit negative value	Low 16-bit
…		
Y6 (group 1 parameters)		
SFD1730	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1731	Pulse default speed high 16-bit	
SFD1732	Acceleration time of pulse default speed	
SFD1733	deceleration time of pulse default speed	
SFD1734	Accerlation and deceleration time	
SFD1735	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1736	Max speed low 16-bit	
SFD1737	Max speed high 16-bit	
SFD1738	Initial speed low 16-bit	
SFD1739	Initial speed high 16-bit	
SFD1740	Stop speed low 16-bit	
SFD1741	Stop speed high 16-bit	

SFD1742	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1743	Follow feedforward compensation	0~100, %
…		
Y6 (group 2 parameters)		
SFD1750	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1751	Pulse default speed high 16-bit	
SFD1752	Acceleration time of pulse default speed	
SFD1753	deceleration time of pulse default speed	
SFD1754	Accerlation and deceleration time	
SFD1755	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1756	Max speed low 16-bit	
SFD1757	Max speed high 16-bit	
SFD1758	Initial speed low 16-bit	
SFD1759	Initial speed high 16-bit	
SFD1760	Stop speed low 16-bit	
SFD1761	Stop speed high 16-bit	
SFD1762	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1763	Follow feedforward compensation	0~100, %
…		
Y6 (group 3 parameters)		
SFD1770	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1771	Pulse default speed high 16-bit	
SFD1772	Acceleration time of pulse default speed	
SFD1773	deceleration time of pulse default speed	
SFD1774	Accerlation and deceleration time	
SFD1775	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1776	Max speed low 16-bit	

SFD1777	Max speed high 16-bit	
SFD1778	Initial speed low 16-bit	
SFD1779	Initial speed high 16-bit	
SFD1780	Stop speed low 16-bit	
SFD1781	Stop speed high 16-bit	
SFD1782	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1783	Follow feedforward compensation	0~100, %
…		
Y6 (group 4 parameters)		
SFD1790	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1791	Pulse default speed high 16-bit	
SFD1792	Acceleration time of pulse default speed	
SFD1793	deceleration time of pulse default speed	
SFD1794	Accerlation and deceleration time	
SFD1795	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1796	Max speed low 16-bit	
SFD1797	Max speed high 16-bit	
SFD1798	Initial speed low 16-bit	
SFD1799	Initial speed high 16-bit	
SFD1800	Stop speed low 16-bit	
SFD1801	Stop speed high 16-bit	
SFD1802	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1803	Follow feedforward compensation	0~100, %
…		
Y7 (common parameters)		

SFD1810	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1811	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD1812	Pulse number/1 rotation low 16-bit	
SFD1813	Pulse number/1 rotation high 16-bit	
SFD1814	Moving amount/1 rotation low 16-bit	
SFD1815	Moving amount/1 rotation high 16-bit	
SFD1816	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1817	Direction delay time	Default is 20, unit: ms
SFD1818	Gear clearance positive compensation	
SFD1819	Gear clearance negative compensation	
SFD1820	Electrical origin position low 16-bit	
SFD1821	Electrical origin position high 16-bit	
SFD1822	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1824	Near-point signal terminal setting	
SFD1825	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD1827	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1828	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1829	Return speed VH low 16-bit	
SFD1832	Return speed VH high 16-bit	
SFD1833	Creeping speed VC low 16-bit	
SFD1834	Creeping speed VC high 16-bit	
SFD1835	Mechanical origin position low 16-bit	
SFD1836	Mechanical origin position high 16-bit	
SFD1837	Z phase number	
SFD1838	CLR signal delay time	Default is 20, unit: ms
SFD1839	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1840		High 16-bit
SFD1841	Soft limit positive value	Low 16-bit
SFD1842		High 16-bit
SFD1843	Soft limit negative value	Low 16-bit
…		
Y7 (group 1 parameters)		
SFD1860	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1861	Pulse default speed high 16-bit	
SFD1862	Acceleration time of pulse default speed	
SFD1863	deceleration time of pulse default speed	
SFD1864	Accerlation and deceleration time	
SFD1865	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1866	Max speed low 16-bit	
SFD1867	Max speed high 16-bit	
SFD1868	Initial speed low 16-bit	
SFD1869	Initial speed high 16-bit	
SFD1870	Stop speed low 16-bit	
SFD1871	Stop speed high 16-bit	

SFD1872	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1873	Follow feedforward compensation	0~100, %
…		
Y7 (group 2 parameters)		
SFD1880	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1881	Pulse default speed high 16-bit	
SFD1882	Acceleration time of pulse default speed	
SFD1883	deceleration time of pulse default speed	
SFD1884	Accerlation and deceleration time	
SFD1885	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1886	Max speed low 16-bit	
SFD1887	Max speed high 16-bit	
SFD1888	Initial speed low 16-bit	
SFD1889	Initial speed high 16-bit	
SFD1890	Stop speed low 16-bit	
SFD1891	Stop speed high 16-bit	
SFD1892	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1893	Follow feedforward compensation	0~100, %
…		
Y7 (group 3 parameters)		
SFD1900	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1901	Pulse default speed high 16-bit	
SFD1902	Acceleration time of pulse default speed	
SFD1903	deceleration time of pulse default speed	
SFD1904	Accerlation and deceleration time	
SFD1905	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1906	Max speed low 16-bit	

SFD1907	Max speed high 16-bit	
SFD1908	Initial speed low 16-bit	
SFD1909	Initial speed high 16-bit	
SFD1910	Stop speed low 16-bit	
SFD1911	Stop speed high 16-bit	
SFD1912	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1913	Follow feedforward compensation	0~100, %
…		
Y7 (group 4 parameters)		
SFD1920	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1921	Pulse default speed high 16-bit	
SFD1922	Acceleration time of pulse default speed	
SFD1923	deceleration time of pulse default speed	
SFD1924	Accerlation and deceleration time	
SFD1925	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1926	Max speed low 16-bit	
SFD1927	Max speed high 16-bit	
SFD1928	Initial speed low 16-bit	
SFD1929	Initial speed high 16-bit	
SFD1930	Stop speed low 16-bit	
SFD1931	Stop speed high 16-bit	
SFD1932	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD1933	Follow feedforward compensation	0~100, %
…		
Y10 (common parameters)		

SFD1940	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD1941	Pulse sending mode	Bit 0： pulse sending mode 0： complete mode； 1： continue mode Default is 0
SFD1942	Pulse number/1 rotation low 16-bit	
SFD1943	Pulse number/1 rotation high 16-bit	
SFD1944	Moving amount/1 rotation low 16-bit	
SFD1945	Moving amount/1 rotation high 16-bit	
SFD1946	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD1947	Direction delay time	Default is 20, unit: ms
SFD1948	Gear clearance positive compensation	
SFD1949	Gear clearance negative compensation	
SFD1950	Electrical origin position low 16-bit	
SFD1951	Electrical origin position high 16-bit	
SFD1952	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD1954	Near-point signal terminal setting	
SFD1955	Z phase terminal setting	Bit0~Bit7： Specify the number of the X terminal, 0xFF is no terminal

SFD1957	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD1958	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD1959	Return speed VH low 16-bit	
SFD1962	Return speed VH high 16-bit	
SFD1963	Creeping speed VC low 16-bit	
SFD1964	Creeping speed VC high 16-bit	
SFD1965	Mechanical origin position low 16-bit	
SFD1966	Mechanical origin position high 16-bit	
SFD1967	Z phase number	
SFD1968	CLR signal delay time	Default is 20, unit: ms
SFD1969	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD1970	Soft limit positive value	High 16-bit
SFD1971		Low 16-bit
SFD1972	Soft limit negative value	High 16-bit
SFD1973		Low 16-bit
…		
Y10 (group 1 parameters)		
SFD1990	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD1991	Pulse default speed high 16-bit	
SFD1992	Acceleration time of pulse default speed	
SFD1993	deceleration time of pulse default speed	
SFD1994	Accerlation and deceleration time	
SFD1995	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD1996	Max speed low 16-bit	
SFD1997	Max speed high 16-bit	
SFD1998	Initial speed low 16-bit	
SFD1999	Initial speed high 16-bit	
SFD2000	Stop speed low 16-bit	
SFD2001	Stop speed high 16-bit	

SFD2002	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2003	Follow feedforward compensation	0~100, %
…		
Y10 (group 2 parameters)		
SFD2010	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2011	Pulse default speed high 16-bit	
SFD2012	Acceleration time of pulse default speed	
SFD2013	deceleration time of pulse default speed	
SFD2014	Accerlation and deceleration time	
SFD2015	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2016	Max speed low 16-bit	
SFD2017	Max speed high 16-bit	
SFD2018	Initial speed low 16-bit	
SFD2019	Initial speed high 16-bit	
SFD2020	Stop speed low 16-bit	
SFD2021	Stop speed high 16-bit	
SFD2022	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2023	Follow feedforward compensation	0~100, %
…		
Y10 (group 3 parameters)		
SFD2030	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2031	Pulse default speed high 16-bit	
SFD2032	Acceleration time of pulse default speed	
SFD2033	deceleration time of pulse default speed	
SFD2034	Accerlation and deceleration time	
SFD2035	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2036	Max speed low 16-bit	

SFD2037	Max speed high 16-bit	
SFD2038	Initial speed low 16-bit	
SFD2039	Initial speed high 16-bit	
SFD2040	Stop speed low 16-bit	
SFD2041	Stop speed high 16-bit	
SFD2042	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2043	Follow feedforward compensation	0~100, %
…		
Y10 (group 4 parameters)		
SFD2050	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2051	Pulse default speed high 16-bit	
SFD2052	Acceleration time of pulse default speed	
SFD2053	deceleration time of pulse default speed	
SFD2054	Accerlation and deceleration time	
SFD2055	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2056	Max speed low 16-bit	
SFD2057	Max speed high 16-bit	
SFD2058	Initial speed low 16-bit	
SFD2059	Initial speed high 16-bit	
SFD2060	Stop speed low 16-bit	
SFD2061	Stop speed high 16-bit	
SFD2062	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2063	Follow feedforward compensation	0~100, %
…		
Y11 (common parameters)		

SFD2070	Pulse parameters	Bit 1: Pulse Direction Logic 0: positive logic, 1: negative logic; default is 0 Bit 2: Soft Limit 0: Not enabled, 1: enabled; default is 0 Bit 3: direction of mechanical return to origin 0: Negative, 1: Positive; Default is 0 Bit 10~8: Pulse Unit Bit 8: 0: Number of Pulses, 1:Equivalent 000: Number of pulses 001: 1 μm 011: 0.01 μm 101: 0.1 μm 111: 1 mm The default is 000. Bit15: Interpolated coordinate mode 0: Cross coordinates, 1: Polar coordinates; The default is 0.
SFD2071	Pulse sending mode	Bit 0：pulse sending mode 0：complete mode；1：continue mode Default is 0
SFD2072	Pulse number/1 rotation low 16-bit	
SFD2073	Pulse number/1 rotation high 16-bit	
SFD2074	Moving amount/1 rotation low 16-bit	
SFD2075	Moving amount/1 rotation high 16-bit	
SFD2076	Pulse direction terminal	Appoint to Y terminal, 0xFF is no terminal
SFD2077	Direction delay time	Default is 20, unit: ms
SFD2078	Gear clearance positive compensation	
SFD2079	Gear clearance negative compensation	
SFD2080	Electrical origin position low 16-bit	
SFD2081	Electrical origin position high 16-bit	
SFD2082	Signal terminal switch state	Bit0: Origin Signal Switch State Settings Bit1:Z Phase Switch State Settings Bit2: Positive Limit Switching State Settings Bit3: Negative Limit Switching State Settings 0: Normally open (positive logic), 1: Normally closed (negative logic); default is 0
SFD2084	Near-point signal terminal setting	
SFD2085	Z phase terminal setting	Bit0~Bit7：Specify the number of the X terminal, 0xFF is no terminal

SFD2087	Limit terminal setting	Bit7~Bit0：Specifies the X terminal number of the positive limit, and 0xFF is no terminal. Bit15~Bit8：Specifies the X terminal number of the negative limit, and 0xFF is no terminal.
SFD2088	Zero clear CLR output signal	Bit0~Bit7：Specify the number of the Y terminal, 0xFF is no terminal
SFD2089	Return speed VH low 16-bit	
SFD2092	Return speed VH high 16-bit	
SFD2093	Creeping speed VC low 16-bit	
SFD2094	Creeping speed VC high 16-bit	
SFD2095	Mechanical origin position low 16-bit	
SFD2096	Mechanical origin position high 16-bit	
SFD2097	Z phase number	
SFD2098	CLR signal delay time	Default is 20, unit: ms
SFD2099	Grinding wheel radius (polar coordinates)	Low 16-bit
SFD2100	Soft limit positive value	High 16-bit
SFD2101		Low 16-bit
SFD2102	Soft limit negative value	High 16-bit
SFD2103		Low 16-bit
…		
Y11 (group 1 parameters)		
SFD2120	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2121	Pulse default speed high 16-bit	
SFD2122	Acceleration time of pulse default speed	
SFD2123	deceleration time of pulse default speed	
SFD2124	Accerlation and deceleration time	
SFD2125	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2126	Max speed low 16-bit	
SFD2127	Max speed high 16-bit	
SFD2128	Initial speed low 16-bit	
SFD2129	Initial speed high 16-bit	
SFD2130	Stop speed low 16-bit	
SFD2131	Stop speed high 16-bit	

SFD2132	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2133	Follow feedforward compensation	0~100, %
…		
Y11 (group 2 parameters)		
SFD2140	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2141	Pulse default speed high 16-bit	
SFD2142	Acceleration time of pulse default speed	
SFD2143	deceleration time of pulse default speed	
SFD2144	Accerlation and deceleration time	
SFD2145	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2146	Max speed low 16-bit	
SFD2147	Max speed high 16-bit	
SFD2148	Initial speed low 16-bit	
SFD2149	Initial speed high 16-bit	
SFD2150	Stop speed low 16-bit	
SFD2151	Stop speed high 16-bit	
SFD2152	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2153	Follow feedforward compensation	0~100, %
…		
Y11 (group 3 parameters)		
SFD2160	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2161	Pulse default speed high 16-bit	
SFD2162	Acceleration time of pulse default speed	
SFD2163	deceleration time of pulse default speed	
SFD2164	Accerlation and deceleration time	
SFD2165	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2166	Max speed low 16-bit	

SFD2167	Max speed high 16-bit	
SFD2168	Initial speed low 16-bit	
SFD2169	Initial speed high 16-bit	
SFD2170	Stop speed low 16-bit	
SFD2171	Stop speed high 16-bit	
SFD2172	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2173	Follow feedforward compensation	0~100, %
…		
Y11 (group 4 parameters)		
SFD2180	Pulse default speed low 16-bit	Pulse is sent at the default speed when the speed is 0.
SFD2181	Pulse default speed high 16-bit	
SFD2182	Acceleration time of pulse default speed	
SFD2183	deceleration time of pulse default speed	
SFD2184	Accerlation and deceleration time	
SFD2185	Acceleration/deceleration mode	Bit1~Bit0：acc/dec mode 00：linear acc/dec 01：S curve acc/dec 10：sine curve acc/dec 11：reserved Bit15~Bit2：reserved
SFD2186	Max speed low 16-bit	
SFD2187	Max speed high 16-bit	
SFD2188	Initial speed low 16-bit	
SFD2189	Initial speed high 16-bit	
SFD2190	Stop speed low 16-bit	
SFD2191	Stop speed high 16-bit	
SFD2192	Follow performance	1~100, 100 means the time constant is 1 Tick, 1 means the time constant is 100 Ticks
SFD2193	Follow feedforward compensation	0~100, %
…		

Appendix 4． External interruption terminal list

XD series PLC external interrupt terminal allocation is as follows:

XD2/XD3/XD5/XL1/XL3 series 16 I/O

Input terminal	Pointer		Disable interruption instruction
	Rising interruption	Falling interruption	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052
X5	I0300	I0301	SM053
X6	I0400	I0401	SM054
X7	I0500	I0501	SM055

XD2/XD3/XD5 series 24/32/48/60I/O, XDM series 24/32/60I/O, XDC series 24/32/48/60I/O XD5E series 30/60I/O, XDME series 60I/O, XL5/XL5E/XLME series 32 I/O

Input terminal	Pointer		Disable interruption instruction
	Rising interruption	Falling interruption	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052
X5	I0300	I0301	SM053
X6	I0400	I0401	SM054
X7	I0500	I0501	SM055
X10	I0600	I0601	SM056
X11	I0700	I0701	SM057
X12	I0800	I0801	SM058
X13	I0900	I0901	SM059

Appendix 5. PLC resource conflict table

When PLC is used in practice, conflicts may arise due to the simultaneous use of some resources. This section will list the resources that may cause conflicts in each PLC model. This part mainly refers to high-speed counting, accurate timing and pulse output.

	Precise timing	High speed counter				Pulse output	
XD2-16, XD3-16, XD5-16, XL3-16							
	ET0	-	-	-	-	-	-
	ET2						
	ET4						
	ET6						
	ET8	HSC0					
	ET10		HSC2				
	ET12			HSC4			
	ET14					Y0	
	ET16					Y0	
	ET18					Y1	
	ET20					Y1	
	ET22						
	ET24						
XD3-24/32/48/60, ZG3-30							
	ET0						
	ET2						
	ET4						
	ET6						
	ET8						
	ET10						
	ET12	HSC0					
	ET14		HSC2				
	ET16			HSC4			
	ET18					Y0	
	ET20					Y0	
	ET22					Y1	
	ET24					Y1	
XD5-24/32/48/60, XDM-24/32/48/60, XD5E-30/60, XDME-60, XL5-32, XL5E-32, XLME-32							
	ET0	-	-	-	-	-	-
	ET2				HSC6		
	ET4			HSC4			
	ET6		HSC2				
	ET8	HSC0					

	ET10					Y3	
	ET12					Y3	
	ET14					Y2	
	ET16					Y2	
	ET18					Y1	
	ET20					Y1	
	ET22					Y0	
	ET24					Y0	
XDC-24/32/48/60							
	ET0	-	-	-	HSC6	-	-
	ET2			HSC4			
	ET4		HSC2				
	ET6	HSC0					
	ET8					Y3	
	ET10					Y3	
	ET12					Y2	
	ET14					Y2	
	ET16					Y1	
	ET18					Y1	
	ET20					Y0	
	ET22					Y0	
	ET24						

※1: This form should be read horizontally. Any two resources in each row cannot be used at the same time. Otherwise, it will cause conflict.

WE CHAT ID

WUXI XINJE ELECTRIC CO., LTD.

4th Floor Building 7,Originality Industry park, Liyuan Development Zone, Wuxi City, Jiangsu Province

214072

Tel: 400-885-0136

Fax: (510) 85111290

XD/XL series PLC
User manual [Instruction]

WUXI XINJE ELECTRIC CO., LTD.

Data No. PD05 20190410 3.5

XINJE

XD/XL series PLC
User manual［Instruction］

- Basic explanation

Thank you for purchasing Xinje XD/XL series PLC.
This manual mainly introduces XD/XL series PLC instructions.
Please read this manual carefully before using and wire after understanding the content.
About software and programming instructions, please refer to related manuals.
Please hand this manual over to operation users.

- Notices for users

Only experienced operator can wire the plc. If any problem, please contact our technical department.
The listed examples are used to help users to understand, so it may not act.
Please conform that PLC specifications and principles are suitable when connect PLC to other products. Please conform safety of PLC and machines by yourself when use the PLC. Machines may be damaged by PLC errors.

- Responsibility declaration

The manual content has been checked carefully, however, mistakes may happen.
We often check the manual and will correct the problems in subsequent version. Welcome to offer advices to us.
Excuse us that we will not inform you if manual is changed.

- Contact information

If you have any problem about products, please contact the agent or Xinje company.
Tel: 0086 510-85134136 85123803
Fax: 0086 510-85111290
Address: Building 7 fourth floor, No.100, Dicui Rd, Wuxi, China.
Code : 214072

Catalog

1 Programming Summary

XD/XL series PLC accept the signal and execute the program in the controller, to fulfill the requirements of the users. This chapter introduces the PLC features, two kinds of programming language and etc.

1-1. PLC Features

Programming Language

XD/XL series PLC support two kinds of program language, instruction and ladder chart, the two kinds of language can convert to each other.

Security of the Program

To avoid the stolen or wrong modifying of user program, we encrypt the program. When uploading the encrypted program, it will check in the form of password. This can protect the user copyright; meanwhile, it limits the downloading, to avoid change program by mistake. XD/XL series added new register FS. (For different XD/XL models, please check the Data monitor in XDPpro software for FS register range, common range is FS0~FS47). FS value can be modified but cannot be read through Modbus instruction. FS cannot be compared to register but only constant in XDPpro software. The value cannot be read. FS is used to protect the user's copyright. The register D, HD… can replace by FS.

Program comments

When the user program is too long, the comments of program and soft components are necessary in order to change the program easily later.

Offset Function

Add offset appendix (like X3[D100], M10[D100], D0[D100]) after coils, data registers can make indirect addressing. For example, when D100=9, X3[D100] =X[3+9]=X14; M10[D100]=M19, D0[D100]=D9

Rich Basic Functions

XD/XL series PLC has enough basic instructions including basic sequential control, data moving and comparing, arithmetic operation, logic control, data loop and shift etc.
XD/XL series PLC also support interruption, high speed pulse, frequency testing, precise time, PID control and so on.

C Language Function Block

XD/XL series PLC support C language; users can call the C program in ladder chart. This function improves the programming efficiency.

Stop PLC when reboot

XD/XL series PLC support "Stop PLC when reboot" function. When there is a serious problem during PLC running, this method can stop all output immediately. Besides, if the COM port parameters are changed by mistake, this function can help PLC connect to the PC.

Communication Function

XD/XL series PLC has many communication modes, such as Modbus-RTU, Modbus-ASCII. When the COM port parameters are changed, the new parameters will be valid immediately without restarting the PLC.
Wait time can be added before Modbus instructions.

1-2. Programming Language

1-2-1. Type

XD/XL series PLC support two types of programming language:

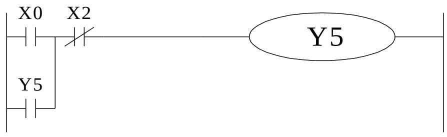
Instruction

Make the program with instructions directly, such as "LD", "AND", "OUT" etc. This is the basic input form of the programs, but it's hard to read and understand;

E.g.:

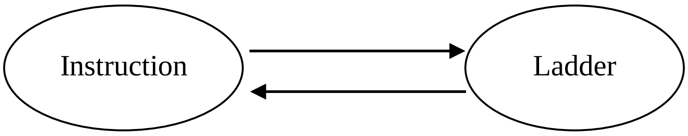
step	instruction	operand
0	LD	X000
1	OR	Y005
2	ANI	X002
3	OUT	Y005

Make sequential control graph with sequential control signal and soft components. This method is called "Ladder chart". This method uses coils and contactors to represent sequential circuit. The ladder chart is easy to understand and can be used to monitor the PLC status online.
E.g.:



1-2-2. Alternation

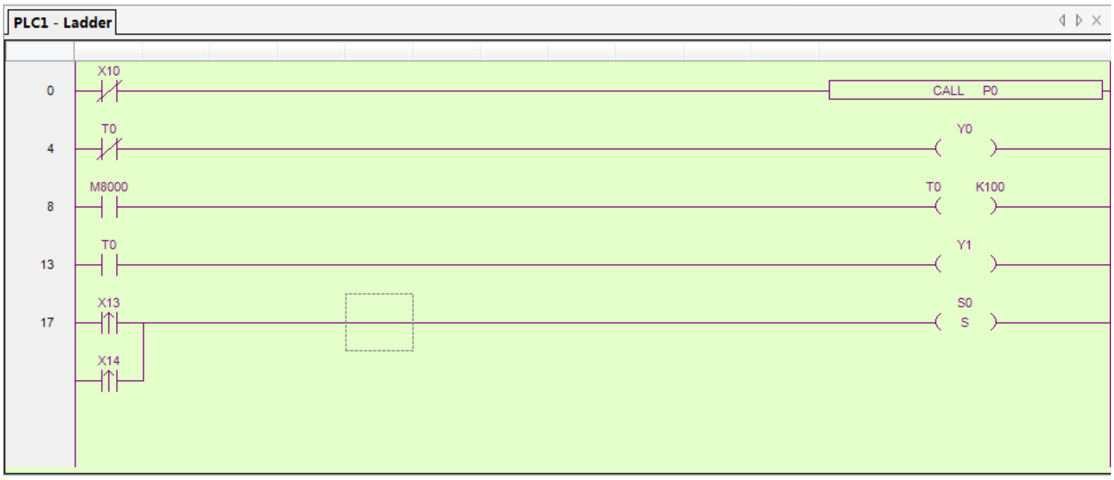
The two kinds of programming language can be transformed to each other.



1-3. Programming mode

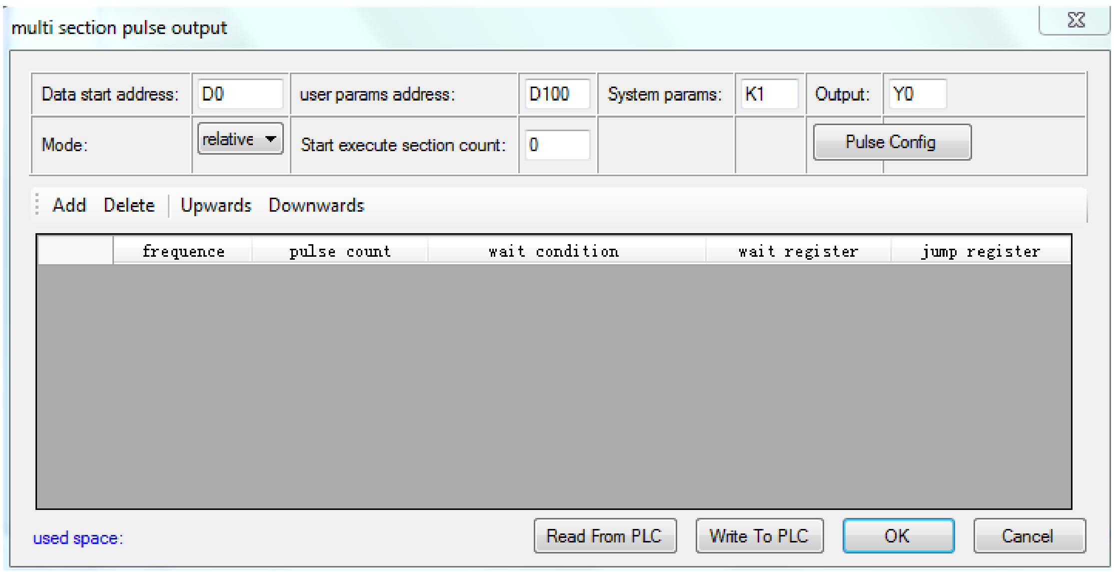
Direct Input

The two kinds of programming language can be input directly in the editing window. The ladder chart window has hint function which improves the programming efficiency greatly.



Instruction Configuration

Some instruction is complicated to use, like pulse output, PID etc. XDPPro software has the configuration window for these special instructions. User just needs to input parameters in the configuration window without remembering complicated instructions. The following window is multi section pulse output.



For the details of instruction configuration, please refer to XD/XL series PLC user manual 【software part】.

2 Soft Component Function

In chapter 1, we briefly introduce the programming language. However, the most important element in a program is the operands. These elements include the relays and registers. In this chapter, we will describe the functions and using methods of these relays and registers.

2-1. Summary of the Soft Components

There are many relays, timers and counters inside PLC. They all have countless NO (Normally ON) and NC (Normally Closed) contactors. Connect these contactors with the coils will make a sequential control circuit. Next we will introduce these soft components.

Input Relay (X)

- The functions of input relays

The input relays are used to receive the external ON/OFF signal, the sign is **X**.

- Address Assignment Principle
  - In each basic unit, X address is in the form of octal, such as X0~X7, X10~X17 …
  - The extension module address: module 1 starts from X10000, module 2 starts from X10100… XD1/XD2/XL1 cannot support extension module. Up to 10 extension modules can be connected to the XD3/XL3 main unit. XD5/XDM/XDC/XD5E/XDME/XL5/XL5E/XLME can connect 16 extension modules.
  - Extension BD board: BD 1 starts from X20000; The 24-32 points PLC can connect one extended BD board and the 48-60 points PLC can connect two extended BD boards. (16-point PLC does not support extended BD board, XL series does not support extended BD board.)
  - The address number of the left extended ED module, starting from X30000 according to octal system, XD/XL series PLC supports a left extended I/O ED module.
- Using notes

The digital filter is used in the input filter of the input relay. Users can change the filter parameters by setting the special register SFD0, default value is 10ms, modification range: 0 ~ 1000ms.

There are enough input relays in the PLC. The input relay whose address is more than input points can be seemed to auxiliary relay.

Output Relay (Y)

- Function of the output relays

Output relays are the interface to drive the external loads, the sign is **Y**;

- Address Assignment Principle

In each basic unit, Y address is in the form of octal, such as Y0~Y7, Y10~Y17 …

The extension module address: module 1 starts from Y10000, module 2 starts from Y10100… XD1/XD2/XL1 does not support extension modules, XD3/XL3 can accept 10 extension modules, XD5/XDM/XDC/XD5E/XDME/XL5/XL5E/XLME can accept 16 extension modules.

Expanding the address number of BD board, starting from X20000 according to octal system, 24-32 points PLC can extend one BD board, 48-60 points PLC can extend two BD boards. (16-point PLC does not support extended BD board, XL series does not support extended BD board.)
The address number of the left extended ED module, starting from Y30000 according to octal system, XD/XL series PLC supports a left extended input and output ED module.

Using notes
There are enough output relays in the PLC. The output relay whose address is more than output points can be seemed to auxiliary relay.

Auxiliary Relays (M, HM)

- Function of Auxiliary Relays

Auxiliary relays is internal relays of PLC, the sign is M and HM;

- Address assignment principle

In basic units, assign the auxiliary address in decimal form

- Using notes

This type of relays are different from the input/output relays, they can't drive external load and receive external signal, but only be used in the program;
Retentive relays can keep its ON/OFF status when PLC power OFF;

Status Relays (S, HS)

- Function of status relays

Used as relays in Ladder, the sign is S, HS.

- Address assignment principle

In basic units, assign the address in decimal form.

- Using notes

If it is not used as operation number, they can be used as auxiliary relays, programming as normal contactors/coils. Besides, they can be used as signal alarms, for external diagnose.

Timer (T, HT)

- Function of the timers

Timers are used to accumulate the time pulse like 1ms, 10ms, 100ms etc. when reach the set value, the output contactors acts, represent sign is T and HT.

- Address assignment principle

In basic units, assign the timer address in decimal form. Please refer to chapter 2-2 for details.

- Time pulse

There are three timer pulses: 1ms, 10ms, and 100ms. For example, 10ms means accumulate 10ms pulses.

- Accumulation/not accumulation

The timer has two modes: accumulation timer means even the timer drive coil is OFF, the timer will still keep the current value; while the not accumulation timer means when the accumulation value reaches the set value, the output acts, the accumulation value reset to 0.

Counter (C, HC)

According to different application purposes, the counters contain different types:

- For internal counting (for general using/power off retentive usage)

16 bits counter: for increment count, the count range is 1~32,767
32 bits counter: for increment count, the count range is 1~2,147,483,647
These counters are for PLC internal signal. The response speed is one scan cycle or longer.

- For High Speed Counting (Power-off retentive)

32 bits counter: the count range is -2,147,483,648~ +2,147,483,647
(Single phase increment count, AB phase count). For special input terminals.
The high speed counter will not be affected by PLC scanning period. For increment mode, it can count max 80KHz pulses; for AB phase mode, it can count max 50KHz pulses.

- Address assignment principle

In basic units, assign the timer address in decimal form.

Data Register (D, HD)

- Function of Data Registers

Data Registers are used to store data, the sign is D and HD.

- Address assignment principle

The data registers in XD/XL series PLC are 16 bits (the highest bit is sign bit), combine two data registers together is for 32 bits (the highest bit is sign bit) data processing.

- Using notes

Same to other soft components, data registers also have common type and power-off retentive type.

FlashROM Register (FD)

- Function of FlashROM registers

FlashROM registers are used to store data, the sign is FD.

- Address assignment principle

In basic units, FlashROM registers address is in form of decimal;

- Using notes

Even the battery powered off, this area can remember the data. So this area can store important parameters. FlashROM can be writen for about 1,000,000 times, and it takes time when writing. Frequently writing can cause permanent damage for FD.

Special secret Register (FS)

- The Function of Secret Register

A part of the FlashROM register is used to store data in soft components, which are represented by the symbol FS. The values in the FS register can be written but can not be read, so they can be used to protect the intellectual property rights of users.

- Address Allocation Principle

In the basic unit, FS registers are addressed in decimal numbers.

- Since the number of FS registers of different types of PLC may be different, please refer to the "PLC Initial Settings" shown in the online PLC software, generally FS0-FS47.
- Attention Points in Use

The storage area can remember data even if the battery is powered down, so it can be used to store important process parameters. FS can be written about 1,000,000 times, and it takes more time to write each time. Frequent writing will cause permanent damage to FS, so it is not recommended that users write frequently. When using MOV instruction to transmit data to FS, the rising edge is valid.

- The value of the soft element can be set arbitrarily in the FS register, but the value of the register can not be read (always returned to 0); and it can not be compared with the register in the PLC software, only with the constant, so the actual value of the register can not be read.

Constant (B) (K) (H)

B means Binary, K represents Decimal, H represents Hexadecimal. They are used to set timers and counters value, or operands of application instructions. For example hex FF will be HFF.

2-2. Structure of Soft Components

2-2-1. Structure of Memory

In XD/XL series PLC, there are many registers. Besides D, HD, FlashROM registers, we can also combine bit to register.

Data Register D, HD

For common use, 16 bits

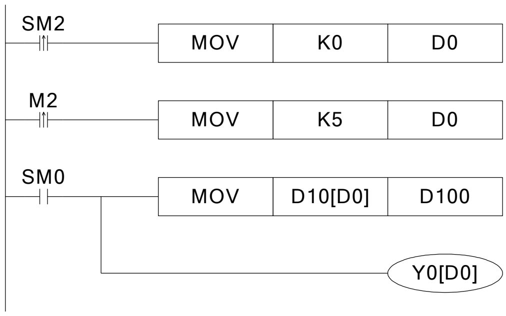
For common use, 32 bits (combine two continuous 16-bits registers)

For power off retentive use, cannot modify the retentive range

For special use, occupied by the system, can’t be used to common instruction parameters

For offset use (indirect assignment)

Form: Dn[Dm], HDn[Dm], Xn[Dm] , Yn[Dm] , Mn[Dm] , etc.



When D0=0, D100=D10, Y0 is ON.

When M2 turns from OFF to ON, D0=5, then D100=D15, Y5 is ON.

Therein, D10[D0]=D[10+D0], Y0[D0]=Y[0+D0].

The word offset combined by bit: DXn[Dm] represents DX[n+Dm].

The soft components with offset, the offset can represent by soft component D, HD.

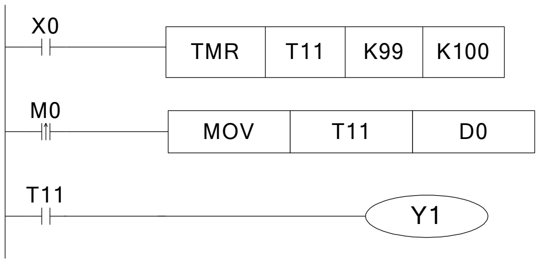
Timer T, HT/Counter C, HC

For common usage, 16 bits, represent the current value of timer/counter;

For common usage, 32 bits, (combine two continuous 16 bits registers)

To represent them, just use the letter+address method, such as T10, C11, HT10, HC11.

E.g.



In the above example, MOV T11 D0, T11 represents word register;

LD T11, T11 represents bit register.

FlashROM Register FD

For power off retentive usage, 16 bits

For power off retentive usage, 32 bits, (combine two continuous 16 bits registers)

For special usage, occupied by the system, can’t be used as common instruction parameters

Register combined by bits

For common usage, 16 bits, (combine 16 bits)

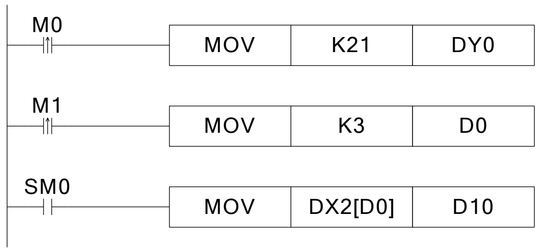
The soft components which can be combined to words are: X, Y, M, S, T, C, HM, HS, HT, HC.

Format: add “D” in front of soft components, like DM10, represents a 16-bits register from M10~M25

Get 16 bits beginning from DXn, cannot beyond the soft components range;

The word combined by bits cannot do bit addressing;

E.g.:



When M0 changes from OFF to ON, the value in the word which is combined by Y0~Y17 equals to 21, i.e. Y0, Y2, Y4 become ON.
Before M1 activates, if D0=0, DX2[D0] represents a word combined by X2~X21.
If M1 changes from OFF to ON, D0=3, then DX2[D0] represents a word combined by X5~X24.

2-2-2. Structure of Bit Soft Components

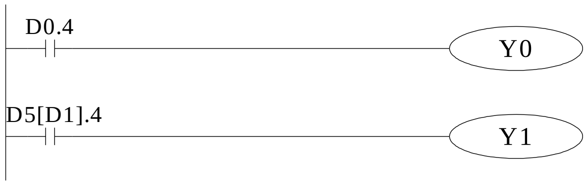
Bit soft components include X, Y, M, S, T, C, HM, HS, HT, HC. Besides, the bit of the register also can be used as bit sofst component.

Relay

Input Relay X, octal form
Output Relay Y, octal form
Auxiliary Relay M, HM, S, HS; decimal form
Auxiliary Relay T, HT, C, HC, decimal form. The represent method is same to registers, so we need to judge if it's word register or bit register according to the instruction.

The Bit of register

Composed by bit of register, support register D
Represent method: Dn.m (0≤m≤15): for example D10.2 means the second bit of D10
The represent method of bit with offset: Dn[Dm].x
Bit of register can't compose to word soft component again;
E.g.:



D0.4 means when the fourth bit of D0 is 1, set Y0 ON.
D5[D1].4 means bit addressing with offset, if D1=5, then D5[D1] means the fourth bit of D10

2-3. Soft Components List

2-3-1. Soft Components List

XD1 series PLC soft components list:

	Name	Range		Points	
		16 I/O	32 I/O	16	32
X	Input points	X0~X7	X0~X17	8	16
Y	Output points	Y0~Y7	Y0~Y17	8	16
X	Input points[※3]	X10000～X10077（#1 expansion module） …… X11100～X11177（#10 expansion module）		640	
Y	Output points[※3]	Y10000～Y10077（#1 expansion module） …… Y11100～Y11177（#10 expansion module）		640	
X	Input points[※4]	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）		128	
Y	Output points[※4]	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）		128	
X	Input points[※5]	X30000～X30077（#1 expansion ED）		64	
Y	Output points[※5]	Y30000～Y30077（#1 expansion ED）		64	
M	Internal relay	M0~M7999		8000	
HM		HM0~HM959[※1]		960	
SM		Special purpose SM0~SM2047[※2]		2048	
S	Flow	S0~S1023		1024	
HS		HS0~HS127[※1]		128	
T	Timer	T0~T575		576	
HT		HT0~HT95[※1]		96	
ET		Precise timer ET0~ET31		32	
C	Counter	C0~C575		576	
HC		HC0~HC95[※1]		96	
HSC		High speed counter HSC0~HSC31		32	
D	Data register	D0~D7999		8000	
HD		HD0~HD999[※1]		1000	
SD		Special purpose SD0~SD2047		2048	
HSD		Special purpose HSD0~HSD499[※2]		500	
FD	FlashROM register	FD0~FD5119		5120	
SFD		Special purpose SFD0~SFD1999[※2]		2000	

FS	Special secret register	FS0~FS47	48
ID※6	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID10900～ID10999（#10 expansion module）	1000
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD10900～QD10999（#10 expansion module）	1000
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

XD2 series PLC soft components list:

	Name	Range					Points				
		16 I/O	24 I/O	32 I/O	48 I/O	60 I/O	16	24	32	48	60
X	Input points	X0~X7	X0~X15	X0~X21	X0~X33	X0~X43	8	14	18	28	36
Y	Output points	Y0~Y7	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	8	10	14	20	24
X	Input points※3	X10000～X10077（#1 expansion module） …… X11100～X11177（#10 expansion module）					640				
Y	Output points※3	Y10000～Y10077（#1 expansion module） …… Y11100～Y11177（#10 expansion module）					640				
X	Input points ※4	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）					128				

Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）	128
X	Input points※5	X30000～X30077（#1 expansion ED）	64
Y	Output points※5	Y30000～Y30077（#1 expansion ED）	64
M	Internal relay	M0~M7999	8000
HM		HM0~HM959※1	960
SM		Special purpose SM0~SM2047※2	2048
S	Flow	S0~S1023	1024
HS		HS0~HS127※1	128
T	Timer	T0~T575	576
HT		HT0~HT95※1	96
ET		Precise timer ET0~ET31	32
C	Counter	C0~C575	576
HC		HC0~HC95※1	96
HSC		High speed counter HSC0~HSC31	32
D	Data register	D0~D7999	8000
HD		HD0~HD999※1	1000
SD		Special purpose SD0~SD2047	2048
HSD		Special purpose HSD0~HSD499※2	500
FD	FlashROM register	FD0~FD5119	5120
SFD		Special purpose SFD0~SFD1999※2	2000
FS	Special secret register	FS0~FS47	48
ID※6	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID10900～ID10999（#10 expansion module）	1000
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD10900～QD10999（#10 expansion module）	1000
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

XD3 series PLC soft components list:

	Name	Range					Points				
		16 I/O	24 I/O	32 I/O	48 I/O	60 I/O	16	24	32	48	60
X	Input points	X0~X7	X0~X15	X0~X21	X0~X33	X0~X43	8	14	18	28	36
Y	Output points	Y0~Y7	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	8	10	14	20	24
X	Input points※3	X10000～X10077（#1 expansion module） …… X11100～X11177（#10 expansion module）					640				
Y	Output points※3	Y10000～Y10077（#1 expansion module） …… Y11100～Y11177（#10 expansion module）					640				
X	Input points ※4	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）					128				
Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）					128				
X	Input points※5	X30000～X30077（#1 expansion ED）					64				
Y	Output points※5	Y30000～Y30077（#1 expansion ED）					64				
M	Internal relay	M0~M7999					8000				
HM		HM0~HM959※1					960				
SM		special purpose SM0~SM2047※2					2048				
S	Flow	S0~S1023					1024				
HS		HS0~HS127※1					128				
T	Timer	T0~T575					576				
HT		HT0~HT95※1					96				
ET		precise timer ET0~ET31					32				
C	Counter	C0~C575					576				
HC		HC0~HC95※1					96				
HSC		high speed counter HSC0~HSC31					32				
D	Data register	D0~D7999					8000				
HD		HD0~HD999※1					1000				
SD		special purpose SD0~SD2047					2048				
HSD		special purpose HSD0~HSD499※2					500				
FD	FlashROM register	FD0~FD5119					5120				
SFD		special purpose SFD0~SFD1999※2					2000				
FS	Special secret register	FS0~FS47					48				
ID※6	Main body	ID0~ID99					100				
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID10900～ID10999（#10 expansion module）					1000				
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）					200				

	expansion ED	ID30000～ID30099（#1expansion ED）	100
QD[※7]	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD10900～QD10999（#10 expansion module）	1000
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

XD5 series PLC soft components list:

	Name	Range				Points			
		24 I/O	32 I/O	48 I/O	60 I/O	24	32	48	60
X	Input points	X0~X15	X0~X21	X0~X33	X0~X43	14	18	28	36
Y	Output points	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	10	14	20	24
X	Input points[※3]	X10000～X10077（#1 expansion module） …… X11700～X11777（#16 expansion module）				1024			
Y	Output points[※3]	Y10000～Y10077（#1 expansion module） …… Y11700～Y11777（#16 expansion module）				1024			
X	Input points [※4]	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）				192			
Y	Output points [※4]	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）				192			
X	Input points[※5]	X30000～X30077（#1 expansion ED）				64			
Y	Output points[※5]	Y30000～Y30077（#1 expansion ED）				64			
M	Internal relay	M0~M69999				70000			
HM		HM0~HM11999[※1]				12000			
SM		special purpose SM0~SM4999[※2]				5000			
S	Flow	S0~S7999				8000			
HS		HS0~HS999[※1]				1000			
T	Timer	T0~T4999				5000			
HT		HT0~HT1999[※1]				2000			
ET		precise timer ET0~ET39				40			
C	Counter	C0~C4999				5000			
HC		HC0~HC1999[※1]				2000			

HSC		high speed counter HSC0~HSC39	40
D	Data register	D0~D69999（firmware V3.5.3 and up）	70000
		D0~D59999（firmware V3.5.2 and down）	60000
HD		HD0~HD24999※1	25000
SD		special purpose SD0~SD4999	5000
HSD		special purpose HSD0~HSD1023※2	1024
FD	FlashROM Register	FD0~FD8191	8192
SFD		special purpose SFD0~SFD5999※2	6000
FS	Special secret register	FS0~FS47	48
ID※6	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID11500～ID11599（#16 expansion module）	1600
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD11500～QD11599（#16 expansion module）	1600
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM127	128

XDM series PLC soft components list:

	Name	Range			Points		
		24 I/O	32 I/O	60 I/O	24	32	60
X	Input points	X0~X15	X0~X21	X0~X43	14	18	36
Y	Output points	Y0~Y11	Y0~Y15	Y0~Y27	10	14	24
X	Input points[※3]	X10000～X10077（#1 expansion module） …… X11700～X11777（#16 expansion module）			1024		
Y	Output points[※3]	Y10000～Y10077（#1 expansion module） …… Y11700～Y11777（#16 expansion module）			1024		
X	Input points [※4]	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）			128		
Y	Output points [※4]	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）			128		
X	Input points[※5]	X30000～X30077（#1 expansion ED）			64		
Y	Output points[※5]	Y30000～Y30077（#1 expansion ED）			64		
M	Internal relay	M0~M69999			70000		
HM		HM0~HM11999[※1]			12000		
SM		special purpose SM0~SM4999[※2]			5000		
S	Flow	S0~S7999			8000		
HS		HS0~HS999[※1]			1000		
T	Timer	T0~T4999			5000		
HT		HT0~HT1999[※1]			2000		
ET		precise timer ET0~ET39			40		
C	Counter	C0~C4999			5000		
HC		HC0~HC1999[※1]			2000		
HSC		high speed counter HSC0~HSC39			40		
D	Data register	D0~D69999			70000		
HD		HD0~HD24999[※1]			25000		
SD		special purpose SD0~SD4999			5000		
HSD		special purpose HSD0~HSD1023[※2]			1024		
FD	FlashROM register	FD0~FD8191			8192		
SFD		special purpose SFD0~SFD5999[※2]			6000		
FS	Special secret register	FS0~FS47			48		
ID[※6]	Main body	ID0~ID99			100		
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID11500～ID11599（#16 expansion module）			1600		

	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD11500～QD11599（#16 expansion module）	1600
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM127	128

XDC series PLC soft components list:

	Name	Range				Points			
		24 I/O	32 I/O	48 I/O	60 I/O	24	32	48	60
X	Input points	X0~X15	X0~X21	X0~X33	X0~X43	14	18	28	36
Y	Output points	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	10	14	20	24
X	Input points※3	X10000～X10077（#1 expansion module） …… X11700～X11777（#16 expansion module）				1024			
Y	Output points※3	Y10000～Y10077（#1 expansion module） …… Y11700～Y11777（#16 expansion module）				1024			
X	Input points ※4	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）				128			
Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）				128			
X	Input points※5	X30000～X30077（#1 expansion ED）				64			
Y	Output points※5	Y30000～Y30077（#1 expansion ED）				64			
M	Internal relay	M0~M69999				70000			
HM		HM0~HM11999※1				12000			
SM		special purpose SM0~SM4999※2				5000			
S	Flow	S0~S7999				8000			

HS		HS0~HS999※1	1000
T	Timer	T0~T4999	5000
HT		HT0~HT1999※1	2000
ET		precise timer ET0~ET39	40
C	Counter	C0~C4999	5000
HC		HC0~HC1999※1	2000
HSC		high speed counter HSC0~HSC39	40
D	Data register	D0~D69999	70000
HD		HD0~HD24999※1	25000
SD		special purpose SD0~SD4999	5000
HSD		special purpose HSD0~HSD1023※2	1024
FD	FlashROM register	FD0~FD8191	8192
SFD		special purpose SFD0~SFD5999※2	6000
FS	Special secret register	FS0~FS47	48
ID※6	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID11500～ID11599（#16 expansion module）	1600
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD11500～QD11599（#16 expansion module）	1600
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM127	128

XD5E series PLC soft components list:

	Name	Range		Points	
		30 I/O	60 I/O	30	60
X	Input points	X0~X17	X0~X43	16	36
Y	Output points	Y0~Y15	Y0~Y27	14	24
X	Input points※3	X10000～X10077（#1 expansion module） …… X11700～X11777（#16 expansion module）		1024	
Y	Output points※3	Y10000～Y10077（#1 expansion module） …… Y11700～Y11777（#16 expansion module）		1024	
X	Input points ※4	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）		128	
Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）		128	
X	Input points※5	X30000～X30077（#1 expansion ED）		64	
Y	Output points※5	Y30000～Y30077（#1 expansion ED）		64	
M	Internal relay	M0~M69999		70000	
HM		HM0~HM11999※1		12000	
SM		special purpose SM0~SM4999※2		5000	
S	Flow	S0~S7999		8000	
HS		HS0~HS999※1		1000	
T	Timer	T0~T4999		5000	
HT		HT0~HT1999※1		2000	
ET		precise timer ET0~ET39		40	
C	Counter	C0~C4999		5000	
HC		HC0~HC1999※1		2000	
HSC		high speed counter HSC0~HSC39		40	
D	Data register	D0~D69999		70000	
HD		HD0~HD24999※1		25000	
SD		special purpose SD0~SD4999		5000	
HSD		special purpose HSD0~HSD1023※2		1024	
FD	FlashROM register	FD0~FD8191		8192	
SFD		special purpose SFD0~SFD5999※2		6000	
FS	Special secret register	FS0~FS47		48	
ID※6	Main body	ID0~ID99		100	
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID11500～ID11599（#16 expansion module）		1600	

	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD11500～QD11599（#16 expansion module）	1600
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM127	128

XDME series PLC soft components list:

	Name	Range	Points
		60 I/O	60
X	Input points	X0~X43	36
Y	Output points	Y0~Y27	24
X	Input points※3	X10000～X10077（#1 expansion module） …… X11700～X11777（#16 expansion module）	1024
Y	Output points※3	Y10000～Y10077（#1 expansion module） …… Y11700～Y11777（#16 expansion module）	1024
X	Input points ※4	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）	128
Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）	128
X	Input points※5	X30000～X30077（#1 expansion ED）	64
Y	Output points※5	Y30000～Y30077（#1 expansion ED）	64
M	Internal relay	M0~M69999	70000
HM		HM0~HM11999※1	12000
SM		special purpose SM0~SM4999※2	5000

S	Flow	S0~S7999	8000
HS		HS0~HS999※1	1000
T	Timer	T0~T4999	5000
HT		HT0~HT1999※1	2000
ET		precise timer ET0~ET39	40
C	Counter	C0~C4999	5000
HC		HC0~HC1999※1	2000
HSC		high speed counter HSC0~HSC39	40
D	Data register	D0~D69999	70000
HD		HD0~HD24999※1	25000
SD		special purpose SD0~SD4999	5000
HSD		special purpose HSD0~HSD1023※2	1024
FD	FlashROM register	FD0~FD8191	8192
SFD		special purpose SFD0~SFD5999※2	6000
FS	Special secret register	FS0~FS47	48
ID※6	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID11500～ID11599（#16 expansion module）	1600
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD11500～QD11599（#16 expansion module）	1600
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM127	128

XL1, XL3 series PLC soft components list:

	Name	Range	Points
		16 I/O	16
X	Input points	X0~X7	8
Y	Output points	Y0~Y7	8
X	Input points※3	X10000～X10077（#1 expansion module） …… X11100～X11177（#10 expansion module）	640
Y	Output points※3	Y10000～Y10077（#1 expansion module） …… Y11100～Y11177（#10 expansion module）	640
X	Input points ※4	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）	128
Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）	128
X	Input points※5	X30000～X30077（#1 expansion ED）	64
Y	Output points※5	Y30000～Y30077（#1 expansion ED）	64
M	Internal relay	M0~M7999	8000
HM		HM0~HM959※1	960
SM		special purpose SM0~SM2047※2	2048
S	Flow	S0~S1023	1024
HS		HS0~HS127※1	128
T	Timer	T0~T575	576
HT		HT0~HT95※1	96
ET		precise timer ET0~ET31	32
C	Counter	C0~C575	576
HC		HC0~HC95※1	96
HSC		high speed counter HSC0~HSC31	32
D	Data register	D0~D7999	8000
HD		HD0~HD999※1	1000
SD		special purpose SD0~SD2047	2048
HSD		special purpose HSD0~HSD499※2	500
FD	FlashROM register	FD0~FD5119	5120
SFD		special purpose SFD0~SFD1999※2	2000
FS	Special secret register	FS0~FS47	48

ID[※6]	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID10900～ID10999（#10 expansion module）	1000
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD[※7]	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD10900～QD10999（#10 expansion module）	1000
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100
SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM31	32

XL5, XL5E, XLME series PLC soft components list:

	Name	Range	Points
		32 I/O	32
X	Input points	X0~X17	16
Y	Output points	Y0~Y17	16
X	Input points[※3]	X10000～X10077（#1 expansion module） …… X11700～X11777（#16 expansion module）	1024
Y	Output points[※3]	Y10000～Y10077（#1 expansion module） …… Y11700～Y11777（#16 expansion module）	1024
X	Input points [※4]	X20000～X20077（#1 expansion BD） X20100～X20177（#2 expansion BD）	192

Y	Output points ※4	Y20000～Y20077（#1 expansion BD） Y20100～Y20177（#2 expansion BD）	192
X	Input points※5	X30000～X30077（#1 expansion ED）	64
Y	Output points※5	Y30000～Y30077（#1 expansion ED）	64
M	Internal relay	M0~M69999	70000
HM		HM0~HM11999※1	12000
SM		special purpose SM0~SM4999※2	5000
S	Flow	S0~S7999	8000
HS		HS0~HS999※1	1000
T	Timer	T0~T4999	5000
HT		HT0~HT1999※1	2000
ET		precise timer ET0~ET39	40
C	Counter	C0~C4999	5000
HC		HC0~HC1999※1	2000
HSC		high speed counter HSC0~HSC39	40
D	Data register	D0~D69999	70000
HD		HD0~HD24999※1	25000
SD		special purpose SD0~SD4999	5000
HSD		special purpose HSD0~HSD1023※2	1024
FD	FlashROM register	FD0~FD8191	8192
SFD		special purpose SFD0~SFD5999※2	6000
FS	Special secret register	FS0~FS47	48
ID※6	Main body	ID0~ID99	100
	Expansion module	ID10000～ID10099（#1 expansion module） …… ID11500～ID11599（#16 expansion module）	1600
	expansion BD	ID20000～ID20099（#1 expansion BD） ID20100～ID20199（#2 expansion BD）	200
	expansion ED	ID30000～ID30099（#1 expansion ED）	100
QD※7	Main body	QD0~QD99	100
	Expansion module	QD10000～QD10099（#1 expansion module） …… QD11500～QD11599（#16 expansion module）	1600
	expansion BD	QD20000～QD20099（#1 expansion BD） QD20100～QD20199（#2 expansion BD）	200
	expansion ED	QD30000～QD30099（#1 expansion ED）	100

SEM	Special coil of Sequence block instruction WAIT	SEM0~SEM127	128

※1: 【】Memory area is the default power outage holding area (Note: XD/XL series PLC power outage holding area can not be modified).
※2: Special use (non-power-down maintenance) refers to registers for special use occupied by the system, which can not be used for other purposes. For details, refer to the relevant sections of the List of Special Soft Components in the appendix of this manual.
※3: I/O address assignment (octal) of the extended module, which can be used as intermediate relay when the extension module is not connected. (XL1/XD1/XD2 does not support extension modules, XD3/XL3 can expand up to 10 at the same time, XD5/XDM/XDC/XD5E/XDME/XL5/XL5E/XLME can expand up to 16 at the same time)
※4: Extended BD I/O address allocation (octal), can be used as intermediate relay when not connected to BD. (24/32/30 points can be extended up to 1, 48/60 points can be extended up to 2, 16 points do not support extended BD, XL series does not support extended BD)
※5: Extended ED I/O address allocation (octal), can be used as intermediate relay when not connected to ED. (XD/XL series can extend up to one ED module)
※6: Analog input soft component address, can be used as auxiliary register when not connected to extended equipment.
※7: Analog output soft component address, can be used as auxiliary registers when not connected to extended devices.
※8: The range of soft components mentioned above is the valid range of PLC in X-NET communication mode. In MODBUS communication mode, some relays can not read and write. The specific usable range is shown in chapter 6-2-3.

2-4. Input/output relays (X, Y)

Number List

XD/XL series PLC input/output are all in octal form, each series numbers are listed below:

Series	Name	Range		Points	
		16 I/O	32 I/O	16	32
XD1	X	X0~X7	X0~X17	8	16
	Y	Y0~Y7	Y0~Y17	8	16

Series	Name	Range					Points				
		16 I/O	24 I/O	32 I/O	48 I/O	60 I/O	16	24	32	48	60
XD2 XD3 XD5	X	X0~X7	X0~X15	X0~X21	X0~X33	X0~X43	8	14	18	28	36
	Y	Y0~Y7	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	8	10	14	20	24

Series	Name	Range			Points		
		24 I/O	32 I/O	60 I/O	24	32	60
XDM	X	X0~X15	X0~X21	X0~X43	14	18	36
	Y	Y0~Y11	Y0~Y15	Y0~Y27	10	14	24

Series	Name	Range				Points			
		24 I/O	32 I/O	48 I/O	60 I/O	24	32	48	60
XDC	X	X0~X15	X0~X21	X0~X33	X0~X43	14	18	28	36
	Y	Y0~Y11	Y0~Y15	Y0~Y23	Y0~Y27	10	14	20	24

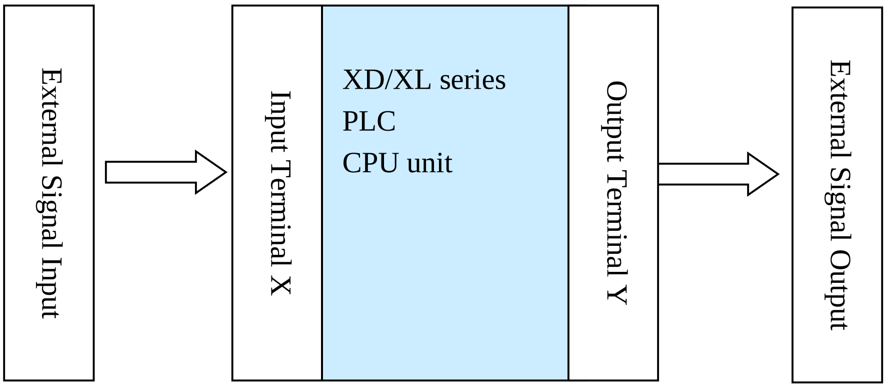
Series	Name	Range		Points	
		30 I/O	60 I/O	30	60
XD5E	X	X0~X17	X0~X43	16	36
	Y	Y0~Y15	Y0~Y27	14	24

Series	Name	Range	Points
		60 I/O	60
XDME	X	X0~X43	36
	Y	Y0~Y27	24

Series	Name	Range	Points
		16 I/O	16
XL1 XL3	X	X0~X7	8
	Y	Y0~Y7	8

Series	Name	Range	Points
		32 I/O	32
XL5 XL5E XLME	X	X0~X17	16
	Y	Y0~Y17	16

Function



Input Relay X

PLC input terminals are used to recive the external signal. the input relays are optocoupler to connect PLC and input terminals

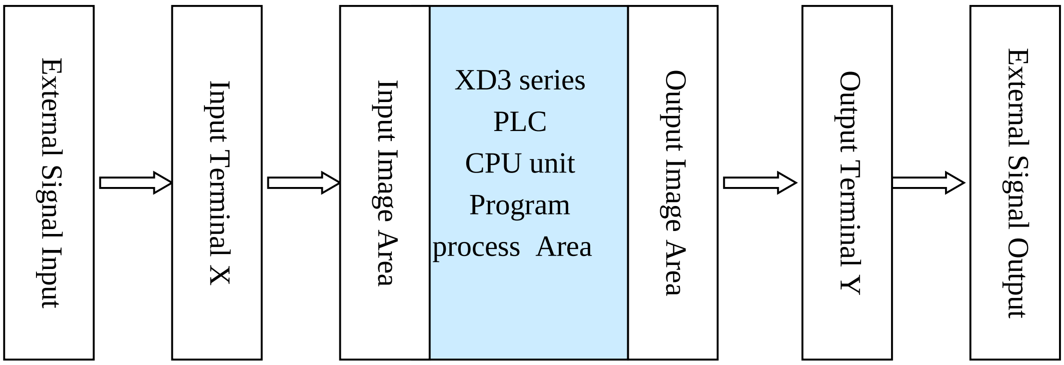
The input relays which are not connected with external devices can be seemed to fast internal relays

Output Relay Y

PLC output terminals can be used to send signals to external loads. Inside PLC, output relay's external output contactors (including relay contactors, transistor's contactors) connect with output terminals

The output relays which are not connected with external devices can be seemed to fast internal relays

Execution Order



Input processing

Before PLC executing the program, read every input terminal's ON/OFF status to the image area.

When the program is running, even the input changed, the content in the input image area will not change until the next scanning period coming.
Output processing
After running all the instructions, transfer the ON/OFF status of output Y image area to the output lock memory area. This will be the actual output of the PLC.
The output contactors will delay the action according to the output soft components reponse.

2-5. Auxiliary Relay (M, HM, SM)

Number List

The auxiliary relays in XD/XL series PLC are all in decimal form, please see the following table:

Series	Name	Range		
		Normal	Power-off holding	Special
XD1	M	M0~M7999	HM0-HM959	SM0~SM2047
XD2		M0~M7999	HM0-HM959	SM0~SM2047
XD3		M0~M7999	HM0-HM959	SM0~SM2047
XD5		M0~M69999	HM0-HM11999	SM0~SM4999
XDM		M0~M69999	HM0-HM11999	SM0~SM4999
XDC		M0~M69999	HM0-HM11999	SM0~SM4999
XD5E		M0~M69999	HM0-HM11999	SM0~SM4999
XDME		M0~M69999	HM0-HM11999	SM0~SM4999
XL1		M0~M7999	HM0-HM959	SM0~SM2047
XL3		M0~M7999	HM0-HM959	SM0~SM2047
XL5		M0~M69999	HM0-HM11999	SM0~SM4999
XL5E		M0~M69999	HM0-HM11999	SM0~SM4999
XLME		M0~M69999	HM0-HM11999	SM0~SM4999

In PLC, auxiliary relays are used frequently. This type of relay's coil is same to the output relay. They are driven by soft components in PLC;
Auxiliary relays M and HM have countless normally ON/OFF contactors. They can be used freely, but this type of contactors can't drive the external loads.

- For common use

This type of auxiliary relays can be used only as normal auxiliary relays. I.e. if power supply suddenly shut down during the running, the relays will be off.
Common usage relays can't be used for power off retentive, but the zone can be modified;

- For Power Off Retentive Use

The auxiliary relays for power off retentive usage, even the PLC is OFF, they can keep the ON/OFF status.
Power off retentive zone cannot be modified;
Power off retentive relays are usually used to memory the status before stop the power, then when power the PLC on again, the status can run again;

For Special Usage
Special relays are some relays which are defined with special meanings or functions, start from SM0.
There are two functions for special relays, first is used to drive the coil, the other type is for special running.
E.g.: SM2 is the initial pulse, activates only at the moment of start
SM34 is “all output disabled”
Special auxiliary relays can’t be used as normal relay M;

Note: The range of soft components mentioned above is the valid range of PLC in the X-NET communication mode. In the MODBUS communication mode, some relays can not read and write. The specific usable range is shown in chapter 6-2-3.

2-6. Status Relay (S, HS)

Address List

Status relays addresses of XD/XL series PLC are in form of decimal, the address are shown below:

Series	Name	Range	
		Normal	Power-off holding
XD1	S	S0~S1023	HS0~HS127
XD2		S0~S1023	HS0~HS127
XD3		S0~S1023	HS0~HS127
XD5		S0~S7999	HS0~HS999
XDM		S0~S7999	HS0~HS999
XDC		S0~S7999	HS0~HS999
XD5E		S0~S7999	HS0~HS999
XDME		S0~S7999	HS0~HS999
XL1		S0~S1023	HS0~HS127
XL3		S0~S1023	HS0~HS127
XL5		S0~S7999	HS0~HS999
XL5E		S0~S7999	HS0~HS999
XLME		S0~S7999	HS0~HS999

Function

Status relays S and HS are very import in ladder program; they are used together with instruction “STL” in the flow. The flow can make the program clear and easy to modify.

- For common use

After shut off the PLC power, S relays will be OFF

- For Power Off Retentive Use

HS relays can keep the ON/OFF status even PLC power is off

The status relays also have countless "normally ON/OFF" contactors. So users can use them freely in the program.

Note: The range of soft components mentioned above is the valid range of PLC in the X-NET communication mode. In the MODBUS communication mode, some relays can not read and write. The specific usable range is shown in chapter 6-2-3.

2-7. Timer (T, HT)

Address List

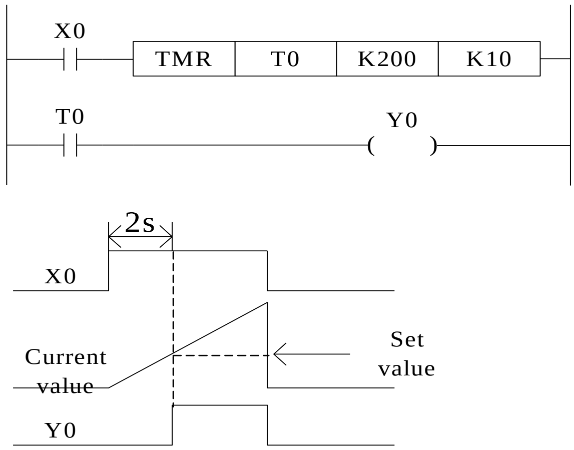
The timer addresses of XD/XL series PLC are in the form of decimal; please see the following table:

Series	Name	Range		
		Normal	Power-off holding	Precise timer
XD1	T HT ET	T0~T575	HT0~HT95	ET0~ET31
XD2		T0~T575	HT0~HT95	ET0~ET31
XD3		T0~T575	HT0~HT95	ET0~ET31
XD5		T0~T4999	HT0~HT1999	ET0~ET39
XDM		T0~T4999	HT0~HT1999	ET0~ET39
XDC		T0~T4999	HT0~HT1999	ET0~ET39
XD5E		T0~T4999	HT0~HT1999	ET0~ET39
XDME		T0~T4999	HT0~HT1999	ET0~ET39
XL1		T0~T575	HT0~HT95	ET0~ET31
XL3		T0~T575	HT0~HT95	ET0~ET31
XL5		T0~T4999	HT0~HT1999	ET0~ET39
XL5E		T0~T4999	HT0~HT1999	ET0~ET39
XLME		T0~T4999	HT0~HT1999	ET0~ET39

Function

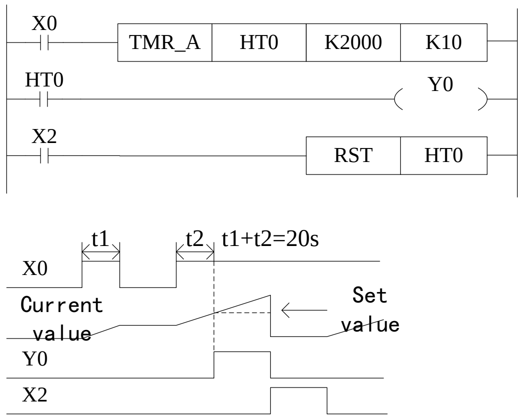
The timers accumulate the 1ms, 10ms, 100ms pulse, the output contactor activates when the accumulation reaches the set value;
TMR instruction is for common timers. The set value can be constant (K) or data register (D).

Normal type



If X0 is ON, then T0 accumulates 10ms pulse based on the current value; when the accumulation value reaches the set value K200, the timer output activates. I.e. the output activates 2s later. If X0 is OFF, the timer resets, the output resets;

Accumulation type



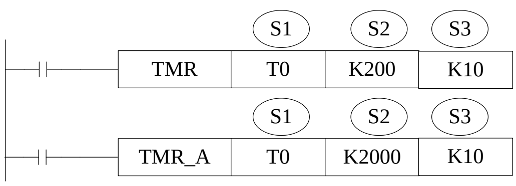
If X0 is ON, HT0 accumulates the 10ms pulse based on the current value. When the accumulation value reaches the set value K2000, the timer output activates.

If X0 is suddenly OFF during timer working, the timer value will be retentive. Then X0 is ON again, the timer will continue working.

When X2 is ON, the timer and output will be reset.

Appoint the set value

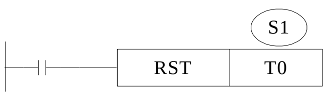
1. Instruction format



(Not accumulation)

(Accumulation)

Reset the timer and output:



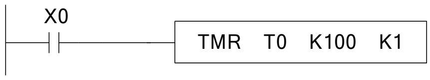
S1: timer (T0, HT10)

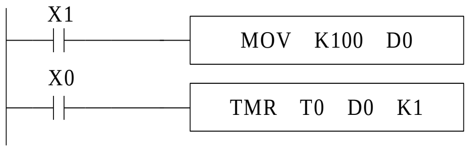
S2: set time (such as K100)

S3: time unit (K1—1ms, K10—10ms, K100—100ms)

Power-off not retentive, not accumulation

(1) Time unit is 1ms, set time is K100, the real time is 1ms *100=0.1s

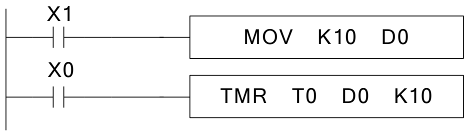




Set value is constant K

set value is register D

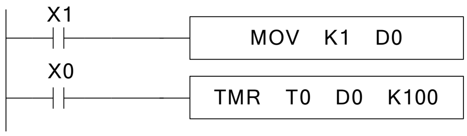
(2) Time unit is 10ms, set time is K10, the real time is 10ms*10=0.1s



Set value is constant K

set value is register D

(3) Time unit is 100ms, set time is K1, the real time is 100ms*1=0.1s

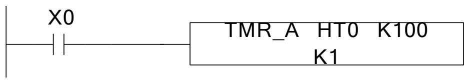


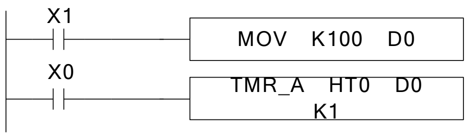
Set value is constant K

set value is register D

Power-off retentive, accumulation

(1) Time unit is 1ms, set time is K100, the real time is 1ms *100=0.1s

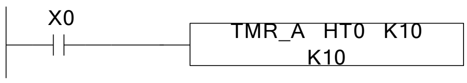


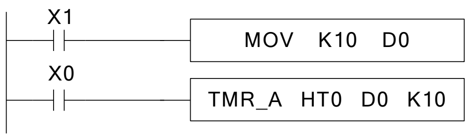


Set value is constant K

set value is register D

(2) Time unit is 10ms, set time is K10, the real time is 10ms*10=0.1s

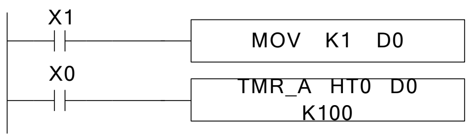




Set value is constant K

set value is register D

(3) Time unit is 100ms, set time is K1, the real time is 100ms*1=0.1s



Set value is constant K

set value is register D

Notes

(1) The timer has cumulative, non-cumulative, 1ms, 10ms and 100ms, so it can be distinguished by instructions; that is to say, the same timer can be used as either cumulative or non-cumulative, and its time base unit is also specified by instructions as 1ms, 10ms or 100ms.

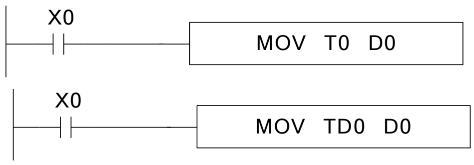
(2) The third parameter of instruction can only be based on K1, K10 and K100. Please do not write other values or registers besides these three parameters. Otherwise, although the program can be written into the programming software and downloaded to the PLC, the timing instruction will not be executed.

(3) The setting range of constant K and the actual setting value of timer are shown in the following table:

Timer	K range	Actual value
1ms timer	1～32,767	0.001～32.767s
10ms timer		0.01～327.67s
100ms timer		0.1～3276.7s

Time value

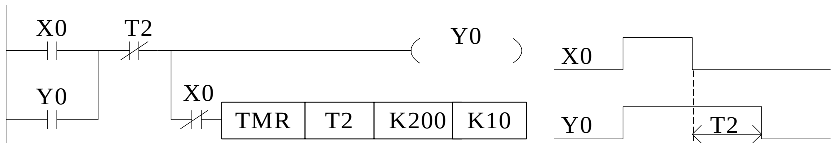
The time value is stored in register TD. The working mode of timer T0~T575 and HT0~HT95 are 16-bits linear increasing. The time range is from 0 to 32767. When the time value in TD reaches 32767, the timer will stop timing and keep the status.



The two instructions are the same.   In the first instruction, T0 is seemed to TD0.

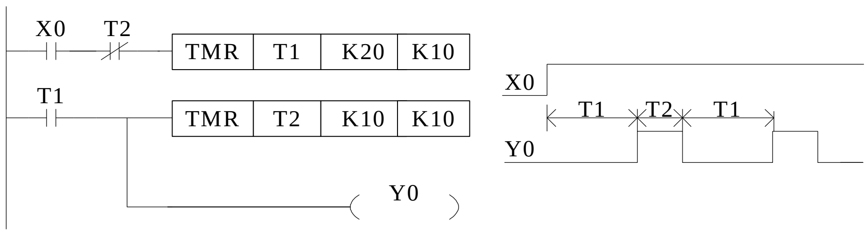
Application

Output delay



X0 is ON, output Y0. X0 changes from ON to OFF, delay 2s then cut off Y0.

Twinkle



X0 is ON, Y0 begin to twinkle. T1 is Y0-OFF time; T2 is Y0-ON time.

Note: The range of soft components mentioned above is the valid range of PLC in the X-NET communication mode. In the MODBUS communication mode, some relays can not read and write. The specific usable range is shown in chapter 6-2-3.

2-8. Counter (C, HC)

Number list

The counter addresses of XD/XL series PLC are in decimal; please see the following table for details:

Series	Name	Range		
		Normal	Power-off holding	High speed counter
XD1	C HC HSC	C0~C575	HC0~HC95	HSC0~HSC31
XD2		C0~C575	HC0~HC95	HSC0~HSC31
XD3		C0~C575	HC0~HC95	HSC0~HSC31
XD5		C0~C4999	HC0~HC1999	HSC0~HSC39
XDM		C0~C4999	HC0~HC1999	HSC0~HSC39
XDC		C0~C4999	HC0~HC1999	HSC0~HSC39
XD5E		C0~C4999	HC0~HC1999	HSC0~HSC39
XDME		C0~C4999	HC0~HC1999	HSC0~HSC39
XL1		C0~C575	HC0~HC95	HSC0~HSC31
XL3		C0~C575	HC0~HC95	HSC0~HSC31
XL5		C0~C4999	HC0~HC1999	HSC0~HSC39
XL5E		C0~C4999	HC0~HC1999	HSC0~HSC39
XLME		C0~C4999	HC0~HC1999	HSC0~HSC39

The counter range:

Counter type	Explanation
16/32 bits up/down counter	C0~C575 HC0~HC95 (32-bits counter occupies two registers, the counter address must be even number)
High speed counter	HSC0~HSC30 (HSC0,HSC2...HSC30) (each counter occupies two registers, the counter address must be even number)

1: Please refer to chapter 5 for details of high speed counter.

2: XD/XL series counters can be 16 or 32 bits count up/down mode. The mode is appointed by the instruction. Which means the same counter can be used as 16-bit or 32-bit. The increment/subtraction counting mode is also specified by the instruction mode.

Counter features

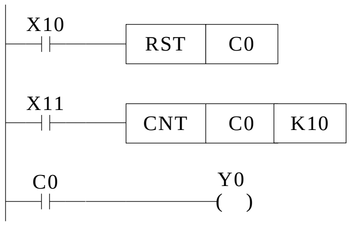
Item	16-bit counter	32-bit counter
Count direction	Count down/up	Count up/down
Set value	0~32,767	-2,147,483,648~+2,147,483,647
Set value type	Constant K or register	Constant K or a couple of registers
Count value	The value will not change when reaching the max or min value	The value will not change when reaching the max or min value
Output	Keep the state for count up	Reset for count down
Reset	Run RST instruction, the counter and output will be reset	
Present count value register	16-bit	32-bit

Function

The soft component will appoint the type of counter: common counter or power-off retentive counter.

16-bit common counter and power-off retentive counter

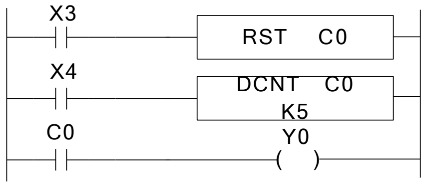
The set value range of 16-bit count-up counter is K1~K32,767 (decimal). K0 and K1 have the same function. They mean the counter output will act at the first counting.
If the PLC power supply is cut off, common counter value will be reset. The power-off retentive counter value will be kept.

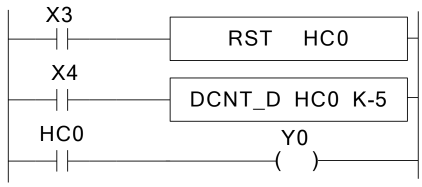


The counter C0 increases one when the X11 drives once. When C0 value reaches 10, the output acts. Then X11 drives again, C0 will continue increase one.
If X10 is ON, the C0 and output will be reset.
The counter set value can be constant K or register. For example, if D10 is 123, the set value is equal to K123.

32-bit common counter and power-off retentive counter

The set value range of 32-bit count-up/down counter is K+2,147,483,648~K-2,147,483,647 (decimal). The count direction is set through instruction.





Common count up counter power-off retentive count down counter

If X3 is ON, the counter and output will be reset.

For power-off retentive counter, the present counter value, output state will be kept after power supply is off.

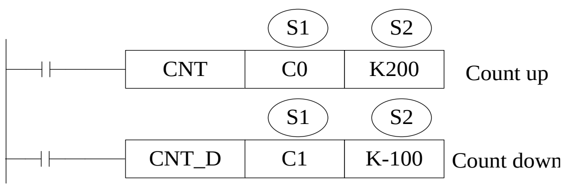
32-bit counter can be seemed to 32-bit register.

Counter set value

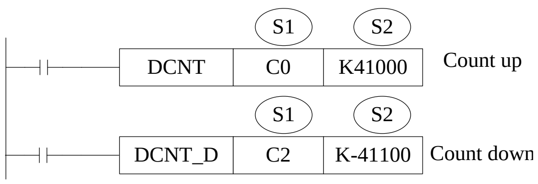
The set value contains two conditions: 16-bit and 32-bit. The counter types include common counter (C) and power-off retentive counter (HC).

Count instruction:

16-bit counter:



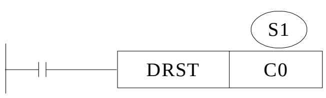
32-bit counter:



Reset instruction:

16-bit counter:

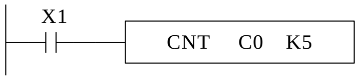
32-bit counter:



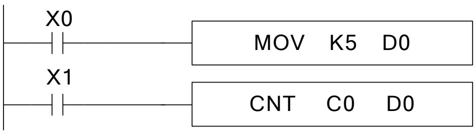
S1: counter (such as C0, HC10)
S2: counter set value (such as K100)
The counter is different from XC series. They don’t have 16-bit and 32-bit type. The type is set through instruction.

16-bit counter (common, count up)

《set value is constant K》

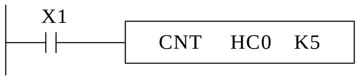


《set value is register 》

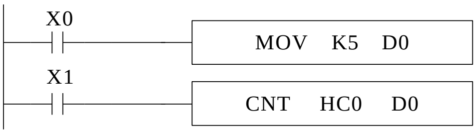


16-bit counter (power-off retentive, count up)

《set value is constant K》

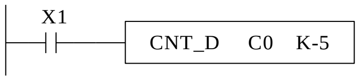


《set value is register 》

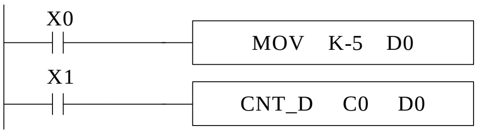


16-bit counter (common, count down)

《set value is constant K》

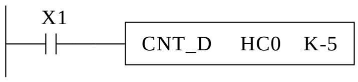


《set value is register 》

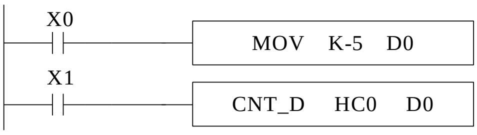


16-bit counter (power-off retentive, count down)

《set value is constant K》



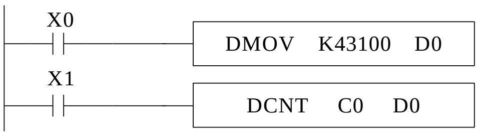
《set value is register 》



32-bit counter (common, count up)

《set value is constant K》

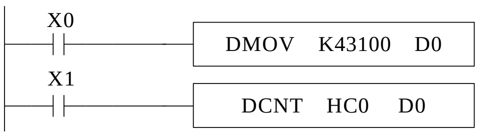
《set value is register 》



32-bit counter (power-off retentive, count up)

《set value is constant K》

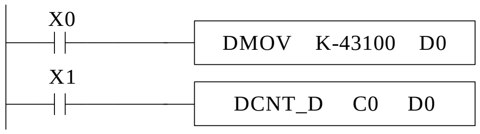
《set value is register 》



32-bit counter (common, count down)

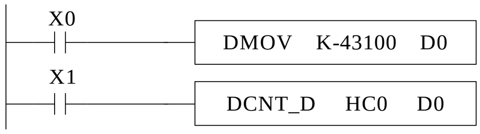
《set value is constant K》

《set value is register 》



32-bit counter (power-off retentive, count down)

《set value is constant K》 《set value is register 》



Note: The setting range and actual setting value of constant K are shown in the following table:

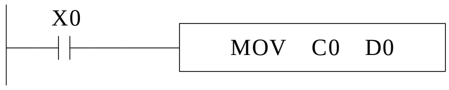
Counter	K setting range	Actual setting range
16-bit counter	1～32,767	1～32,767
32-bit counter	1～2,147,483,647	1～2,147,483,647

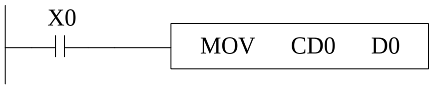
Count value

The counter counting mode is 16-bit linear incremental mode (0~K32,767). When the counter's count value CD reaches the maximum value K32,767, the counter will stop counting and the state of the counter will remain unchanged.

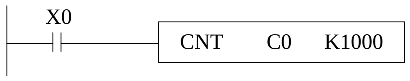
The counter counting mode is a 16-bit linear decreasing mode (-32768-0). When the counter counting value CD decreases to the minimum value K-32, 768 will stop counting and the state of the counter remains unchanged.

The counter counting mode is 32-bit linear increase/decrease mode (-2,147,483,648~+2,147,483,647). When the counter counting value increases to the maximum value K2,147,483,647, it will become K-2,147,483,648. When the counter counting value decreases to the minimum value K-2,147,483,648 will become K2,147,483,647, the ON/OFF state of the counter will also change with the change of the count value.





The above two instructions are equivalent. In the left instruction, C0 is processed as a register, while in the right instruction, CD0 is a data register corresponding to the timer C0. CD and C are one-to-one correspondences.



The highest frequency that this instruction can count is related to the selection of filter parameters and the scanning period of PLC. A high-speed counter is recommended when the input frequency exceeds 25Hz. High-number counter must use HSC0-HSC30 and corresponding hardware wiring.

High-speed counter, when SM0 is on, HSC0 counts the pulse signal of input terminal X0. High-speed counter is not affected by the response lag time of input filter and cycle scan time. Therefore, higher frequency input pulses can be processed. Refer to the details in chapter 5.

Note: The range of soft components mentioned above is the valid range of PLC in the X-NET communication mode. In the MODBUS communication mode, some relays can not read and write. The specific usable range is shown in chapter 6-2-3.

2-9. Data register (D, HD)

Address list

The data register of XD/XL series PLC is in decimal format. Please see the following table:

Series	Name	Range			
		Normal	Power-off holding	Special	Special power-off holding
XD1	D	D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XD2		D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XD3		D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XD5		D0~D59999 Or D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XDM		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XDC		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XD5E		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XDME		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XL1		D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XL3		D0~D7999	HD0~HD999	SD0~SD2047	HSD0~HSD499
XL5		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XL5E		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023
XLME		D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023

Note: For XD5 firmware version V3.5.3 and above, data register D ranges from D0 to D69999; XD5 firmware version of V3.5.2 and below, and data register D ranges from D0 to D59999.

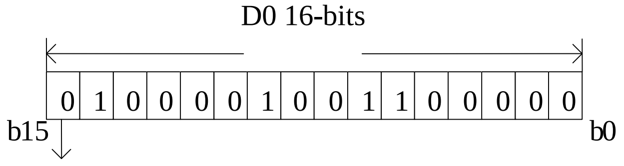
Structure

Data register is used to store data; it includes 16 bits(the higheset bit is sign bit) and 32 bits. (32 bits contains two registers, the highest bit is sign bit)

16 bits

16-bits register range is -32,768 ~ +32,767

Read and write the register data through instruction or other device such as HMI.



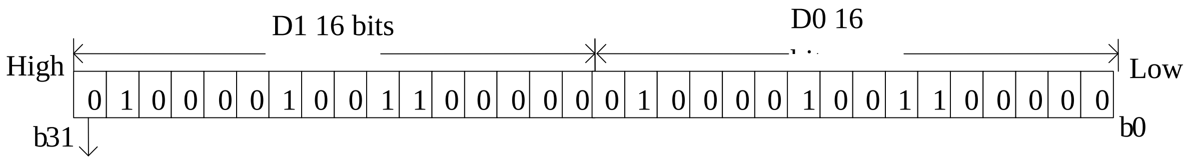
Sign bit

0: positive 1: negative

32 bits

32 bits value is consisted of two continuous registers. The range is -2147483648 ~ 2147483647. For example: (D1 D0) D1 is high 16 bits, D0 is low 16 bits.

For 32 bits register, if the low 16-bits are appointed, such as D0, then D1 will be the high 16 bits automatically. The address of low 16-bits register must be even number.



Sign bit

0: positive 1: negative

Function

- Normal type

When write a new value in the register, the former value will be covered.

When PLC changes from RUN to STOP or STOP to RUN, the value in the register will be cleared.

- Retentive type

When PLC changes from RUN to STOP or power off, the value in the register will be retained.

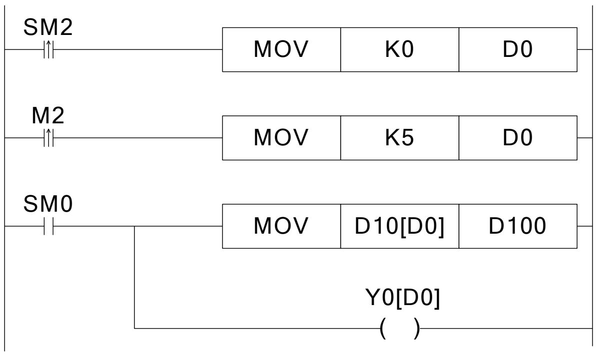
The retentive register range cannot be changed.

- Special type

Special register is used to set special data, or occupied by the system.
Some special registers are initialized when PLC is power on.
Please refer to the appendix for the special register address and function.

- Used as offset (indirect appoint)

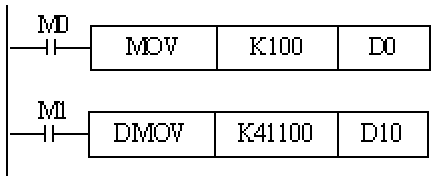
Data register can be used as offset of soft element.
Format : Dn[Dm], Xn[Dm], Yn[Dm], Mn[Dm].
Word offset: DXn[Dm] means DX[n+Dm].
The offset value only can be set as D register.



When D0=0, D100=D10, Y0 is ON;
When M2 is from OFF→ON, D0=5, D100=D15, Y5 is ON.
D10[D0]=D[10+D0], Y0[D0]=Y[0+D0].

Example

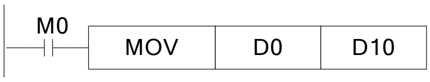
Data register D can deal with many kinds of data.
Data storage



When M0 is ON, write 100 into D0.(16 bits value)

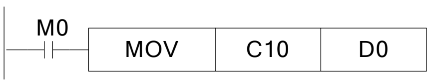
When M1 is ON, write 41100 into D11,D10 (32bits value)

Data transfer



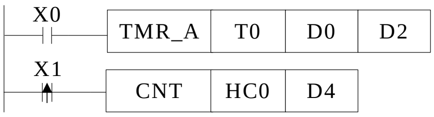
When M0 is ON, transfer the value of D10 to D0

Read the timer and counter



When M0 is ON, move the value of C10 to D0.

As the set value of timer and counter



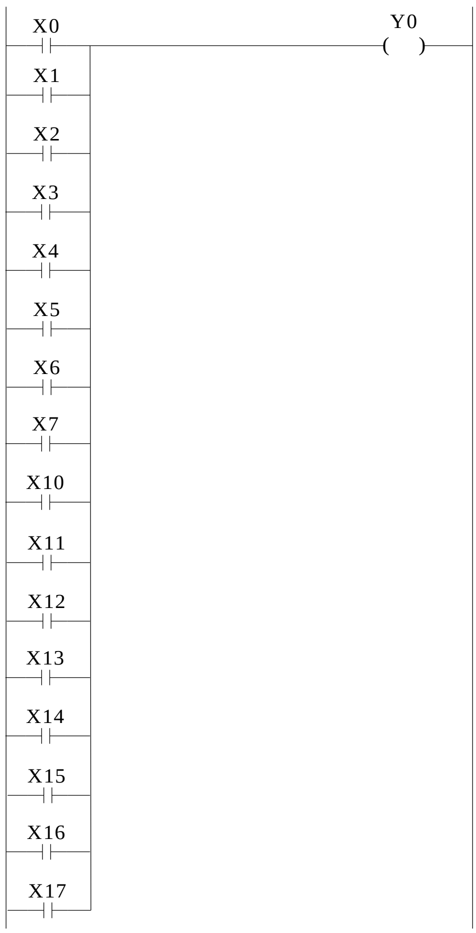
When X0 is ON, T10 starts to work, T0 will set ON when D0 value is equal to timer value, time unit is D2.

X1 is ON, HC0 starts to work, HC0 will set ON when D4 value is equal to counter value.

Note: The range of soft components m
communication mode. In the MODBUS communication mode, some relays can not read and write. The specific usable range is shown in chapter 6-2-3.

2-9-1. Word consist of bits

One of the coils from X0 to X17 is ON, Y0 will be ON.
Programming method one:

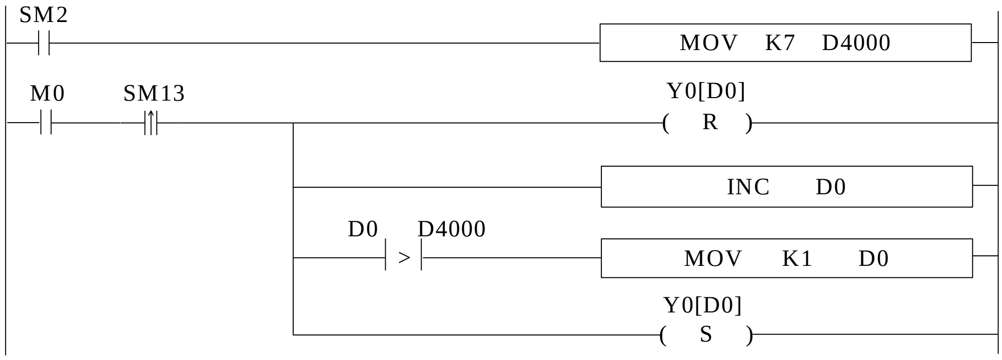


Programming method two: (application of word consists of bits)

2-9-2．Offset application

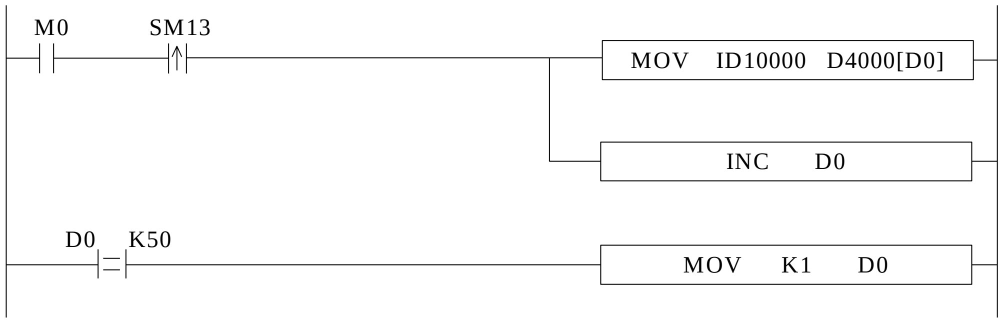
Application 1:

When M0 is ON, the output from Y1 to Y7 will be ON one by one. D0 is offset address. If there are many output points, M can replace Y.



Application 2:

When M0 is ON, read the ID10000 value every second and store in the register starting from D4000 (amounts is 50 registers). D0 is offset address.



2-10．Flash register (FD, SFD, FS)

The FLASH registers of XD/XL series PLC are all addressed in decimal system. The serial numbers are shown in the corresponding table.

Series	Name	Range		
		FLASH user data register	FLASH system data register	Password read protection FLASH register
XD1	FD SFD FS	FD0~FD5119	SFD0~SFD1999	FS0~FS47
XD2		FD0~FD5119	SFD0~SFD1999	FS0~FS47
XD3		FD0~FD5119	SFD0~SFD1999	FS0~FS47
XD5		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XDM		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XDC		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XD5E		FD0~FD8191	SFD0~SFD5999	FS0~FS47

XDME		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XL1		FD0~FD5119	SFD0~SFD1999	FS0~FS47
XL3		FD0~FD5119	SFD0~SFD1999	FS0~FS47
XL5		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XL5E		FD0~FD8191	SFD0~SFD5999	FS0~FS47
XLME		FD0~FD8191	SFD0~SFD5999	FS0~FS47

Function

- FLASH User Data Register (FD)

Used to store important data of users, can be maintained when the power is off.
This storage area can remember data even if the battery is powered down, so it can be used to store important process parameters.

- FLASH System Data Register (SFD)

Used to store system parameters and be able to maintain the data when power off.
The storage area is a system parameter block, and users can not modify it at will.

- Password Read Protection FLASH Register (FS)

A part of the FlashROM register is used to store data soft components, which are represented by the symbol FS. The values in the FS register can be written but can not be read, so they can be used to protect the intellectual property rights of users.
The value of the soft element can be set arbitrarily in the FS register, but the value of the register can not be read (always returned to 0); and it can not be compared with the register in the host computer software, only with the constant, so the actual value of the register can not be read.
This storage area can remember data even if the battery is powered down, so it can be used to store important process parameters.

Note:
(1) When using MOV instruction to transmit data to FD, SFD and FS, only the rising edge is valid, even if the driving condition is normally open/closed coil, the instruction is executed only once.
(2) Flash registers can be written about 1,000,000 times, and each write is erased for the whole Flash registers, which is time-consuming. Frequent writing will cause permanent damage to Flash registers, so it is not recommended that users write frequently. Do not use oscillating coil (e.g. SM11) as driving condition.
(3) When data is transmitted to the same Flash register several times, if the value in the source register does not change from the previous transmission, the transmission instruction will not be executed even if the driving condition is established again. For example, if the value in D0 is transmitted to FD100, the value in D0 is 300 when the transmission instruction is executed for the first time; if the driving condition is established for the second time, the transmission instruction is not executed if the value in D0 is still 300.
(4) In order to prevent the interference of burr signal when transmitting data to Flash registers, it is not recommended to use coils such as SM0 and SM2 as direct driving

conditions. It is suggested that the transmission instructions be executed after the PLC power-on for a period of time.

2-11. Constant

Data process

XD/XL series PLC has the following 5 number systems.

- DEC: DECIMAL NUMBER

The preset number of counter and timer (constant K)
The number of Auxiliary relay M, HM; timer T, HT; counter C, HC; state S, HS; register D, HD.
Set as the operand value and action of applied instruction (constant K)

- HEX: HEXADECIMAL NUMBER

Set as the operand value and action of applied instruction (constant H)

- BIN: BINARY NUMBER

Inside the PLC, all the numbers will be processed in binary. But when monitoring on the device, all the binary will be transformed into HEX or DEC.

- OCT: OCTAL NUMBER

XD/XL series PLC I/O relays are in octal. Such as [X0-7, X10-17,….X70-77].

- BCD: BINARY CODE DECIMAL

BCD uses 4 bits binary number to represent decimal number 0-9. BCD can be used in 7 segments LED and BCD output digital switch

- Other numbers (float number)

XD/XL series PLC can calculate high precision float numbers. It is calculated in binary numbers, and display in decimal numbers.

Display

PLC program should use K, H to process values. K means decimal numbers, H means hex numbers. Please note the PLC input/output relay use octal address.

- Constant K

K is used to display decimal numbers. K10 means decimal number 10. It is used to set timer and counter value, operand value of applied instruction.

- Constant H

H is used to display hex numbers. HA means decimal number 10. It is used to set operand value of applied instruction.

- Constant B

B is used to display binary numbers. B10 means decimal number 2. It is used to set operand value of applied instruction.

2-12. Programming principle

Sign P and I

P is the program sign for condition and subprogram jump.

I is the program sign for interruption (external interruption, timer interruption, high speed counter interruption, precise time interruption…).

P and I addresses are in decimal. Please refer to the following table:

Series	Sign	Address
XD, XL	P	P0~P9999

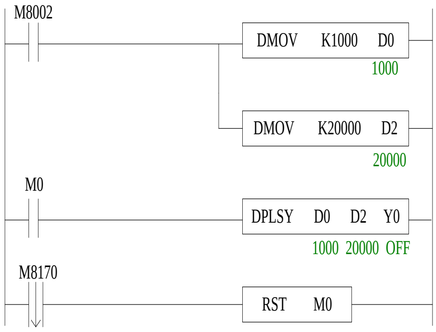
<table>
<tr><th rowspan="3">Model</th><th rowspan="3">Name</th><th colspan="4">Range</th></tr>
<tr><th colspan="3">External interruption</th><th rowspan="2">Timer interruption</th></tr>
<tr><th>Input terminal</th><th>Rising interruption</th><th>Falling interruption</th></tr>
<tr><td rowspan="6">XD1-16
XD2-16
XD3-16
XD5-16
XL1-16
XL3-16</td><td rowspan="6">I</td><td>X2</td><td>I0000</td><td>I0001</td><td rowspan="6">There are 20 timer interruptions. From I40** to I59**. “**” means the time of timer interruption, the unit is ms.</td></tr>
<tr><td>X3</td><td>I0100</td><td>I0101</td></tr>
<tr><td>X4</td><td>I0200</td><td>I0201</td></tr>
<tr><td>X5</td><td>I0300</td><td>I0301</td></tr>
<tr><td>X6</td><td>I0400</td><td>I0401</td></tr>
<tr><td>X7</td><td>I0500</td><td>I0501</td></tr>
</table>

<table>
<tr><th rowspan="3">Model</th><th rowspan="3">Name</th><th colspan="4">Range</th></tr>
<tr><th colspan="3">External interruption</th><th rowspan="2">Timer interruption</th></tr>
<tr><th>Input terminal</th><th>Rising interruption</th><th>Falling interruption</th></tr>
<tr><td rowspan="10">XD1-32
XD2-24/32/48/60
XD3-24/32/48/60
XD5-24/32/48/60
XDM
XDC
XD5E
XDME
XL5
XL5E
XLME</td><td rowspan="10">I</td><td>X2</td><td>I0000</td><td>I0001</td><td rowspan="10">There are 20 timer interruptions. From I40** to I59**. “**” means the timeof timer interruption, the unit is ms.</td></tr>
<tr><td>X3</td><td>I0100</td><td>I0101</td></tr>
<tr><td>X4</td><td>I0200</td><td>I0201</td></tr>
<tr><td>X5</td><td>I0300</td><td>I0301</td></tr>
<tr><td>X6</td><td>I0400</td><td>I0401</td></tr>
<tr><td>X7</td><td>I0500</td><td>I0501</td></tr>
<tr><td>X10</td><td>I0600</td><td>I0601</td></tr>
<tr><td>X11</td><td>I0700</td><td>I0701</td></tr>
<tr><td>X12</td><td>I0800</td><td>I0801</td></tr>
<tr><td>X13</td><td>I0900</td><td>I0901</td></tr>
</table>

Sign P

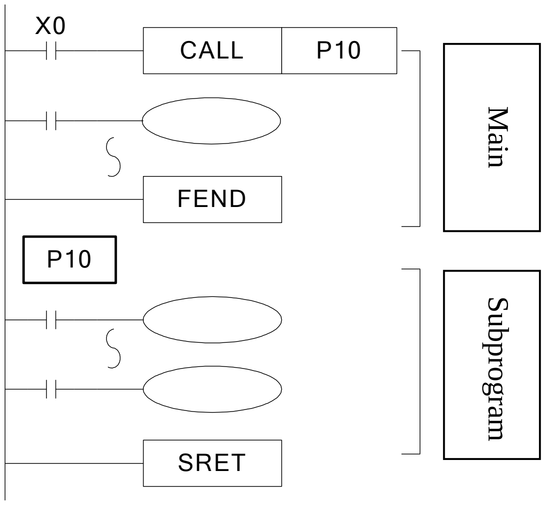
P is usually used in flow; it is used together with CJ (condition jump), CALL (call subprogram), etc.

Condition Jump CJ



If coil X0 is ON, jump to the program after P1;
If the coil X0 is not ON, do not execute jump action, but run the original program;

Call the subprogram (CALL)



If X0 is ON, jump to the subprogram
If the coil is not ON, run the original program;
After executing the subprogram, return to the main program;

The subprogram will start from Pn and finish with SRET. CALL Pn is used to call the subprogram. n is a integer in the range of 0 to 9999.

Sign I

Tag I is usually used in interruption, including external interruption, time interruption etc. It often works together with IRET (interruption return), EI (enable interruption), DI (disable interruption);

- External interruption

Accept the input signal from the special input terminals, not affected by the scan cycle. Activate the input signal, execute the interruption subroutine.
With external interruption, PLC can dispose the signal shorter than scan cycle; So it can be used as essential priority disposal in sequence control, or used in short time pulse control.

- Time interruption

Execute the interruption subroutine at each specified interruption loop time. Use this interruption in the control which is different from PLC's operation cycle;

- Action sequence of input/output relays and response delay

Input

Before PLC executing the program, read all the input terminal's ON/OFF status to the image area. In the process of executing the program, even the input changed, the content in the input image area will not change. However, in the next scan cycle, the changes will be read.

Output

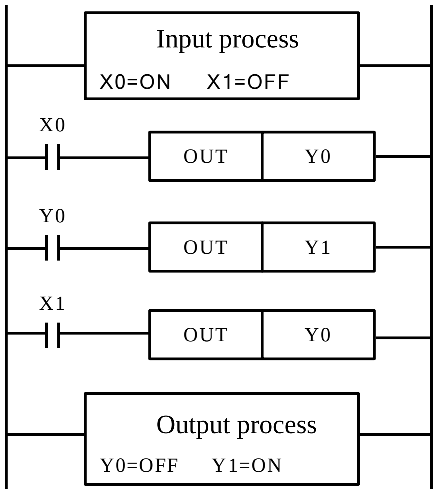
Once all the instructions end, transfers the ON/OFF status of output Y image area to the output lock memory area. This will be the actual output of the PLC. The output contactors will act according to the device's response delay time.

When use batch input/output mode, the drive time and operation cycle of input filter and output device will also show response delay.

- Not accept narrow input pulse signal

PLC's input ON/OFF time should be longer than its loop time. If consider input filter's response delay 10ms, loop time is 10ms， then ON/OFF time needs 20 ms separately. So, up to 1, 000/(20+20)=25Hz input pulse can't be processed. But, this condition could be improved when use PLC's special function and applied instructions (such as high speed count, input interruption, input filter adjustment).

- Dual output（Dual coils）action



As shown in the left map, please consider the case of using the same coil Y0 at many positions:

E.g. X0=ON, X1=OFF

The first Y0: X0 is ON, its image area is ON, output Y1 is also ON.

The second Y0: as input X1 is OFF, the image area is OFF.

So, the actual output is: Y0=OFF, Y1= ON.

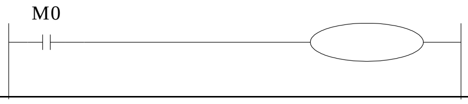
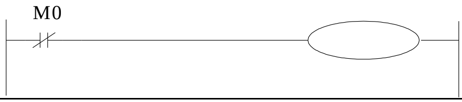
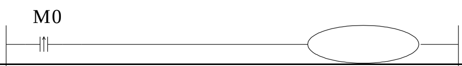
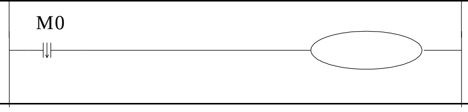
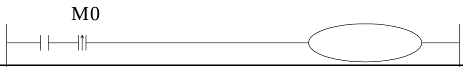
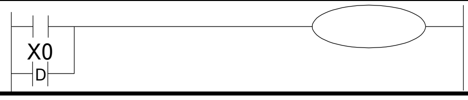
When executing dual output (use dual coil), the after one is act in priority.

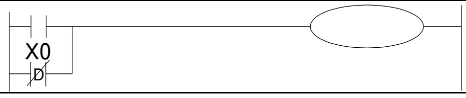
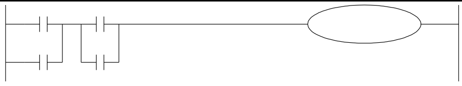
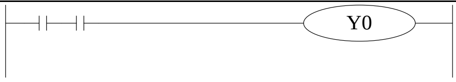
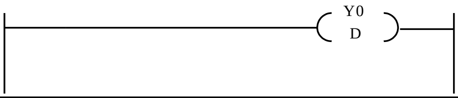
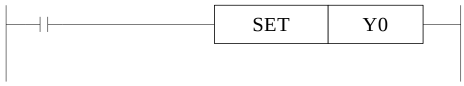
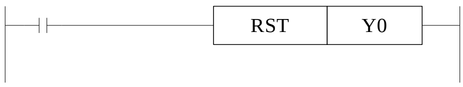
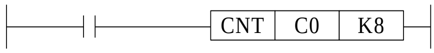
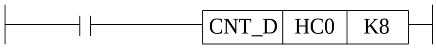
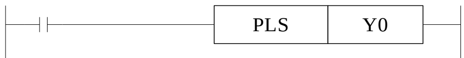
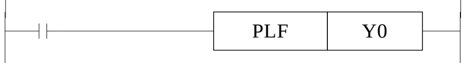
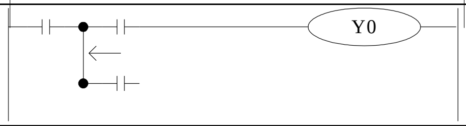
3 Basic Program Instructions

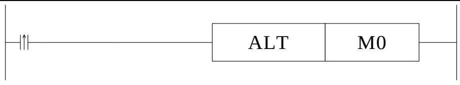
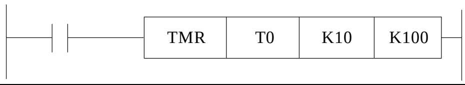
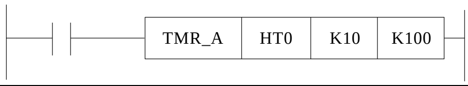
This chapter introduces the basic instructions and their functions.

3-1． Basic Instructions List

XD, XL series support all the basic instructions:

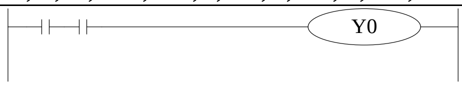
Mnemonic	Function	Format and Device	**Chapter**
LD	Initial logical operation contact type NO (normally open)		3-2
LDD	Read the status from the contact directly		3-6
LDI	Initial logical operation contact type NC (normally closed)		3-2
LDDI	Read the normally closed contact directly		3-6
LDP	Initial logical operation-Rising edge pulse		3-5
LDF	Initial logical operation-Falling /trailing edge pulse		3-5
AND	Serial connection of NO (normally open) contacts		3-3
ANDD	Read the status from the contact directly		3-6
ANI	Serial connection of NC (normally closed) contacts		3-3
ANDDI	Read the normally closed contact directly		3-6
ANDP	Serial connection of rising edge pulse		3-5
ANDF	Serial connection of falling/trailing edge pulse		3-5
OR	Parallel connection of NO (normally open) contacts		3-4
ORD	Read the status from the contact directly		3-6

ORI	Parallel connection of NC (normally closed) contacts		3-4
ORDI	Read the normally closed contact directly		3-6
ORP	Parallel connection of rising edge pulse		3-5
ORF	Parallel connection of falling/trailing edge pulse		3-5
ANB	Serial connection of multiply parallel circuits		3-8
ORB	Parallel connection of multiply parallel circuits		3-7
OUT	Final logic operation type coil drive		3-2
OUTD	Output to the contact directly		3-6
SET	Set a bit device permanently ON		3-12
RST	Reset a bit device permanently OFF		3-12
CNT	16-bit non-power-off retentive incremental count		3-13
CNT_D	16-bit power-off retentive decremented count		3-13
DCNT	32-bit non-power-off retentive incremental count		3-13
DCNT_D	32-bit power-off retentive decremented count		3-13
PLS	Turn on a scan cycle when rising edge		3-11
PLF	Turn on a scan cycle when falling edge		3-11
MCS	Connect the public serial contacts		3-9
MCR	Clear the public serial contacts		3-9

ALT	The status of the assigned device is inverted on every operation of the instruction		3-10
TMR	Non-power-off holding timer		3-14
TMR_A	Power-off holding timer		3-14
END	Force the current program scan to end		3-15
GROUP	Group		3-15
GROUPE	Group End		3-16

3-2. [LD] , [LDI] , [OUT]

Mnemonic and Function

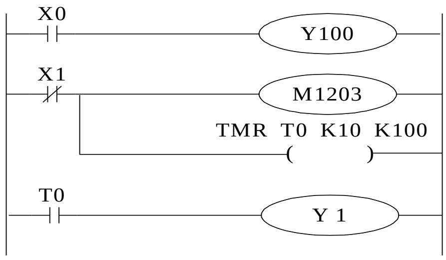
Mnemonic	Function	Format and Operands
LD (positive)	Initial logic operation contact type NO (Normally Open)	 Operands: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
LDI (negative)	Initial logic operation contact type NC (Normally Closed)	 Devices: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
OUT (OUT)	Final logic operation type drive coil	 Operands: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statement

- Connect the LD and LDI instructions directly to the left bus bar. It can work with ANB and be used at the branch start.
- OUT instruction can drive the output relays, auxiliary relays, status, timers, and counters.

But this instruction can’t be used for the input relays

Program




```
LD     X0
OUT    Y100
LDI    X1
OUT    M1203
TMR    T0       K10  K100
LD     T0
OUT    Y1
```

3-3. [AND] , [ANI]

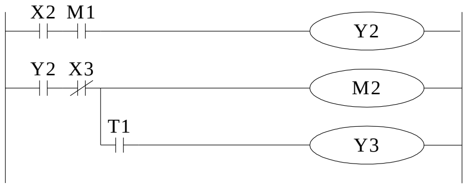
Mnemonic and Function

Mnemonic	Function	Format and Operands
AND (and)	Normal open contactor in series	 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
ANI (and reverse)	Normal close contactor in series	 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statements

- Use AND and ANI to connect the contactors in series. There is no limit for contactors in series. They can be used for many times.
- Use OUT instruction through other coil is called “follow-on” output (For an example see the program below: OUT M2 and OUT Y3). Follow-on output can repeat as long as the output order is correct. There’s no limit for the serial connected contactors and follow-on output times.

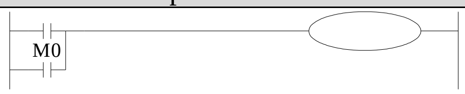
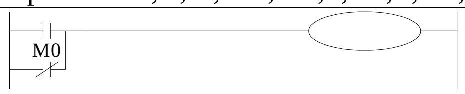
Program




```
LD    X2
AND   M1
OUT   Y2
LD    Y2
ANI   X3
OUT   M2
AND   T1
OUT   Y3
```

3-4. [OR] , [ORI]

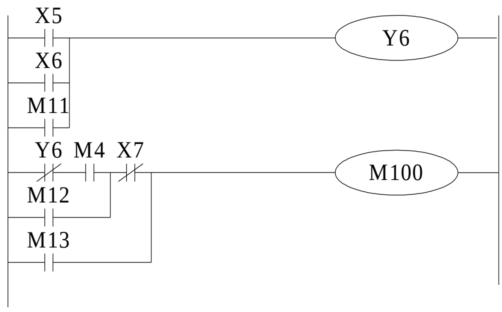
Mnemonic and Function

Mnemonic	Function	Format and Operands
OR (OR)	Parallel connection of NO (Normally Open) contactors	 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
ORI (OR reverse)	Parallel connection of NC (Normally Closed) contactors	 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statements

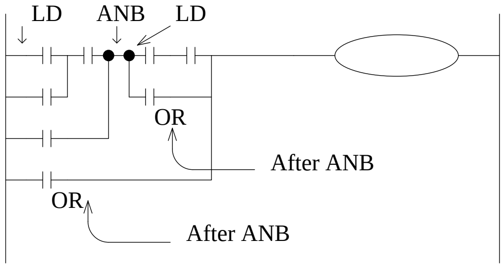
- Use the OR and ORI instructions for parallel connection of contactors. To connect a block that contains more than one contactor connected in series to another circuit block in parallel, use ORB instruction, which will be described later;
- OR and ORI start from the instruction step, parallel connect with the LD and LDI instruction step introduced before. There is no limit for the parallel connect times.

Program



LD	X5
OR	X6
OR	M11
OUT	Y6
LDI	Y6
AND	M4
OR	M12
ANI	X7
OR	M13
OUT	M100

Relationship with ANB



The parallel connection with OR, ORI instructions should connect with LD, LDI instructions in principle. But behind the ANB instruction, it’s still ok to add a LD or LDI instruction.

3-5. [LDP] , [LDF] , [ANDP] , [ANDF] , [ORP] , [ORF]

Mnemonic and Function

Mnemonic	Function	Format and Operands
LDP (LoaD Pulse)	Initial logical operation-Rising edge pulse	 X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
LDF (LoaD Falling pulse)	Initial logical operation Falling/trailing edge pulse	 X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
ANDP (AND Pulse)	Serial connection of Rising edge pulse	 X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
ANDF	Serial connection of Falling/trailing edge pulse	

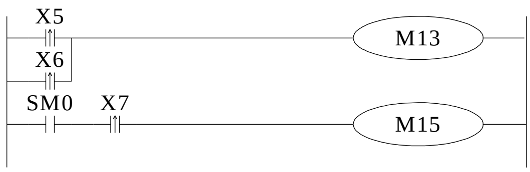
(AND Falling pulse)		X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
ORP (OR Pulse)	Parallel connection of Rising edge pulse	 X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
ORF (OR Falling pulse)	Parallel connection of Falling/trailing edge pulse	 X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statements

LDP, ANDP, ORP will be ON for one scanning period when the signal rising pulse is coming (OFF→ON)

LDF, ANDF, ORF will be ON for one scanning period when the signal falling pulse is coming (ON→OFF)

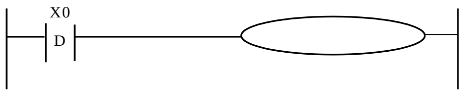
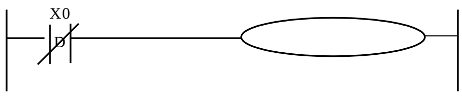
Program

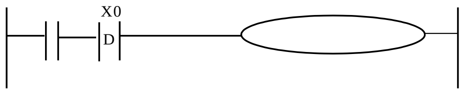
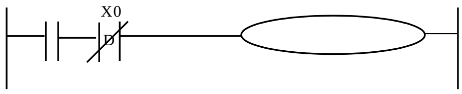
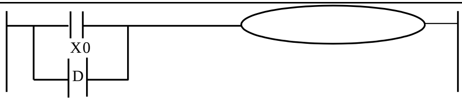
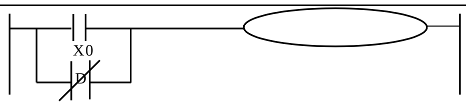
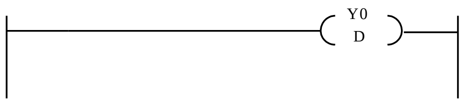



```
LDP     X5
ORP     X6
OUT     M13
LD      M8000
ANDP    X7
OUT     M15
```

3-6. [LDD] , [LDDI] , [ANDD] , [ANDDI] , [ORD] , [ORDI]， [OUTD]

Mnemonic and Function

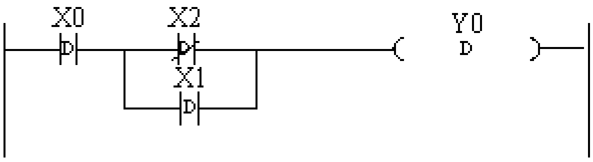
Mnemonic	Function	Format and Operands
LDD	Read the status from the contact directly	 Devices: X
LDDI	Read the normally closed contact directly	 Devices: X

ANDD	Read the status from the contact directly	 Devices: X
ANDDI	Read the normally closed contact directly	 Devices: X
ORD	Read the status from the contact directly	 Devices: X
ORDI	Read the normally closed contact directly	 Devices: X
OUTD	Output to the contact directly	 Devices: Y

Statement

The function of LDD, ANDD, ORD instructions are similar to LD, AND, OR; LDDI, ANDDI, ORDI instructions are similar to LDI, ANDI, ORI; but if the operand is X, the LDD, ANDD, ORD commands read the signal from the terminals directly.
OUTD and OUT are output instructions. OUTD will output immediately when the condition is satisfied, needn't wait for the next scan cycle.

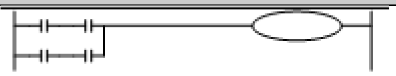
Program



LDD	X0
LDDI	X2
ORD	X2
ANB	
OUTD	Y0

3-7. [ORB]

Mnemonic and Function

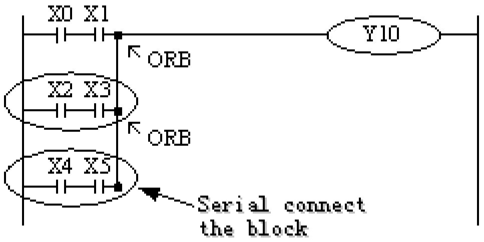
Mnemonic	Function	Format and Devices
ORB (OR Block)	Parallel connect the serial circuits	

		Devices: none

Statements

Two or more contactors is called "serial block". If parallel connect the serial block, use LD, LDI at the branch start point, use ORB at the branch end point;
As the ANB instruction, an ORB instruction is an independent instruction which is not associated with any soft component.
There are no limits for parallel circuits' quantity when using ORB for every circuit.

Program



Recommended good programming method:

LD	X0
AND	X1
LD	X2
AND	X3
ORB	
LD	X4
AND	X5
ORB	
OUT	Y10

Non-preferred programming method:

LD	X0
AND	X1
LD	X2
AND	X3
LD	X4
AND	X5
ORB	
ORB	
OUT	Y10

3-8. [ANB]

Mnemonic and Function

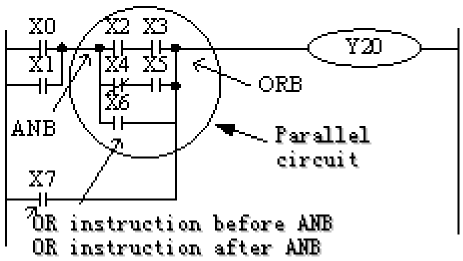
Mnemonic	Function	Format and Devices

ANB (And Block)	Serial connection of parallel circuits	 Devices: none

Statements

Use ANB to serial connects two parallel circuits. Use LD, LDI at the brach start point; use ANB at the branch end point.
There are no limits for ANB instruction using times.

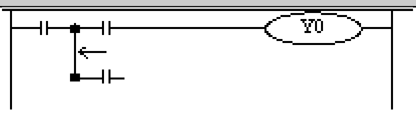
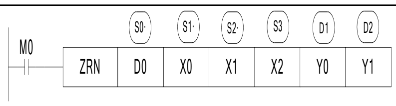
Program



LD	X0
OR	X1
LD	X2
AND	X3
LDI	X4
AND	X5
ORB	
OR	X6
ANB	
OR	X7
OUT	Y20

3-9. [MCS] , [MCR]

Mnemonic and Function

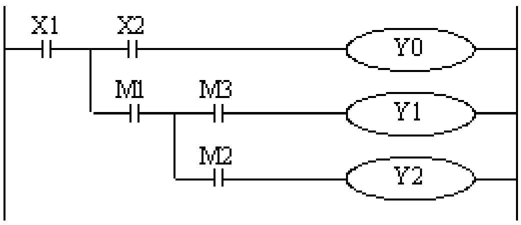
Mnemonic	Function	Format and Devices
MCS (Master control)	The start of new bus line	 Devices：None
MCR (Master control Reset)	Reset the bus line	 Devices：None

Statements

- After the execution of an MCS instruction, the bus line (LD, LDI) moves to a point after the MCS instruction. An MCR instruction resets this to the original bus line.
- MCS, MCR instructions should use in pair.
- The bus line can be nesting. Use MCS, MCR instructions between MCS, MCR instructions. The nesting level increase with the using of MCS instruction. The max nesting level is ten. When executing MCR instruction, go back to the last level of bus line.
- When use flow program, bus line management could only be used in the same flow. When the flow ends, it must go back to the main bus line.

Note: The MCS and MCR instructions can not be written directly in the ladder diagram of XD/XL series PLC programming software. They can be constructed by horizontal and vertical lines.

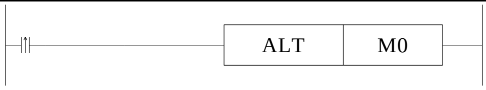
Program



LD	X1
MCS	
LD	X2
OUT	Y0
LD	M1
MCS	
LD	M3
OUT	Y1
LD	M2
OUT	Y2
MCR	
MCR	

3-10. [ALT]

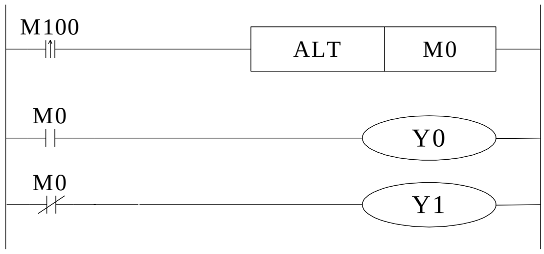
Mnemonic and Function

Mnemonic	Function	Format and Devices
ALT (Alternate)	Alternate the coil	 Coil: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statements

The status of the coil is reversed after using ALT (ON changes to OFF, OFF changes to ON).

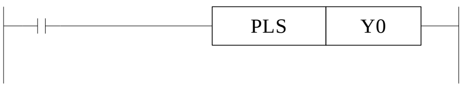
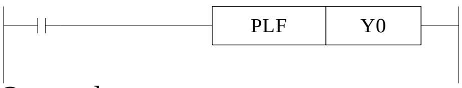
Program




```
LDP    M100
ALT    M0
LD     M0
OUT    Y0
LDI    M0
OUT    Y1
```

3-11. [PLS] , [PLF]

Mnemonic and Function

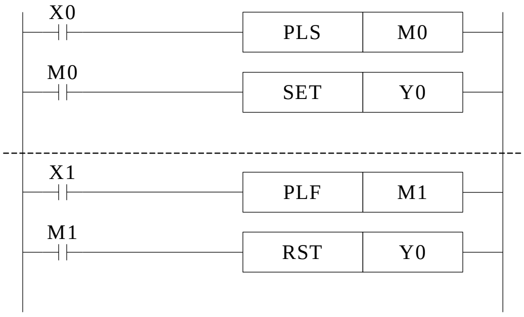
Mnemonic	Function	Format and Devices
PLS (Rising Pulse)	Turn on a scan cycle when Rising edge	 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
PLF (Falling Pulse)	Turn on a scan cycle when Falling edge	 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statements

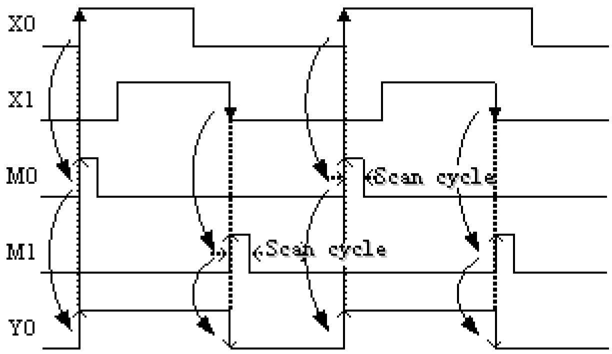
For using PLS instruction: soft component Y and M will act during one scanning period after the drive is ON.

For using PLF instruction: soft component Y and M will act during one scanning period after the drive is OFF.

Program




```
LD      X0
PLS     M0
LD      M0
SET     Y0
----------------------
LD      X1
PLF     M1
LD      M1
RST     Y0
```



3-12. [SET], [RST]

Mnemonic and Function

Mnemonic	Function	Format and Devices
SET (Set)	Set a bit device permanently ON	SET Y0 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m
RST (Reset)	Reset a bit device permanently OFF	RST Y0 Operand: X,Y,M,HM,SM,S,HS,T,HT,C,HC,Dn.m

Statements

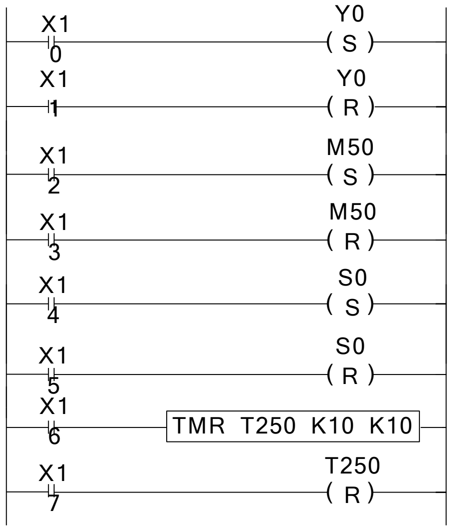
In the following program, Y0 will keep ON even X10 turns OFF after turning ON. Y0 will not ON even X11 turns OFF after turning ON. This is the same to S and M.

SET and RST can be used for many times for the same soft component. Any order is allowed, but the last one is effective.

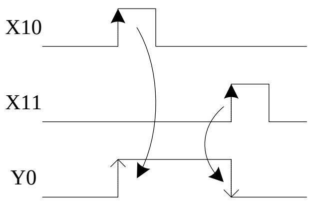
RST can be used to reset the counter, timer and contactor.

When using SET or RST, it cannot use the same soft component with OUT.

Program

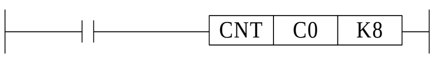
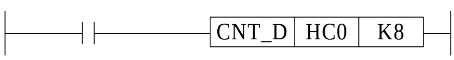


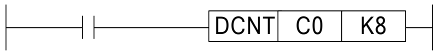
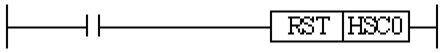

```
LD      X10
SET     Y0
LD      X11
RST     Y0
LD      X12
SET     M50
LD      X13
RST     M50
LD      X14
SET     S0
LD      X15
RST     S0
LD      X16
TMR     T250     K10     K10
LD      X17
RST     T250
```



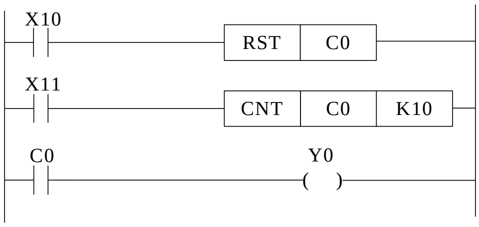
3-13. 【CNT】【CNT_D】【DCNT】【DCNT_D】【RST】for the counters

Mnemonic and Function

Mnemonic	Function	Format and devices
CNT Output	16 bits non power-off retentive increase count, the drive of count coil	 Operand: K, D
CNT_D Output	16 bits power-off retentive decrease count, the drive of count coil	 Operand: K, D

DCNT Output	32 bits non power-off retentive increase count, the drive of count coil	 Operand: K, D
DCNT_D Output	32 bits power-off retentive decrease count, the drive of count coil	 Operand: K, D
RST Reset	Reset the output coil, clear the current count value	 Operand: C, HC, HSC

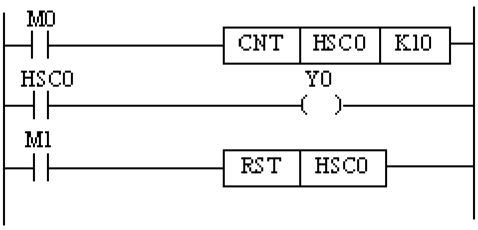
Internal counter programming



C0 increase counts the X11 OFF to ON times. When C0 reaches K10, C0 will become OFF to ON. When X11 becomes OFF to ON, the C0 current value will keep increasing, and the C0 coil will still be ON. When X10 is ON, reset the C0 coil.

Power-off retentive counter will keep the current value and counter coil status when the power is off.

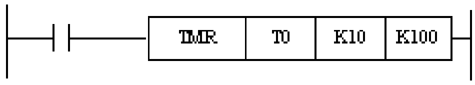
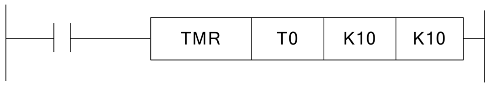
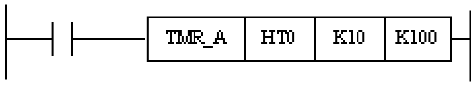
High speed counter programming



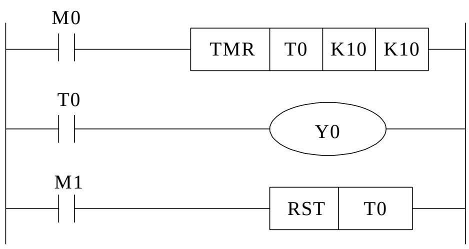
Increase count the OFF to ON times of M0.
When the count value reaches set value (value of K or D), the count coil will be ON.
When M1 is ON, the count coil of HSC0 reset, the current value becomes 0.

3-14. [TMR], [TMR-A] for timers

Mnemonic and Function

Mnemonic	Function	Format and devices
TMR output	Non power-off retentive 100ms timer, the drive of coil	 operand: K, D
TMR output	Non power-off retentive 10ms timer, the drive of coil	 operand: K, D
TMR output	Non power-off retentive 1ms timer, the drive of coil	 operand: K, D
TMR_A output	Power-off retentive 100ms timer, the drive of coil	 operand: K, D
TMR_A output	Power-off retentive 10ms timer, the drive of coil	 operand: K, D
TMR_A output	Power-off retentive 1ms timer, the drive of coil	 operand: C, HC, HSC

Internal timer programming

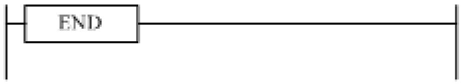


When M0 is ON, T0 starts to timing. When T0 reaches K10, T0 coil is ON. Then T0 continues timing. When M1 is ON, reset the T0.

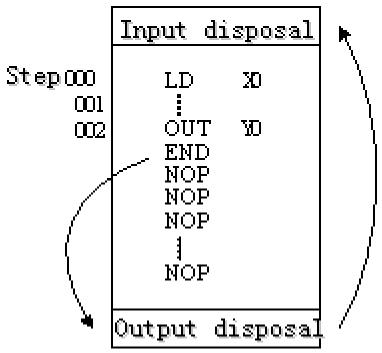
Power-off retentive timer will keep the current value and counter coil status when the power is off.

3-15. [END]

Mnemonic and Function

Mnemonic	Function	Format and Devices：None
END (END)	Force the current program scan to end	 Devices: None

Statements



PLC repeatedly carries on input disposal, program executing and output disposal. If write END instruction at the end of the program, then the instructions behind END instruction won’t be executed. If there’s no END instruction in the program, the PLC executes the end step and then repeats executing the program from step 0.

When debug, insert END in each program segment to check out each program’s action. Then, after confirm the correction of preceding block’s action, delete END instruction. Besides, the first execution of RUN begins with END instruction.

When executing END instruction, refresh monitor timer. (Check if scan cycle is a long timer.)

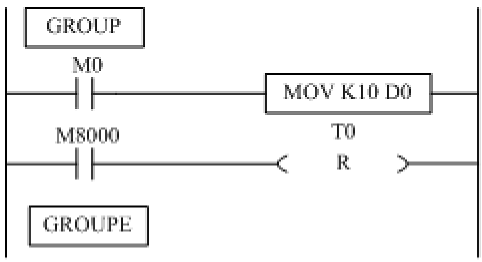
3-16. [GROUP] , [GROUPE]

Mnemonic and Function

Mnemonic	Function	Format and Device
GROUP	GROUP	GROUP Devices: None
GROUPE	GROUP END	GROUPE Devices: None

Statements

GROUP and GROUPE should used in pairs.
GROUP and GROUPE don't have practical meaning; they are used to optimize the program structure. So, add or delete these instructions doesn't affect the program's running;
The using method of GROUP and GROUPE is similar with flow instructions; enter GROUP instruction at the beginning of group part; enter GROUPE instruction at the end of group part.



Generally, GROUP and GROUPE instruction can be programmed according to the group's function. Meantime, the programmed instructions can be FOLDED or UNFOLDED. To a redundant project, these two instructions are quite useful.

3-17. Programming notes

Contactor structure and steps
Even in the sequencial control circuit with the same function, it’s also available to simplify the program and shorten the program steps according to the contactors’ structure. General programming principle is: (a) write the circuit with many serial contacts on the top; (b) write the circuit with many parallel contactors in the left.

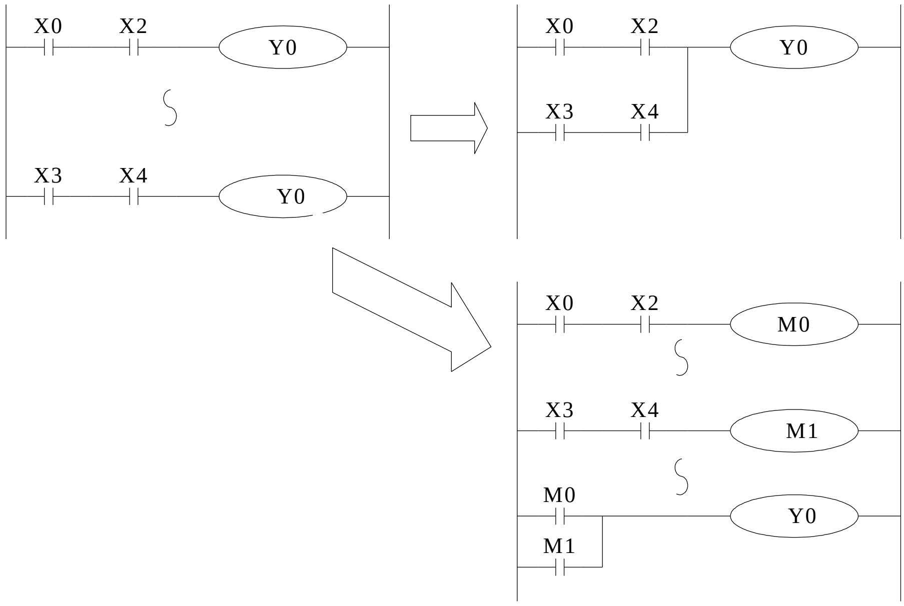
Program’s executing sequence
Handle the sequencial control program by【From top to bottom】and【From left to right】

Sequencial control instructions also encode following this procedure.

Dual output dual coil's activation and the solution

If carry on coil's dual output (dual coil) in the sequencial control program, then the last action is prior.

Dual output (dual coil) doesn't go against the input rule. But as the preceding action is very complicate, please modify the program as in the following example.



There are other methods. E.g. jump instructions or flow instructions.

4 Applied Instructions

In this chapter, we describe applied instruction’s function of XD, XL series PLC.

4-1． Applied Instructions List

Mnemonic	Function	Ladder chart	Chapter
Program Flow			
CJ	Condition jump	CJ Pn	4-3-1
CALL	Call subroutine	CALL Pn	4-3-2
SRET	Subroutine return	SRET	4-3-2
STL	Flow start	STL Sn	4-3-3
STLE	Flow end	STLE	4-3-3
SET	Open the assigned flow, close the current flow	SET Sn	4-3-3
ST	Open the assigned flow, not close the current flow	ST Sn	4-3-3
FOR	Start a FOR-NEXT loop	FOR S	4-3-4
NEXT	End of a FOR-NEXT loop	NEXT	4-3-4
FEND	Main program END	FEND	4-3-5
END	Program END	END	4-3-5
Data Compare			
LD＝	LD activates if (S1) = (S2)	LD= S1 S2	4-4-1
LD＞	LD activates if (S1) > (S2)	LD＞ S1 S2	4-4-1
LD＜	LD activates if (S1) =< (S2)	LD＜ S1 S2	4-4-1
LD＜＞	LD activates if (S1) ≠(S2)	LD＜＞ S1 S2	4-4-1
LD＜＝	LD activates if (S1) ≤ (S2)	LD＜= S1 S2	4-4-1
LD＞＝	LD activates if (S1) ≥ (S2)	LD＞= S1 S2	4-4-1
AND＝	AND activates if (S1)＝(S2)	AND＝ S1 S2	4-4-2

AND＞	AND activates if (S1)＞(S2)	AND> | S1 | S2	4-4-2
AND＜	AND activates if (S1)＜(S2)	AND< | S1 | S2	4-4-2
AND＜＞	AND activates if (S1) ≠ (S2)	AND<> | S1 | S2	4-4-2
AND＜＝	AND activates if (S1) ≤ (S2)	AND<= | S1 | S2	4-4-2
AND＞＝	AND activates if (S1) ≥ (S2)	AND＝ | S1 | S2	4-4-2
OR＝	OR activates if (S1)＝(S2)	OR= | S1 | S2	4-4-3
OR＞	OR activates if (S1)＞(S2)	OR＞ | S1 | S2	4-4-3
OR＜	OR activates if (S1)＜(S2)	OR＜ | S1 | S2	4-4-3
OR＜＞	OR activates if (S1) ≠ (S2)	OR＜＞ | S1 | S2	4-4-3
OR＜＝	OR activates if (S1) ≤ (S2)	OR＜= | S1 | S2	4-4-3
OR＞＝	OR activates if (S1) ≥ (S2)	OR＞= | S1 | S2	4-4-3
Data Move			
CMP	Compare the data	CMP | S1 | S | D	4-5-1
ZCP	Compare the data in certain area	ZCP | S1 | S2 | S | D	4-5-2
MOV	Move	MOV | S | D	4-5-3
BMOV	Block move	BMOV | S | D | n	4-5-4
PMOV	Transfer the Data block	PMOV | S | D | n	4-5-5
FMOV	Multi-points repeat move	FMOV | S | D | n	4-5-6
EMOV	Float number move	EMOV | S | D	4-5-7
FWRT	Flash ROM written	FWRT | S | D	4-5-8
MSET	Zone set	MSET | S1 | S2	4-5-9
ZRST	Zone reset	ZRST | S1 | S2	4-5-10
SWAP	Swap the high and low byte	SWAP | S	4-5-11
XCH	Exchange two values	XCH | D1 | D2	4-5-12
Data Operation			
ADD	Addition	ADD | S1 | S2 | D	4-6-1
SUB	Subtraction	SUB | S1 | S2 | D	4-6-2
MUL	Multiplication	MUL | S1 | S2 | D	4-6-3

DIV	Division	DIV S1 S2 D	4-6-4
INC	Increment	INC D	4-6-5
DEC	Decrement	DEC D	4-6-5
MEAN	Mean	MEAN S D n	4-6-6
WAND	Word And	WAND S1 S2 D	4-6-7
WOR	Word OR	WOR S1 S2 D	4-6-7
WXOR	Word eXD3lusive OR	WXOR S1 S2 D	4-6-7
CML	Compliment	CML S D	4-6-8
NEG	Negative	NEG D	4-6-9
Data Shift			
SHL	Arithmetic Shift Left	SHL D n	4-7-1
SHR	Arithmetic Shift Right	SHR D n	4-7-1
LSL	Logic shift left	LSL D n	4-7-2
LSR	Logic shift right	LSR D n	4-7-2
ROL	Rotation shift left	ROL D n	4-7-3
ROR	Rotation shift right	ROR D n	4-7-3
SFTL	Bit shift left	SFTL S D n1 n2	4-7-4
SFTR	Bit shift right	SFTR S D n1 n2	4-7-5
WSFL	Word shift left	WSFL S D n1 n2	4-7-6
WSFR	Word shift right	WSFR S D n1 n2	4-7-7
Data Convert			
WTD	Single word integer converts to double word integer	WTD S D	4-8-1
FLT	16 bits integer converts to float point	FLT S D	4-8-2
DFLT	32 bits integer converts to float point	DFLT S D	4-8-2
FLTD	64 bits integer converts to float point	FLTD S D	4-8-2
INT	Float point converts to integer	INT S D	4-8-3
BIN	BCD converts to binary	BIN S D	4-8-4
BCD	Binary converts to BCD	BCD S D	4-8-5

ASCI	Hex. converts to ASCII	ASCI S D n	4-8-6
HEX	ASCII converts to Hex.	HEX S D n	4-8-7
DECO	Coding	DECO S D n	4-8-8
ENCO	High bit coding	ENCO S D n	4-8-9
ENCOL	Low bit coding	ENCOL S D n	4-8-10
GRY	Binary to Gray code	GRY S D	4-8-11
GBIN	Gray code to binary	GBIN S D	4-8-12
Float Point Operation			
ECMP	Float compare	ECMP S1 S2 D	4-9-1
EZCP	Float Zone compare	EZCP S1 S2 D1 D2	4-9-2
EADD	Float Add	EADD S1 S2 D	4-9-3
ESUB	Float Subtract	ESUB S1 S2 D	4-9-4
EMUL	Float Multiplication	EMUL S1 S2 D	4-9-5
EDIV	Float division	EDIV S1 S2 D	4-9-6
ESQR	Float Square Root	ESQR S D	4-9-7
SIN	Sine	SIN S D	4-9-8
COS	Cosine	COS S D	4-9-9
TAN	Tangent	TAN S D	4-9-10
ASIN	Float Sine	ASIN S D	4-9-11
ACOS	Float Cosine	ACOS S D	4-9-12
ATAN	Float Tangent	ATAN S D	4-9-13
Clock Operation			
TRD	Read RTC data	TRD D	4-10-1
TWR	Write RTC data	TWR D	4-10-2

4-2． Reading Method of Applied Instructions

In this manual, the applied instructions are described in the following manner.

Summary

ADDITION [ADD]			
16 bits	ADD	32 bits	DADD
Execution condition	Normally ON/OFF, Rising/Falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

Operands	Function	Data Type
S1	Specify the data or register address	16 bits/32 bits, BIN
S2	Specify the data or register address	16 bits/32 bits, BIN
D	Specify the register to store the sum result	16 bits/32 bits, BIN

Suitable Soft Components

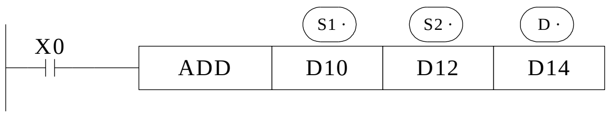
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		
D	●	●	●	●		●	●	●			

*Note: D includes D, HD. TD includes TD, HTD. CD includes CD, HCD, HSCD, HSD. DM includes DM, DHM. DS includes DS, DHS. M includes M, HM, SM. S includes S and HS. T includes T and HT. C includes C and HC.

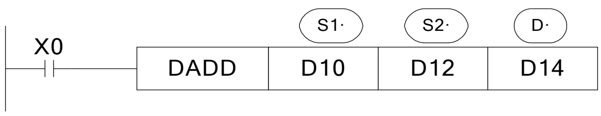
Description

<16 bits instruction>



(D10) ＋ (D12) → (D14)

<32 bits instruction>



(D11D10) ＋ (D13D12) → (D15D14)

Two source data make binary addition and the result data store in object address.
The highest bit of each data is positive (0) and negative (1) sign bit. These data will make addition operation through algebra. Such as 5 + (-8) = -3.

If the result of a calculations is “0”, the “0’ flag acts. If the result exceeds 323,767(16 bits operation) or 2,147,483,648 (32 bits operation), the carry flag acts. (refer to the next page). If the result exceeds -323,768 (16 bits operation) or -2,147,483,648 (32 bits operation), the borrow flag acts (Refer to the next page).
When carry on 32 bits operation, low 16 bits of 32-bit register are assigned, the register address close to the low 16 bits register will be assigned to high 16 bits of 32-bit register. Even number is recommended for the low 16 bits register address.
The source and object can be same register address.
In the above example, when X0 is ON, the addition operation will be excuted in each scanning period.

Related flag

Flag	Name	Function
SM20	Zero	ON: the calculate result is zero OFF: the calculate result is not zero
SM21	Borrow	ON: the calculate result is over 32767(16bits) or 2147483647(32bits) OFF: the calculate result is not over 32767(16bits) or 2147483647(32bits)
SM22	Carry	ON: the calculate result is over 32767(16bits) or 2147483647(32bits) OFF: the calculate result is not over 32767(16bits) or 2147483647(32bits)

Notes

The assignment of the data
The data register of XD, XL series PLC is a single word (16 bit) data register, single word data only occupy one register which is used to single word instruction. The process range is decimal –327,68~327,67, or hex 0000~FFFF.

Single word object instruction D(NUM)

Instruction	D(NUM)	→	Object

Double words (32 bit) occupy two data registers; the two registers’ address is continuous. The process range is: decimal -214,748,364,8~214,748,364,7 or hex 00000000~FFFFFFFF.

Double word object instruction D(NUM+1) D(NUM)

Instruction	D(NUM)	→	Object	Object

The way to represent 32 bits instruction
Add letter “D” before 16 bits instruction to represent 32 bits instruction.
For example:
ADD D0 D2 D4 16 bits instruction
DADD D10 D12 D14 32 bits instruction

※1: It shows the flag bit following the instruction action.
※2: (S·) Source operand which won't change with instruction working
※3: (D·) Destinate operand which will change with instruction working
※4: It introduces the instruction's basic action, using way, applied example, extend function, note items and so on.

4-3. Program Flow Instructions

Mnemonic	Instruction's name	Chapter
CJ	Condition Jump	4-3-1
CALL	Call subroutine	4-3-2
SRET	Subroutine return	4-3-2
STL	Flow start	4-3-3
STLE	Flow end	4-3-3
SET	Open the assigned flow, close the current flow (flow jump)	4-3-3
ST	Open the assigned flow, not close the current flow (Open the new flow)	4-3-3
FOR	Start of a FOR-NEXT loop	4-3-4
NEXT	End of a FOR-NEXT loop	4-3-4
FEND	First End	4-3-5
END	Program End	4-3-5

4-3-1. Condition Jump [CJ]

Summary

As the instruction to execute part of the program, CJ shortens the operation cycle and avoids using the dual coil

Condition Jump [CJ]			
16 bits	CJ	32 bits	-
Execution condition	Normally ON/OFF coil	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

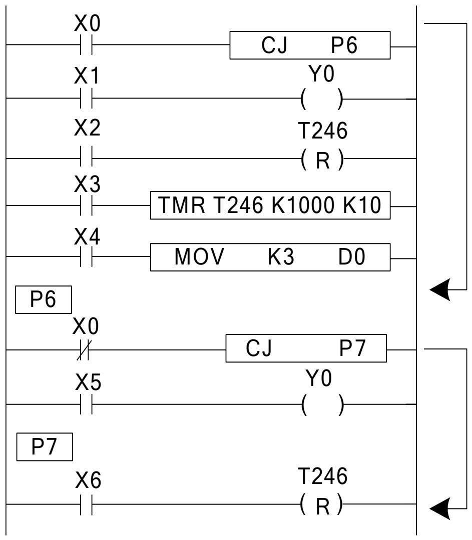
Operands	Function	Data Type
Pn	Jump to the target (with pointer Nr.) P (P0~P9999)	Pointer's Nr.

Suitable Soft Components

Other	Pointer	
	P	I
	●	

Description

In the below graph, if X0 is ON, jump from the first step to the next step behind P6 tag. If X0 is OFF, do not execute the jump instruction;



- In the left graph, Y0 becomes to be dual coil output, but when X0=OFF, X1 activates; when X0=ON, X5 activates
- CJ can't jump from one STL to another STL;
- After driving timer T0~T575, HT0~HT795 and HSC0~HSC30, if executes CJ, continue working, the output activates.
- The Tag must be match when using CJ instruction.

4-3-2. Call subroutine [CALL] and Subroutine return [SRET]

Summary

Call the programs which need to be executed together, decrease the program's steps;

Subroutine Call [CALL]			
16 bits	CALL	32 bits	-
Execution condition	Normally ON/OFF, Rising/Falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Subroutine Return [SRET]			
16 bits	SRET	32 bits	-
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

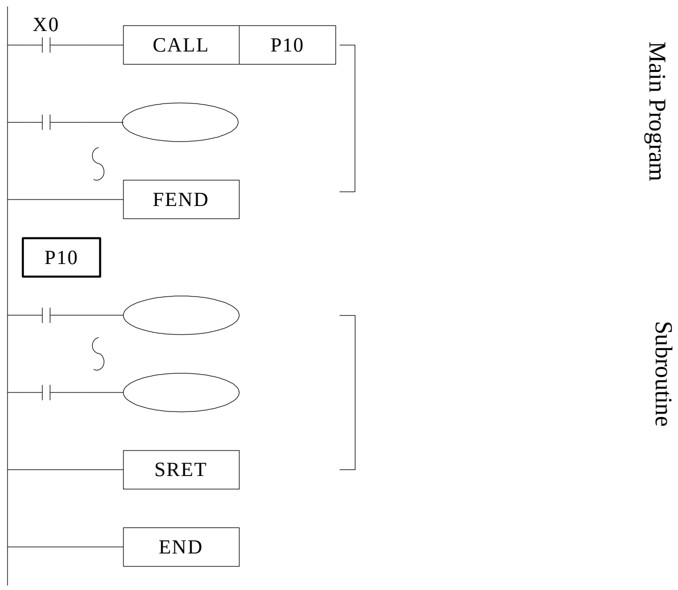
Operands

Operands	Function	Data Type
Pn	Jump to the target (with pointer No.) P (P0~P9999)	Pointer's No.

Suitable Soft Components

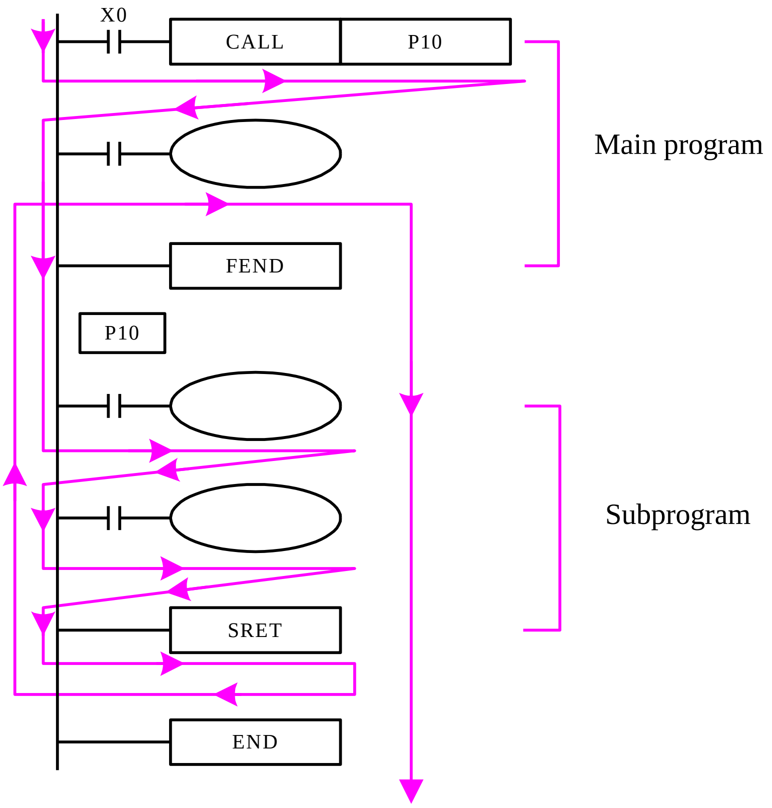
	Pointer	
Others	P	I
	●	

Description



If X0= ON, execute the call instruction and jump to P10. After executing the subroutine, return the original step via SRET instruction.
Program the tag with FEND instruction (will describe this instruction later)
In the subroutine 9 times call is allowed, so totally there can be 10 nestings.
When calling the subprogram, all the timer, OUT, PLS, PLF of the main program will keep the status.
All the OUT, PLS, PLF, timer of subprogram will keep the status when subprogram returning.
Do not write pulse, counter or timer inside the subprogram which cannot be completed in one scan period.

Subprogram executing diagram:



If X0=ON, the program executes as the arrow.
If X0=OFF, the CALL instruction will not work; only the main program works.

The notes to write the subprogram:
Please programming the tag after FEND. Pn is the start of subprogram; SRET is the end of subprogram. CALL Pn is used to call the subprogram. The range of n is 0 to 9999.
The subprogram calling can simplify the programming. If the program will be used in many places, make the program in subprogram and call it.

4-3-3. Flow [SET], [ST], [STL], [STLE]

Summary

Instructions to specify the start, end, open, close of a flow;

Open the specified flow, close the local flow [SET]			
16 bits	SET	32 bits	-
Execution condition	Normally ON/OFF, Rising/Falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Open the specified flow, not close the local flow [ST]			
16 bits	ST	32 bits	-
Execution condition	Normally ON/OFF, Rising/Falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Flow starts [STL]			

16 bits	STL	32 bits	-
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Flow ends [STLE]			
16 bits	STLE	32 bits	-
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

operands

Operands	Function	Data Type
Sn	Jump to the target flow S	Flow No.

3.Suitable Soft Components

Bit

Operand	System					
	X	Y	M*	S*	T*	C*	Dn.m
Sn				●			

*Note: M includes M, HM and SM; S includes S, HS; T includes T and HT; C includes C and HC.

Description

STL and STLE should be used in pairs. STL represents the start of a flow; STLE represents the end of a flow.

Every flow is independent. They cannot be nesting. There is no need to write the flow as the order S0, S1, S2… you can make the order. For example, executing S10, then S5, S0.

After executing of **SET Sxxx** instruction, the flow specified by these instructions is ON.

After executing **RST Sxxx** instruction, the specified flow is OFF.

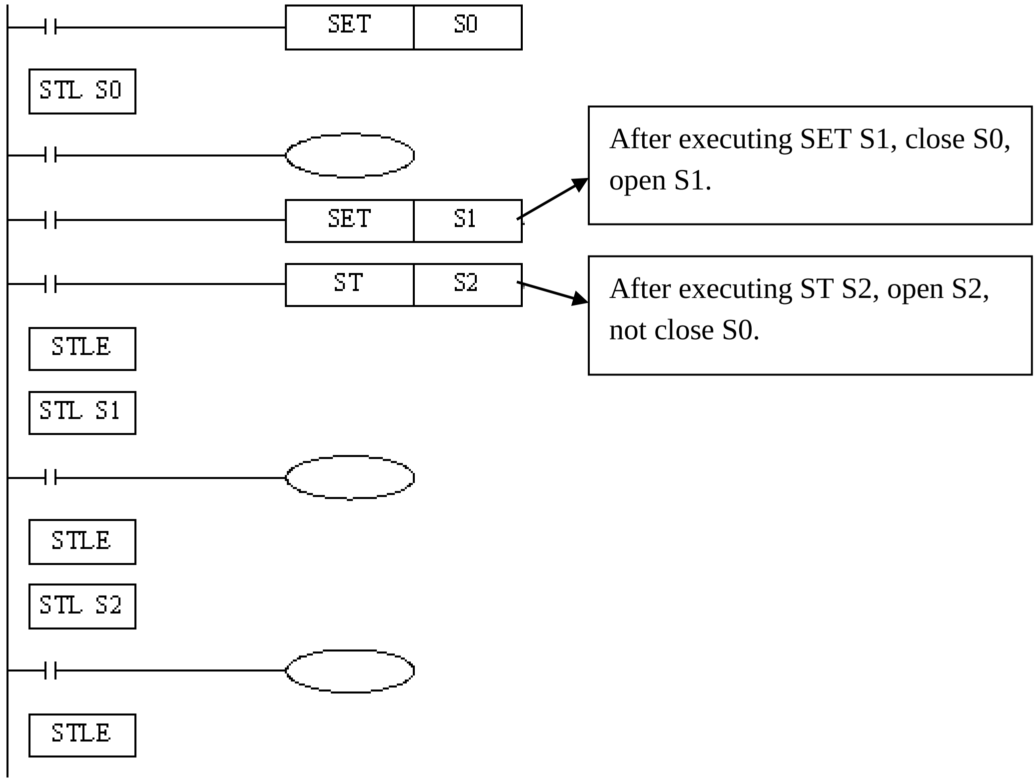
In flow S0, SET S1 close the current flow S0, open flow S1.

In flow S0, ST S2 open the flow S2, but don’t close flow S0.

When flow turns from ON to be OFF, reset OUT, PLS, PLF, not accumulate timer etc. in the flow.

ST instruction is usually used when a program needs to run many flows at the same time.

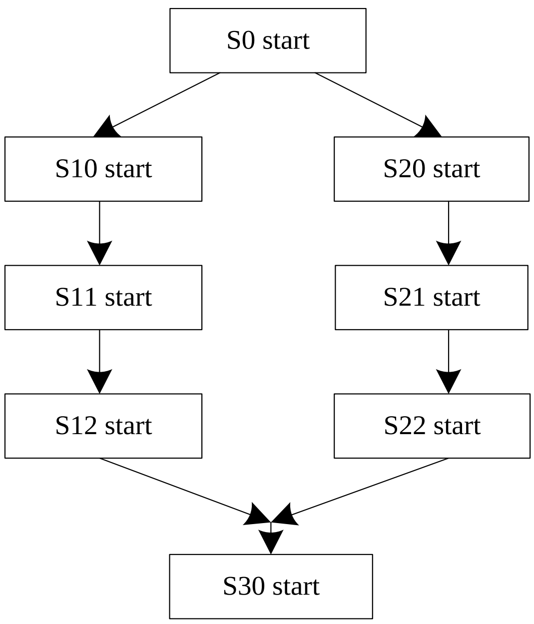
After executing **SET Sxxx** instruction and jump to the next flow, the pulse instructions in the former flow will be closed. (including one-segment, multi-segment, relative or absolute, return to the origin)

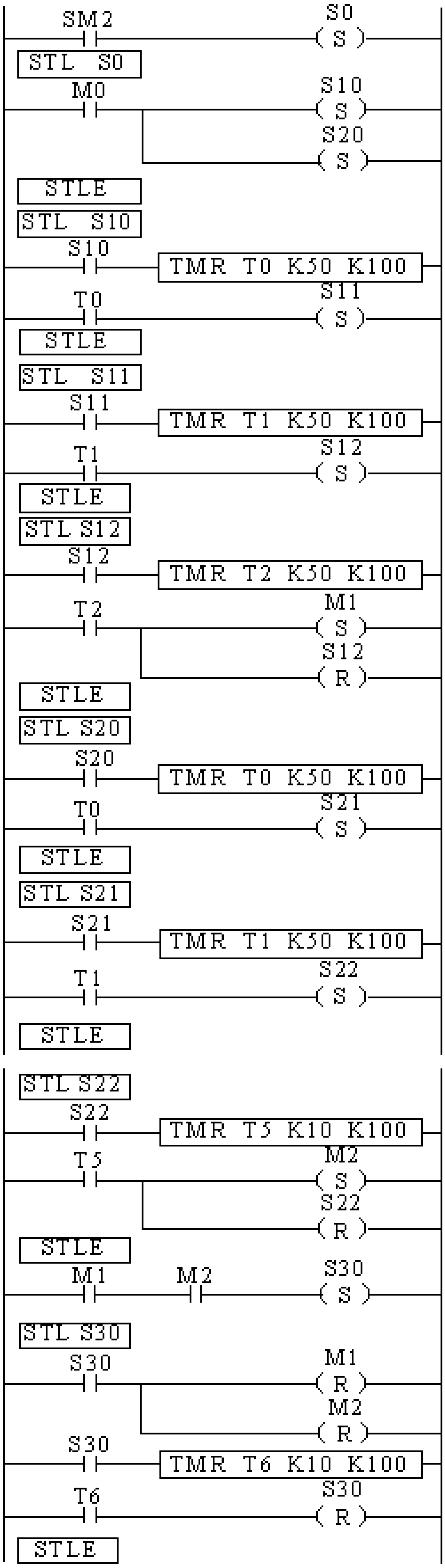


Example

Example 1: the flows run in branch then merge in one flow.

Program diagram:





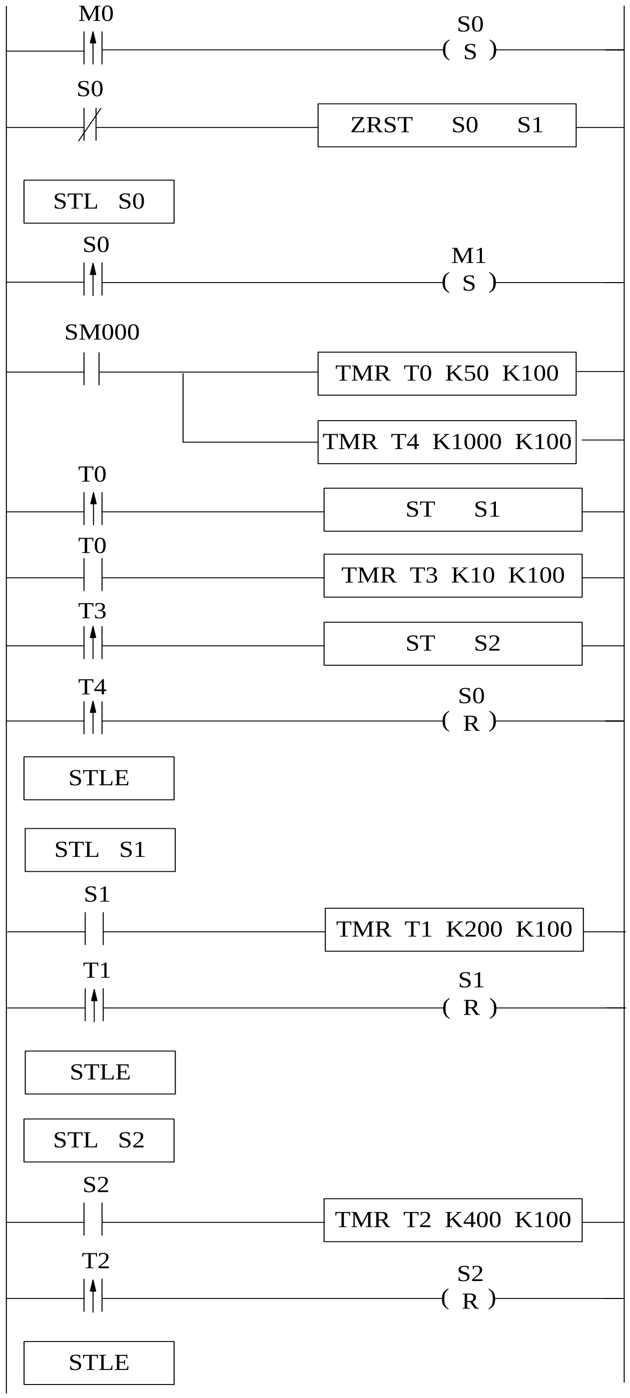
The program explanation:
When SM2 is ON, set ON flow S0. When M0 is ON, set ON flow S10 and S20.

In S10 branch, it runs S10, S11 and S12. Set on M1 means the S10 branch is finished.

In S20 branch, it runs S20, S21 and S22. Set on M2 means the S20 branch is finished.

When both branch S10 and S20 end, set on S30. When S30 end, reset S30.

Example 2: flow nesting. When S0 is running for a while, S1 and S2 start to run; the running status of S1 is kept. When S0 is running for certain time, closes S0 and force close S1 and S2.



4-3-4. [FOR] and [NEXT]

Summary

Loop execute the program between **FOR** and **NEXT** with the specified times;

Loop starts [FOR]			
16 bits	FOR	32 bits	-
Execution condition	Rising/Falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Loop ends [NEXT]			
16 bits	NEXT	32 bits	-
Execution condition	Normally ON/OFF, Rising/Falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

Operands	Function	Data Type
S	Program's loop times between FOR and NEXT	16 bits, BIN

Suitable Soft Components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S	●								●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

FOR.NEXT instructions must be programmed as a pair. Nesting is allowed, and the nesting level is 8.

The program after NEXT will not be executed unless the program between FOR and NEXT is executed for specified times.

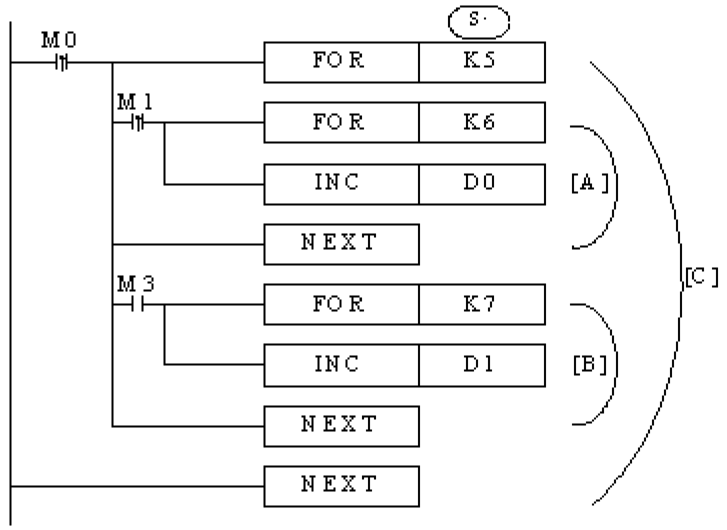
Between FOR and NEXT, LDP, LDF instructions are effective for one time. Every time when M0 turns from OFF to ON, and M1 turns from OFF to ON, [A] loop is executed 5×6=30 times.

Every time if M0 turns from OFF to ON and M3 is ON, [B] loop is executed 5×7=35 times.

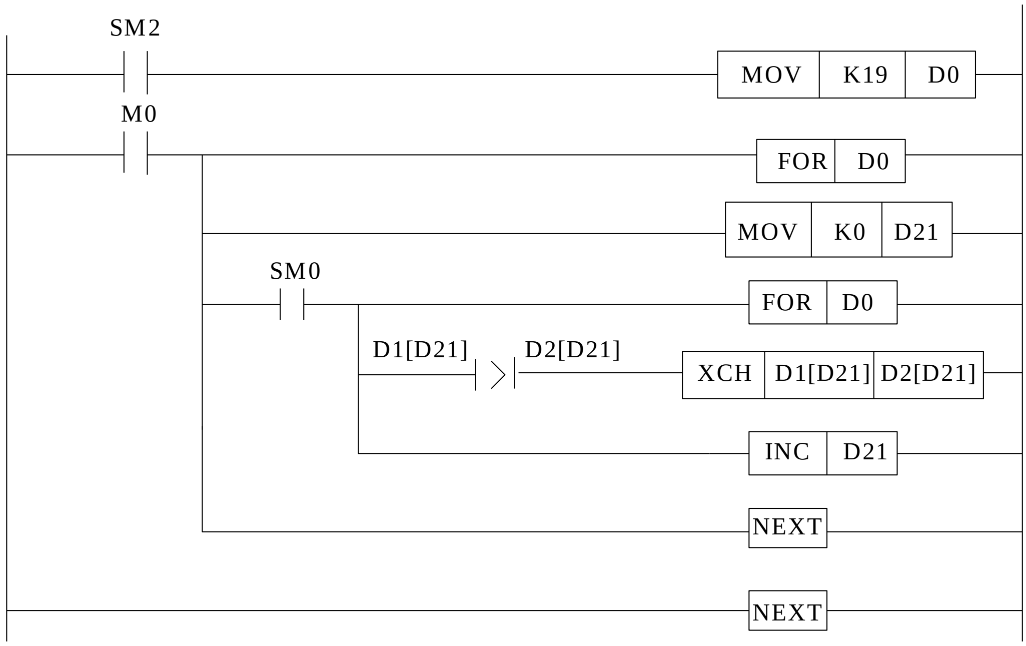
If there are many loop times, the scan cycle will be prolonged. Monitor timer error may occur, please note this.

If NEXT is before FOR, or no NEXT, or NEXT is behind FEND, END, or FOR and NEXT number is not equal, an error will occur.

Between FOR~NEXT, CJ nesting is not allowed. FOR~NEXT must be in pairs in one STL.



Example 1: when M0 is ON, the FOR NEXT starts to sort the numbers in the range of D1 to D20 from small to large. D21 is offset value. If there are many sortings in the program, please use C language to save the programming time and scanning time.




```
LD      SM2                 //SM2 is initial ON coil
MOV     K19     D0          //the times of FOR loop
LD      M0                  //M0 to trigger the FOR loop
MCS                 //
FOR     D0                  //Nesting FOR loop, the loop times is D0
MOV     K0      D21         //the offset starts from 0
LD      SM0                 //SM0 is always ON coil
MCS                 //
FOR             D0          //nesting FOR loop, the loop times is D0
LD>     D1[D21]     D2[D21]     //if the current data is larger than the next, it will be ON
XCH     D1[D21]     D2[D21]   //exchange the two neighbouring data
LD      SM0                 //M8000 is always ON coil
```

```
INC    D21          //increase one for D21
MCR                 //
NEXT                //match the second FOR
MCR                 //
NEXT                //match the first FOR
```

4-3-5. [FEND] and [END]

Summary

FEND means the main program ends, while END means program ends;

main program ends [FEND]			
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
program ends [END]			
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

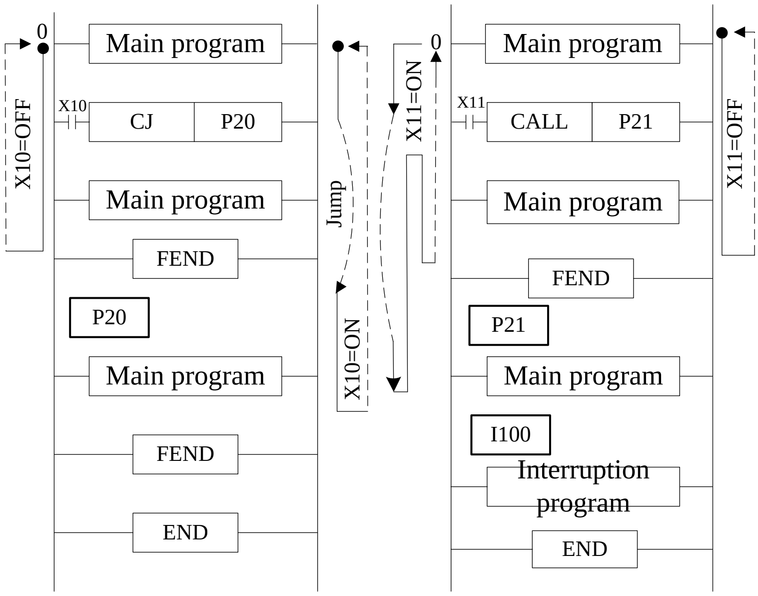
Operands	Function	Data Type
None	-	-

Suitable Soft Components

None

Description

Even though [FEND] instruction represents the end of the main program, the function is same to END to process the output/input, monitor the refresh of the timer, return to program step0.



If program the tag of CALL instruction behind FEND instruction, there must be SRET instruction. If the interrupt pointer program behind FEND instruction, there must be IRET instruction.

After executing CALL instruction and before executing SRET instruction, if execute FEND instruction; or execute FEND instruction after executing FOR instruction and before executing NEXT, an error will occur.

In the condition of using many FEND instructions, please make program or subprogram between the last FEND instruction and END instruction.

4-4．Data compare function

Mnemonic	Function	Chapter
LD＝	LD activates when (S1)＝ (S2)	4-4-1
LD＞	LD activates when (S1)＞ (S2)	4-4-1
LD＜	LD activates when (S1)＜ (S2)	4-4-1
LD＜＞	LD activates when (S1) ≠ (S2)	4-4-1
LD＜＝	LD activates when (S1) ≤ (S2)	4-4-1
LD＞＝	LD activates when (S1) ≥ (S2)	4-4-1
AND＝	AND activates when (S1)＝ (S2)	4-4-2
AND＞	AND activates when (S1)＞ (S2)	4-4-2
AND＜	AND activates when (S1)＜ (S2)	4-4-2
AND＜＞	AND activates when (S1)≠ (S2)	4-4-2
AND＜＝	AND activates when (S1)≤ (S2)	4-4-2
AND＞＝	AND activates when (S1)≥ (S2)	4-4-2
OR＝	OR activates when (S1)＝ (S2)	4-4-3
OR＞	OR activates when (S1)＞ (S2)	4-4-3
OR＜	OR activates when (S1)＜ (S2)	4-4-3
OR＜＞	OR activates when (S1)≠ (S2)	4-4-3

OR<=	OR activates when (S1)≤ (S2)	4-4-3
OR>=	OR activates when (S1)≥ (S2)	4-4-3

4-4-1. LD Compare [LD]

1. Summary

LD is the point compare instruction connected with the generatrix.

LD Compare [LD]			
16 bits	As below	32 bits	As below
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Being compared number address	16/32bits, BIN
S2	Comparand address	16/32 bits, BIN

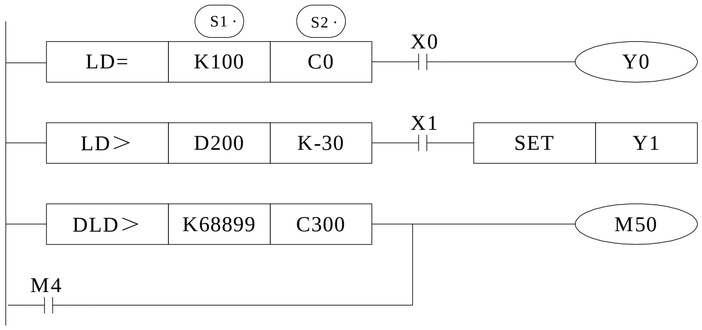
3. Suitable soft components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S1	●	●	●	●	●	●	●	●	●		
	S2	●	●	●	●	●	●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD;
DM includes DM, DHM; DS includes DS, DHS.

Description

16 bits instruction	32 bits instruction	Activate Condition	Not Activate Condition
LD=	DLD=	(S1)=(S2)	(S1)≠(S2)
LD>	DLD>	(S1)> (S2)	(S1)≤(S2)
LD<	DLD<	(S1)< (S2)	(S1)≥(S2)
LD<>	DLD<>	(S1)≠(S2)	(S1) = (S2)
LD<=	DLD<=	(S1)≤(S2)	(S1) > (S2)
LD>=	DLD>=	(S1)≥(S2)	(S1) < (S2)



Note Items

When the source data's highest bit (16 bits: b15, 32 bits: b31) is 1, the data is seemed to a negative number.

The comparison of 32 bits counter should use 32 bits instruction. If using 16 bits instruction, the program or operation will be error.

4-4-2. Serial Compare [AND]

Summary

AND: serial connection comparison instruction.

AND Compare [AND]			
16 bits	As Below	32 bits	As Below
Execution condition	Normally ON/OFF coil	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

Operands	Function	Data Type
S1	Being compared number address	16/32bit, BIN
S2	Comparand address	16/32bit, BIN

suitable soft components

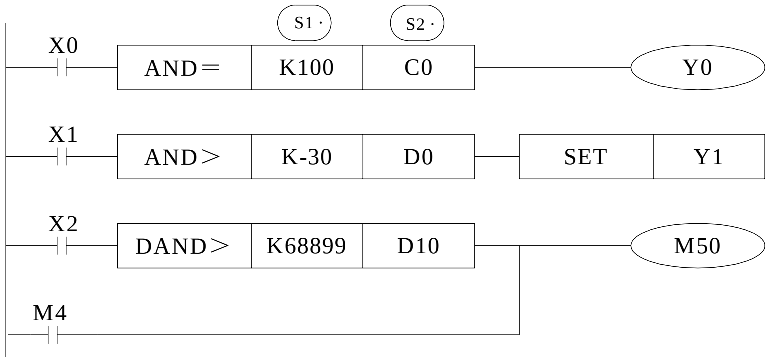
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

16 bits instruction	32 bits instruction	Activate Condition	Not Activate Condition
AND＝	DAND＝	(S1)＝ (S2)	(S1) ≠ (S2)
AND＞	DAND＞	(S1) ＞ (S2)	(S1) ≤ (S2)
AND＜	DAND＜	(S1) ＜ (S2)	(S1) ≥ (S2)
AND＜＞	DAND＜＞	(S1) ≠ (S2)	(S1)＝ (S2)
AND＜＝	DAND＜＝	(S1) ≤ (S2)	(S1) ＞ (S2)
AND＞＝	DAND＞＝	(S1) ≥ (S2)	(S1) ＜ (S2)



Note Items

When the source data’s highest bit (16 bits: b15, 32 bits: b31) is 1, it is seemed to negative number.

The comparison of 32 bits counter should use 32 bits instruction. If using 16 bits instruction, the program or operation will be error.

4-4-3． Parallel Compare [OR]

1. Summary

OR: parallel connection comparison instruction.

Parallel Compare [OR]			
16 bits	As below	32 bits	As below
Execution condition	-	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Being compared number address	16/32 bit,BIN
S2	Comparand address	16/32 bit,BIN

3. Suitable soft components

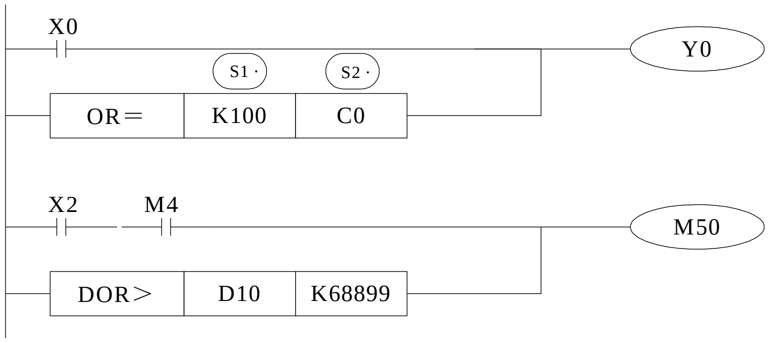
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

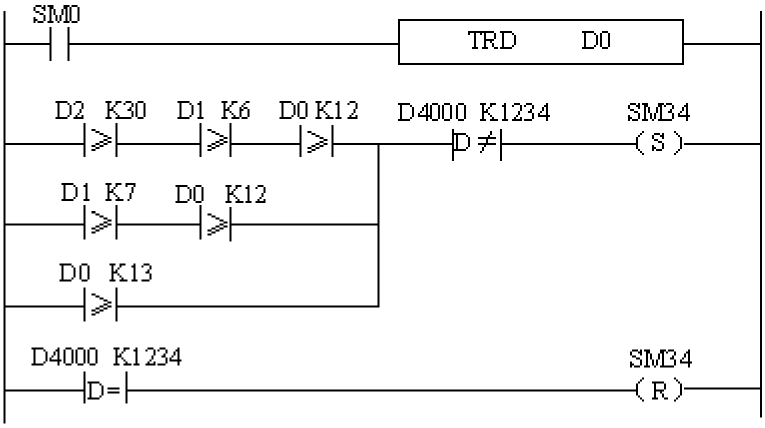
16 bits instruction	32 bits instruction	Activate Condition	Not Activate Condition
OR＝	DOR＝	(S1)＝ (S2)	(S1) ≠ (S2)
OR＞	DOR＞	(S1) ＞ (S2)	(S1) ≤ (S2)
OR＜	DOR＜	(S1) ＜ (S2)	(S1) ≥ (S2)
OR＜＞	DOR＜＞	(S1) ≠ (S2)	(S1) ＝ (S2)
OR＜＝	DOR＜＝	(S1) ≤ (S2)	(S1) ＞ (S2)
OR＞＝	DOR＞＝	(S1) ≥ (S2)	(S1) ＜ (S2)



Note Items

When the source data’s highest bit (16 bits: b15, 32 bits: b31) is 1, it is seemed to negative number.
The comparison of 32 bits counter should use 32 bits instruction. If using 16 bits instruction, the program or operation will be error.

Example: forbid the outputs when it reaches the certain time. In the below program, when the date is June 30th, 2012, all the outputs will be disabled. The password 1234 is stored in (D4000, D4001). When the password is correct, all the outputs are enabled.




```
LD       SM0                  //SM0 is always ON coil
TRD      D0                   //read the RTC (real time clock) value and store in D0~D6
LD>=     D2      K30          //RTC date ≥30
AND>=    D1      K6           //RTC month ≥6
AND>=    D0      K12          //RTC year ≥12
LD>=     D1      K7           //or RTC month ≥ 7
AND>=    D0      K12          //RTC year ≥ 12
ORB                           //or
OR>=     D0      K13          //RTC year ≥ 13
DAND<>   D4000   K1234        //and password ≠1234
SET      SM34                 //set ON M34, all the outputs are disabled
DLD=     D4000   K1234        //password=1234, correct password
RST      SM34                 //reset M34, all the outputs are enabled
```

4-5. Data Move Instructions

Mnemonic	Function	Chapter
CMP	Data compare	4-5-1
ZCP	Data zone compare	4-5-2
MOV	Move	4-5-3
BMOV	Data block move	4-5-4
PMOV	Data block move (with faster speed)	4-5-5
FMOV	Fill move	4-5-6
EMOV	Float number move	4-5-7
FWRT	FlashROM written	4-5-8
MSET	Zone set	4-5-9
ZRST	Zone reset	4-5-10
SWAP	The high and low byte of the destinated devices are exchanged	4-5-11
XCH	Exchange two data	4-5-12

4-5-1. Data Compare [CMP]

1. Summary

Compare the two data, output the result.

Data compare [CMP]			
16 bits	CMP	32 bits	DCMP
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Specify the data (to be compared) or soft component's address code	16 bit,BIN
S	Specify the comparand's value or soft component's address code	16 bit,BIN
D	Specify the compare result's address code	bit

3. Suitable soft component

Word

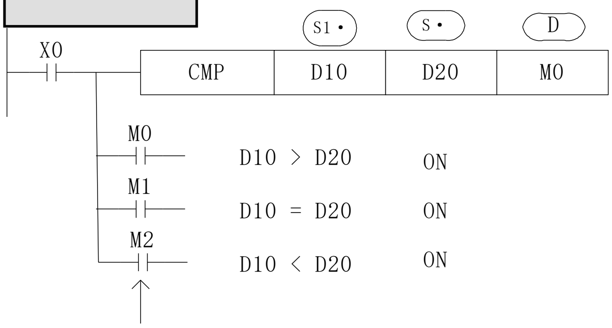
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●	●	●	●	●	●		
S	●	●	●	●	●	●	●	●	●		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD;
DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Description



Even X0=OFF to stop CMP instruction,
M0~M2 will keep the original status

Compare data S1· and S· , show the result in three soft components starting from D· : D· , D· +1, D· +2: the three soft components will show the compare result.

4-5-2. Data zone compare [ZCP]

1. Summary

Compare the current data with the data in the zone, output the result.

Data Zone compare [ZCP]			
16 bits	ZCP	32 bits	DZCP
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	The low limit of zone	16 bit, BIN
S2	The high limit of zone	16 bit, BIN
S	The current data address	16 bit, BIN
D	The compare result	bit

3. Suitable soft components

Word

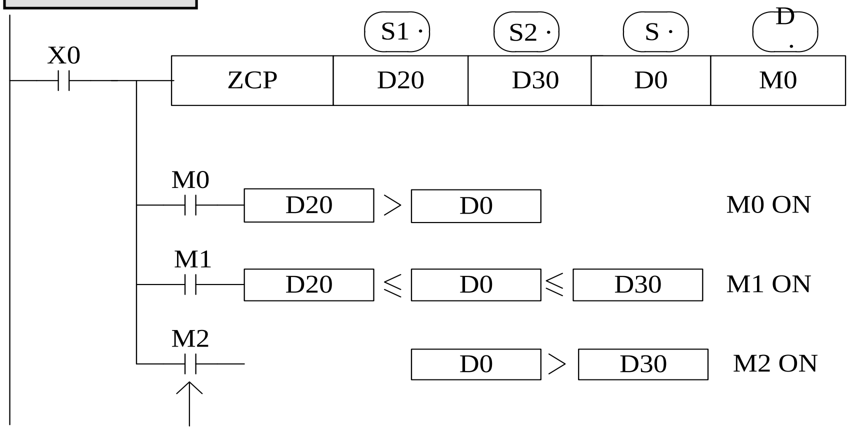
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		
S	●	●	●	●	●	●	●	●	●		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Description



Even X0=OFF stop ZCP instruction, M0~M2
will keep the original status

Compare (S·) with (S1) and (S2), output the three results starting from (D·)
(D·), (D·)+1, (D·)+2 : store the three results

4-5-3. MOV [MOV]

1. Summary

Move the specified data to the other soft components

MOV [MOV]			
16 bits	MOV	32 bits	DMOV
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Specify the source data or register's address code	16 bit/32 bit, BIN
D	Specify the target soft component's address code	16 bit/32 bit, BIN

3. Suitable soft component

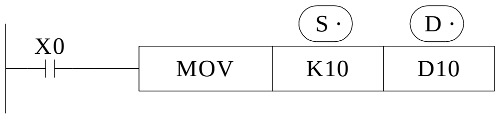
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●	●	
D	●		●	●		●	●	●			●

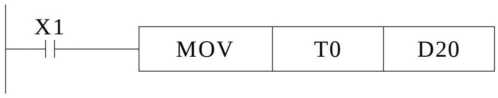
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

Move the source data to the target
When X0 is off, the data will not change
Move K10 to D10

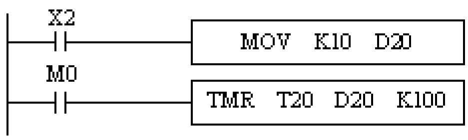


<read the counter or timer current value>



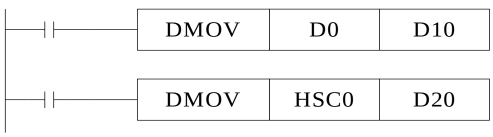
(The current value of T0) → (D20)
The same as counter

<indirect set the timer value>



(K10) (D20)
D20=K10

< Move the 32bits data >
Please use DMOV when the value is 32 bits, such as MUL instruction, high speed counter…



(D1, D0) → (D11, D10)
(the current value of HSC0) → (D21, D20)

4-5-4. Data block Move [BMOV]

1. Summary

Move the data block to other soft component

Data block move [BMOV]			
16 bits	BMOV	32 bits	-
Execution condition	Normally ON/OFF coil, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Specify the source data block or soft component address code	16 bits, BIN; bit
D	Specify the target soft components address code	16 bits, BIN; bit
n	Specify the move data’s number	16 bits, BIN;

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n	●		●	●	●		●	●	●		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S	●	●	●				
D	●	●	●				

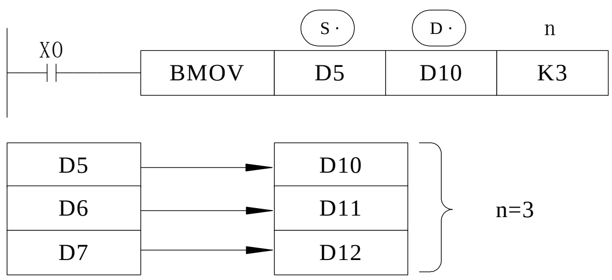
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T and HT; C includes C and HC.

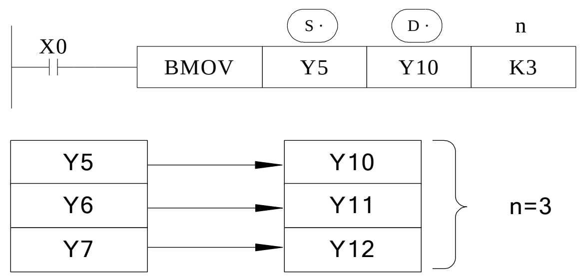
Description

Move the source data block to the target data block. The data quantity is n.

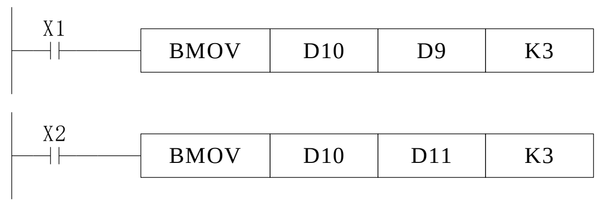
<word move>

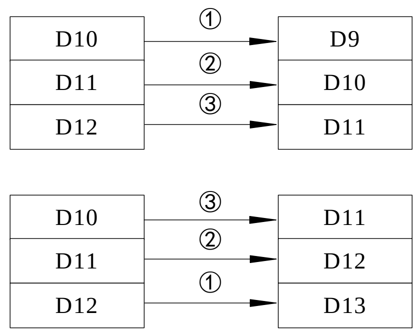


<bit move>



As the following picture, when the data address overlapped, the instruction will do from 1 to 3.





4-5-5. Data block Move [PMOV]

1. Summary

Move the specified data block to the other soft components

Data block mov[PMOV]			
16 bits	PMOV	32 bits	-
Execution condition	Normally ON/OFF coil, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Specify the source data block or soft component address	16 bits, BIN; bit
D	Specify the target soft components address	16 bits, BIN; bit
n	Specify the data quantity	16 bits, BIN;

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n	●		●	●		●	●	●	●		

Bit

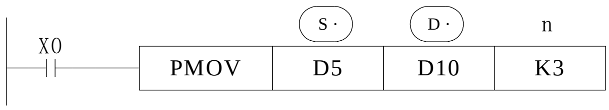
Operand	System						
	X	Y	M*	S*	T*	C	Dn.m
S	●	●	●				
D	●	●	●				

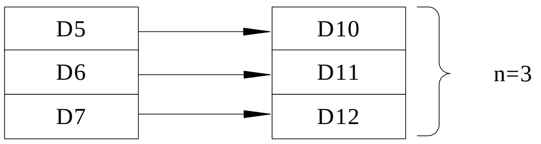
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T and HT; C includes C and HC.

Description

Move the source data block to target data block, the data quantity is n





The function of PMOV and BMOV is mostly the same, but the PMOV execution speed is faster.

PMOV finish in one scan cycle, when executing PMOV, close all the interruptions.

Mistake may happen if the source address and target address are overlapped.

4-5-6. Fill Move [FMOV]

1. Summary

Move the specified data to the other soft components

Fill Move [FMOV]			
16 bits	FMOV	32 bits	DFMOV
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Specify the source data or soft component address	16/32 bits, BIN;
D	Specify the target soft components address	16/32 bits, BIN;
n	Specify the move data's number	16/32 bits, BIN;

3. Suitable soft component

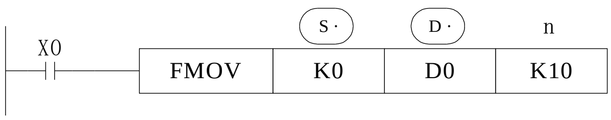
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			
n	●		●	●		●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

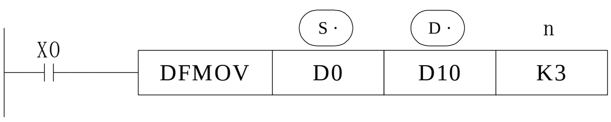
Description

<16 bits instruction>



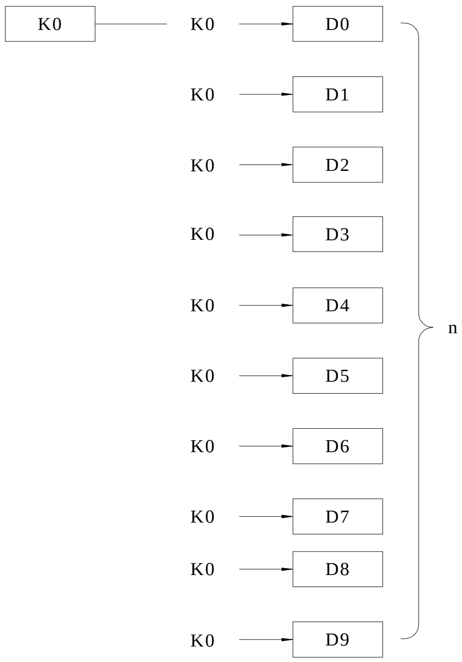
Move K0 to D0~D9, copy a single data device to a range of destination device
Move the source data to target data, the target data quantity is n
If the set range exceeds the target range, move to the possible range

<32 bits instruction >

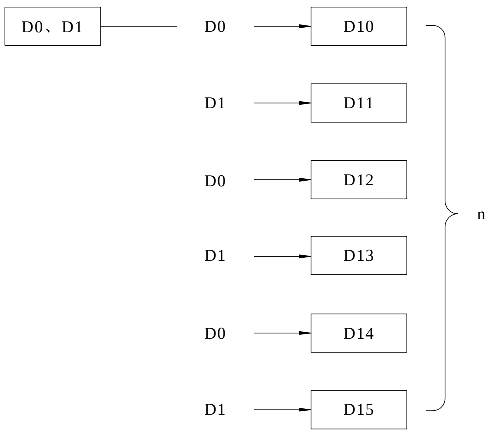


Move D0.D1 to D10.D11:D12.D13:D14.D15.

<16 bits Fill Move >



<32 bits Fill move>



4-5-7. Floating move [EMOV]

Summary

Move the float number to target address

Floating move [EMOV]			
16 bits	-	32 bits	EMOV
Execution condition	Normally on/off, edge trigger	Suitable models	XD, XL
Hardware	-	Software	-

Operands

Operand	Function	Type
S	Source soft element address	32 bits, BIN
D	Destination soft element address	32 bits, BIN

Suitable soft element

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

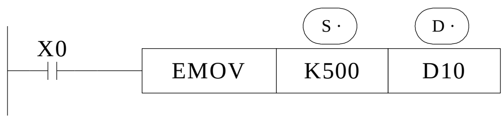
<32 bits instruction>

Binary floating → binary floating

(D1, D0) → (D11, D10)

X0 is ON, send the floating number from (D1, D0) to (D11, D10).

X0 is OFF, the instruction doesn't work



(K500) → (D11, D10)

If constant value K, H is source soft element, they will be converted to floating number. K500 will be converted to floating value.

4-5-8. FlashROM Write [FWRT]

1. Summary

Write the specified data to FlashRom register.

FlashROM Write [FWRT]			
16 bits	FWRT	32 bits	DFWRT
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	The data write in the source or save in the soft element	16 bits/32 bits, BIN
D	target soft element	16 bits/32 bits
D1	target soft element start address	16 bits/32 bits
D2	Write in data quantity	16 bits/32 bits, BIN

3. Suitable soft components

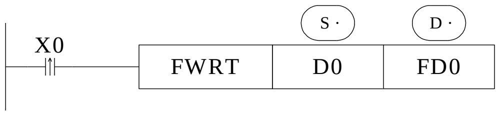
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●		
D		●									
D1		●									
D2	●		●	●	●	●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

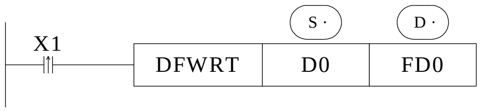
Description

< Written of single word >



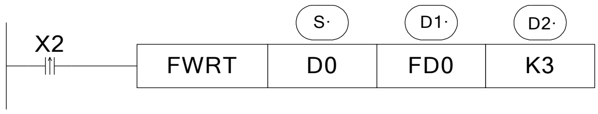
Write value from D0 to FD0

<Written of double words>



Write value from D0,D1 to FD0,FD1

<Written of multi-word>



Write value from D0, D1, D2 to FD0, FD1, FD2

※1: FWRT instruction only can write data into FlashRom register. FlashRom can keep the data even the power supply is off. It can store the important technical parameters.
※2: Written of FWRT needs a long time, about 150ms, so frequently write-in is not recommended
※3: The written time of Flashrom is about 1,000,000 times. So we suggest using edge signal (LDP, LDF etc.) to activate the instruction.
※4: Frequently write-in will damage the FlashRom.

4-5-9．Zone set [MSET]

Summary

Set the soft element in certain range

Multi-set [MSET]			
16 bits	MSET	32 bits	-
Execution condition	Normally ON/OFF; falling or rising pulse edge signal	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D1	Start soft element address	bit
D2	End soft element address	bit

3. Suitable soft components

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D1	●	●	●	●	●	●	
D2	●	●	●	●	●	●	

*Notes: M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Description

Set ON M10~M120

Set the coil from M10 to M120

D1 D2 are specified as the same type of soft component, and D1 < D2

When D1 > D2 , will not run Zone set, but set SM409 SD409 = 2

4-5-10. Zone reset [ZRST]

Summary

Reset the soft element in the certain range

Multi-reset [ZRST]			
16 bits	ZRST	32 bits	-
Execution condition	Normally ON/OFF, falling or rising pulse edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D1	Start address of soft element	Bit, 16 bits,BIN
D2	End address of soft element	Bit, 16 bits,BIN

3. Suitable soft components

Word

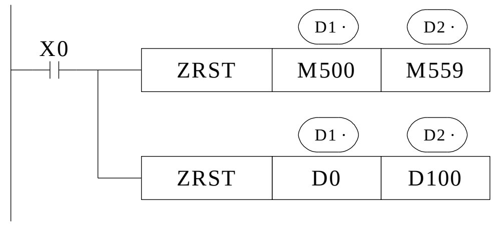
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D1	●				●	●	●				
D2	●			●	●	●	●				

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D1	●	●	●	●	●	●	
D2	●	●	●	●	●	●	

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Description



Reset M500~M559

Reset D0~D100

D1 D2 Are specified as the same type of soft units, and D1 < D2
When D1 > D2 , only reset the specified soft unit, and set SM409, SD409 = 2.

Other Reset Instruction

RST can reset one soft component. The operand can be Y, M, HM, S, HS, T, HT, C, HC, TD, HTD, CD, HCD, D, HD
FMOV can move 0 to these soft components: DX, DY, DM, DS, T(TD), HT(HTD), C(CD), HC(HCD), D, HD.

4-5-11. Swap the high and low byte [SWAP]

1. Summary

Swap the high and low byte of specified register

High and low byte swap [SWAP]			
16 bits	SWAP	32 bits	-
Execution condition	Falling or rising pulse edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

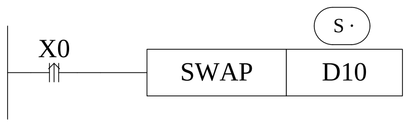
Operands	Function	Data Type
S	The address of the soft element	16 bits; BIN

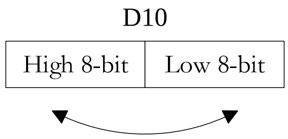
3. Suitable soft components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S	●		●	●							

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description





Exchange the high 8-bit and low 8-bit of 16-bit register.
If this instruction is activated by normal ON/OFF coil, the instruction will be executed in every scanning period when X0 is ON. Falling or rising pulse is recommended to activate the instruction.

4-5-12． Exchange [XCH]

1. Summary

Exchange the data in two soft element

Exchange [XCH]			
16 bits	XCH	32 bits	DXCH
Execution condition	Rising or falling pulse edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D1	The soft element address	16 bits/32 bits, BIN
D2	The soft element address	16 bits/32 bits, BIN

3. Suitable soft component

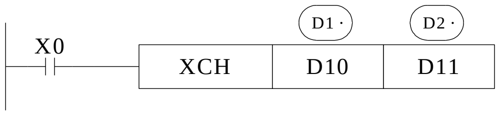
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D1	●		●	●		●	●	●			
D2	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

<16 bits instruction>

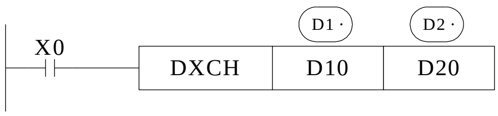


Before (D10) =100 → After (D10) =101
(D11) =101 (D11) =100

The contents of the two destination devices D1 and D2 are swapped,

When X0 is ON, the instruction will be executed in every scanning period. Falling or rising pulse is recommended to activate the instruction.

<32 bits instruction >



32 bits instruction [DXCH] swaps the dword value D10, D11 and D20, D21.

Before (D10) =100 → after (D10) =200
(D11) =1 (D11D10) =65636 (D11) =10 (D11D10) =655460
(D20) =200 (D20) =100
(D21) =10 (D21D20) =655460 (D21) =1 (D21D20) =65636

4-6. Data Operation Instructions

Mnemonic	Function	Chapter
ADD	Addition	4-6-1
SUB	Subtraction	4-6-2
MUL	Multiplication	4-6-3
DIV	Division	4-6-4
INC	Increment	4-6-5
DEC	Decrement	4-6-5
MEAN	Mean	4-6-6
WAND	Logic Word And	4-6-7
WOR	Logic Word Or	4-6-7
WXOR	Logic Exclusive Or	4-6-7
CML	Compliment	4-6-8
NEG	Negation	4-6-9

4-6-1 Addition [ADD]

1. Summary

Add two numbers and store the result

Add [ADD]			
16 bits	ADD	32 bits	DADD
Execution condition	Normal ON/OFF/falling or rising pulse edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
Three operands		
S1	The add operation data address	16 bit/32 bit, BIN
S2	The add operation data address	16 bit/32bit, BIN
D	The result address	16 bit/32bit, BIN
Two operands		
D	Be Added data and result data address	16 bit/32bit, BIN
S1	Add data address	16 bit/32bit, BIN

3. Suitable soft components

Word

Operand	System								constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
Three operands											
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			
Two operands											
D	●										
S1	●	●							●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

< Three operands>

(D10) ＋ (D12) → (D14)

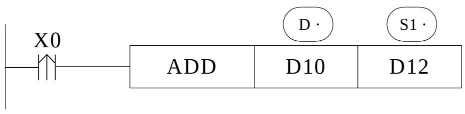
Two source data do binary addition and send the result to target address. Each data’s highest bit is the sign bit, 0 stands for positive, 1 stands for negative. All calculations are algebraic processed. (5+ (-8) =-3)

If the result of a calculation is “0”, the “0” flag acts. If the result exceeds 323767 (16 bits limit) or 2147483647 (32 bits limit), the carry flag acts. (refer to the next page). If the result exceeds –323768（16 bits limit）or –2147483648 (32 bits limit), the borrow flag acts (refer to the next page).

When doing 32 bits operation, word device’s low 16 bits are assigned; the device close to the preceding device’s is the high bits. To avoid ID repetition, we recommend you assign device’s ID to be even number.

The source and target address can be the same. In the above example, when X0 is ON, the instruction will be executed in every scanning period.

<Two operands>

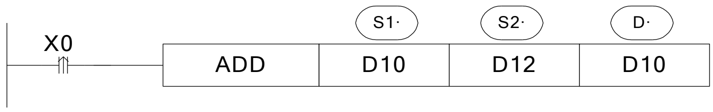


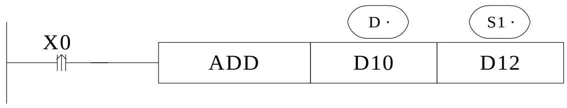
(D10)＋ (D12) → (D10)

Two source data do binary addition and send the result to addend data address. Each data’s highest bit is the sign bit, 0 stands for positive, 1 stands for negative. All calculations are algebraic processed. (5+ (-8) =-3)

If the result of a calculation is “0”, the “0” flag acts. If the result exceeds 323767 (16 bits limit) or 2147483647 (32 bits limit), the carry flag acts. (refer to the next page). If the result

exceeds –323768（16 bits limit）or –2147483648 (32 bits limit), the borrow flag acts (refer to the next page).
When doing 32 bits operation, word device’s low 16 bits are assigned; the device close to the preceding device’s is the high bits. To avoid ID repetition, we recommend you assign device’s ID to be even number.
In the above example, when X0 is ON, the instruction will be executed in every scanning period. The rising or falling pulse edge is recommended to activate the instruction.





The two instructions are the same.

Related flag

Flag meaning

Flag	Name	Function
SM020	Zero	ON: the calculate result is zero OFF: the calculate result is not zero
SM021	Borrow	ON: the calculate result is over -32768(16 bit) or -2147483648(32bit) OFF: the calculate result is less than -32768(16 bit) or -2147483648(32bit)
SM022	Carry	ON: the calculate result is over 32768(16 bit) or 2147483648(32bit) OFF: the calculate result is less than 32768(16 bit) or 2147483648(32bit)

4-6-2． Subtraction [SUB]

1. Summary

Two numbers do subtraction, store the result

Subtraction [SUB]			
16 bits	SUB	32 bits	DSUB
Execution condition	Normally ON/OFF/rising or falling pulse edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

Operands	Function	Data Type
Three operands		
S1	The sub operation data address	16 bits /32 bits,BIN
S2	The sub operation data address	16 bits /32 bits,BIN
D	The result address	16 bits /32 bits,BIN

Two operands		
D	Be subtracted data and result address	16 bits /32 bits,BIN
S1	Subtract data address	16 bits /32 bits,BIN

Suitable soft component

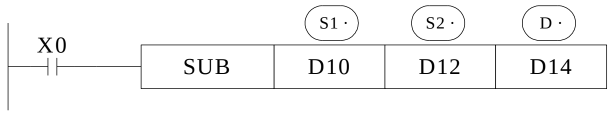
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
Three operands											
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			
Two operands											
D	●										
S1	●	●							●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

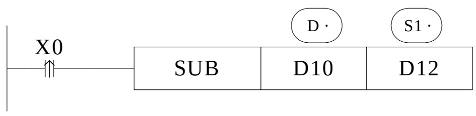
<Three operands>



(D10) — (D12) → (D14)

S1· appoint the soft unit's content, subtract the soft unit's content appointed by S2· in the format of algebra. The result will be stored in the soft unit appointed by D· (5-(-8)=13).
The action of each flag, the setting method of 32 bits operation's soft units are both the same with the preceding ADD instruction.
The importance is: in the preceding program, if X0 is ON, SUB operation will be executed every scan cycle.
Refer to chapter 4-6-1 for flag action and functions.

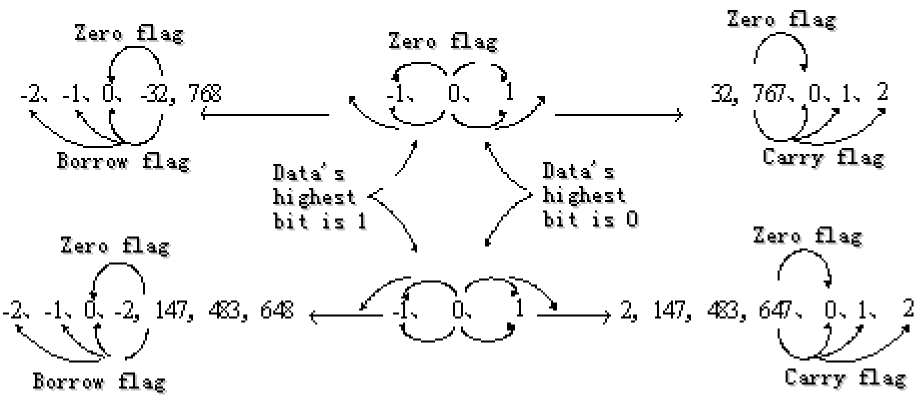
<Two operands>



(D10) — (D12) → (D10)

S1· appoint the soft unit's content, subtract the soft unit's content appointed by S2· in the format of algebra. The result will be stored in the soft unit appointed by D· (5-(-8)=13)
The action of each flag, the setting method of 32 bits operation's soft units are both the same with the preceding ADD instruction.
The importance is: in the preceding program, if X0 is ON, SUB operation will be executed every scan cycle. Rising or falling pulse edge is recommended to activate the instruction.
Refer to chapter 4-6-1 for flag action and functions.

The relationship of the flag's action and vale's positive/negative is shown below:



4-6-3．Multiplication [MUL]

1. Summary

Multiply two numbers, store the result

Multiplication [MUL]			
16 bits	MUL	32 bits	DMUL
Execution condition	Normally ON/OFF / pulse edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	The multiplication operation data address	16 bits/32bits,BIN
S2	The multiplication operation data address	16 bits/32bits,BIN
D	The result address	16 bits/32bits,BIN

3. Suitable soft component

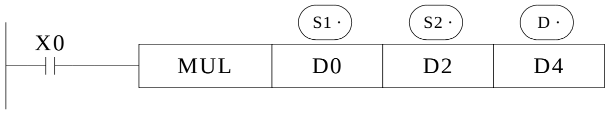
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●	●	●	●	●	●		
S2	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

<16 bits Operation>



BIN BIN BIN

(D0) × (D2) → (D5, D4)

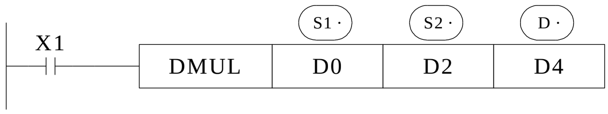
16 bits 16 bits → 32 bits

The contents of the two source devices are multiplied together and the result is stored at the destination device in the format of 32 bits. As the above chart: when (D0)=8, (D2)=9, (D5, D4) =72.

The result’s highest bit is the symbol bit: positive (0), negative (1).

In the above example, when X0 is ON, the instruction will be executed in every scanning period.

<32 bits Operation >



BIN BIN BIN

(D1, D0) × (D3, D2) → (D7, D6, D5, D4)

32 bits 32 bits → 64 bits

When use 32 bits operation, the result is stored at the
bits.

Even use word device, 64 bits results can’t be monitored.

Please change to floating value operation for this case.

4-6-4. Division [DIV]

1. Summary

Divide two numbers and store the result

Division [DIV]			
16 bits	DIV	32 bits	DDIV
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	The divide operation data address	16 bits / 32 bits, BIN
S2	The divide operation data address	16 bits /32 bits, BIN
D	The result address	16 bits /32 bits, BIN

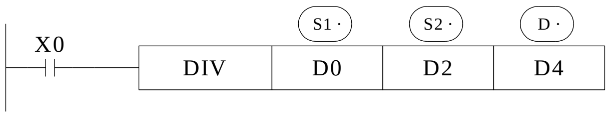
3. Suitable soft components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S1	●	●	●	●	●	●	●	●	●		
	S2	●	●	●	●	●	●	●	●	●		
	D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

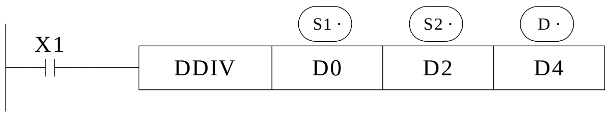
<16 bits operation >



Dividend		Divisor		Result		Remainder
BIN		BIN		BIN		BIN
(D0)	÷	(D2)	→	(D4)	---	(D5)
16 bits		16 bits		16 bits		16 bits

S1· appoints the dividend soft component, S2· appoints the divisor soft component, D· and the next address appoint the soft component of the result and the remainder.
In the above example, if input X0 is ON, devision operation is executed every scan cycle.

<32 bits operation >



Dividend		Divisor	Result		Remainder
BIN		BIN	BIN		BIN
(D1, D0)	÷	(D3, D2)	(D5, D4)	---	(D7, D6)
32 bits		32 bits	32 bits		32 bits

The dividend is composed by the device appointed by S1· and the next one. The divisor is composed by the device appointed by S2· and the next one. The result and the remainder are stored in the four sequential devices, the first one is appointed by D·
If the value of the divisor is 0, the instruction will be error.
The highest bit of the result and remainder is the symbol bit (positive:0, negative: 1). When any of the dividend or the divisor is negative, then the result will be negative. When the dividend is negative, then the remainder will be negative.

4-6-5. Increment [INC] & Decrement [DEC]

1. Summary

Increase or decrease the number

Increase one [INC]			
16 bits	INC	32 bits	DINC
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Decrease one [DEC]			
16 bits	DEC	32 bits	DDEC
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D	The increase or decrease data address	16 bits / 32bits,BIN

3. Suitable soft components

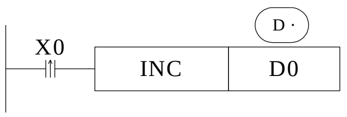
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

< Increment [INC]>

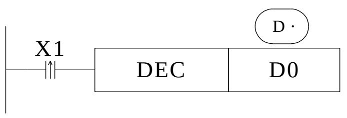


(D0) +1→(D0)

D· will increase one when X0 is ON.

For 16 bits operation, when +32767 increase one, it will become -32768; for 32 bits operation, +2147483647 increases one is -2147483647. The flag bit will act.

<Decrement [DEC]>



(D0) −1 → (D0)

D· will decrease one when X1 is ON.

-32767 or -2147483647 decrease one, the result will be +32767 or +2147483647. The flag bit will act.

4-6-6. Mean [MEAN]

1. Summary

Get the mean value of data

Mean [MEAN]			
16 bits	MEAN	32 bits	DMEAN
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	The source data start address	16 bits, BIN
D	The mean result address	16 bits, BIN
n	The data quantity	16 bits, BIN

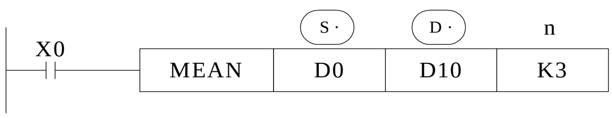
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*注	FD	TD*注	CD*注	DX	DY	DM*注	DS*注	K/H	ID	QD
S	●	●	●	●		●	●	●			
D	●		●	●		●	●	●			
n									●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



$$\frac{(D0) + (D1) + (D2)}{3} \longrightarrow (D10)$$

Store the mean value of source data (source sum divide by source quantity n). give the remainder .

The n cannot larger than soft component quantity, otherwise there will be error.

4-6-7. Logic AND [WAND], Logic OR[WOR], Logic Exclusive OR [WXOR]

1. Summary

Do logic AND, OR, XOR for data

Logic AND [WAND]			
16 bits	WAND	32 bits	DWAND
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Logic OR[WOR]			
16 bits	WOR	32 bits	DWOR
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Logic Exclusive OR [WXOR]			
16 bits	WXOR	32 bits	DWXOR
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	The operation data address	16bit/32bit,BIN
S2	The operation data address	16bit/32bit,BIN
D	The result address	16bit/32bit,BIN

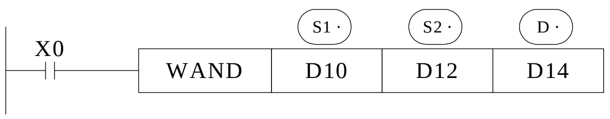
3. Suitable soft components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S1	●	●	●	●	●	●	●	●	●		
	S2	●	●	●	●	●	●	●	●	●		
	D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

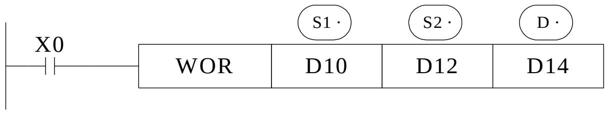
< Logic AND >



0&0=0 0&1=0

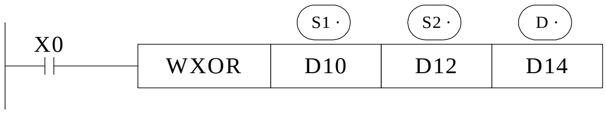
1&0=0 1&1=1

< Logic OR >



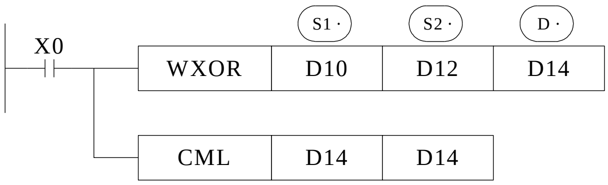
0 or 0=0 0 or 1=1
1 or 0=1 1 or 1=1

< Logic WXOR >



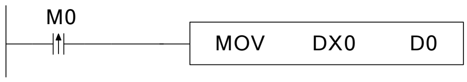
0 xor 0=0 0 xor 1=1
1 xor 0=1 1 xor 1=0

If use this instruction along with CML instruction, XOR NOT executed.



Example 1:

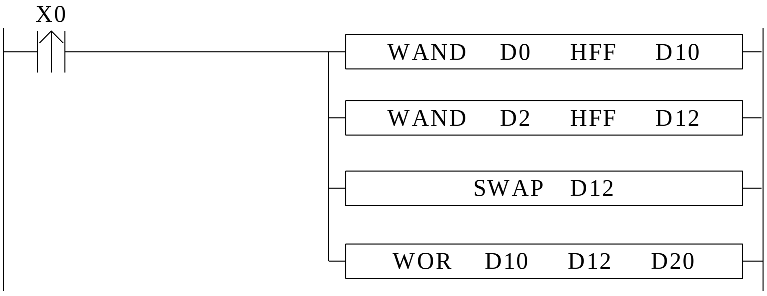
The 16 bits data is composed by X0~X7, and store in D0.



Transform the state of X0, X1, X2, X3 to 8421 code and store in D0.

Example 2:

Combine the low 8 bits of D0 and D2 to a word.




```
LDP     X0
WAND    D0      HFF     D10     //X0 rising edge
WAND    D2      HFF     D12     //Logic and, take the low 8 bits of D0 and save in D10
SWAP    D12                     // Logic and, take the low 8 bits of D2 and save in D12
WOR     D10     D12     D20     //swap the low 8 bits and high 8 bits of D12
                                //combine the low 8 bits of D10 and high 8 bits of D12,
                                and save in D20
```

4-6-8. Logic converse [CML]

1. Summary

Logic converse the data

Converse [CML]			
16 bits	CML	32 bits	DCML
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source data address	16 bits/32 bits, BIN
D	Result address	16 bits/32 bits, BIN

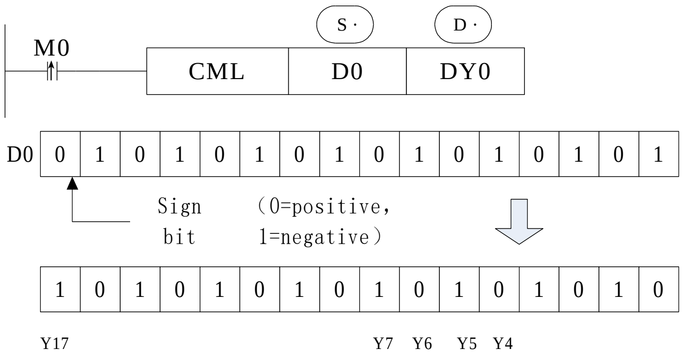
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			

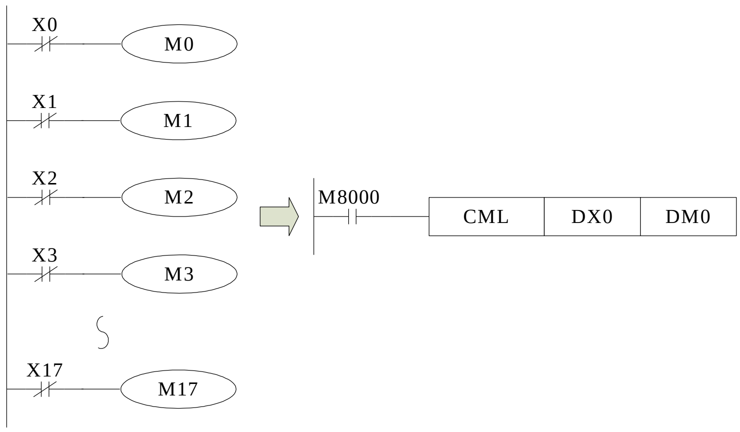
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



Each data bit in the source device is reversed (1→0, 0→1) and sent to the destination device.
If use constant K in the source device, it can be auto convert to be binary.
This instruction is fit for PLC logical converse output.

< Read the converse input >



The sequential control instruction in the left could be denoted by the following CML instruction.

4-6-9． Negative [NEG]

1. Summary

Get the negative data

Negative [NEG]			
16 bits	NEG	32 bits	DNEG
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D	The source data address	16 bits/ 32 bits, BIN

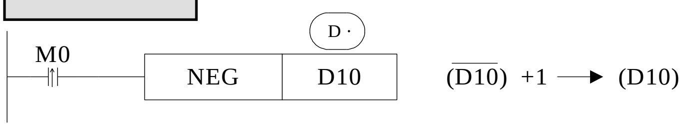
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



Converse each bit of source data (1→0, 0→1), then plus one and store the result in the source data address.

For example, the source data D10 is 20, when M0 rising edge is coming, D10 become -20.

The following two instructions are the same.

4-7. Shift Instructions

Mnemonic	Function	Chapter
SHL	Arithmetic shift left	4-7-1
SHR	Arithmetic shift right	4-7-1
LSL	Logic shift left	4-7-2
LSR	Logic shift right	4-7-2
ROL	Rotation left	4-7-3
ROR	Rotation right	4-7-3
SFTL	Bit shift left	4-7-4
SFTR	Bit shift right	4-7-5
WSFL	Word shift left	4-7-6
WSFR	Word shift right	4-7-7

4-7-1. Arithmetic shift left [SHL], Arithmetic shift right [SHR]

1. Summary

Do arithmetic shift left/right for the numbers

Arithmetic shift left [SHL]			
16 bits	SHL	32 bits	DSHL
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Arithmetic shift right [SHR]			
16 bits	SHR	32 bits	DSHR
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D	The source data address	16bit/32bit,BIN
n	Shift left or right times	16bit/32bit,BIN

3. Suitable soft components

Word

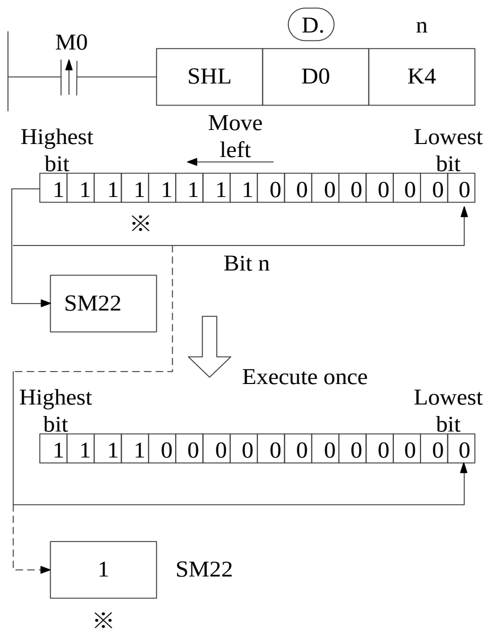
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●		●	●		●	●	●			
n									●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

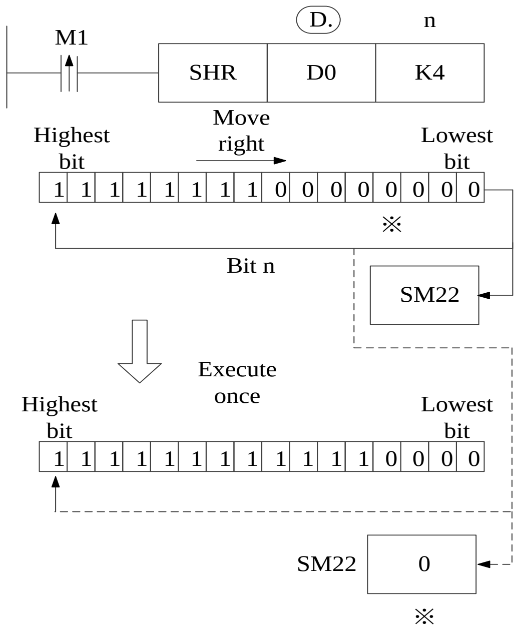
Description

After executing SHL once, the lowest bit is filled with 0, the last bit is stored in carry flag. After executing SHR once, the highest bit is the same; the last bit is stored in carry flag.

< Arithmetic shift left >



< Arithmetic shift right >



4-7-2. Logic shift left [LSL], Logic shift right [LSR]

1. Summary

Do logic shift right/left for the data

Logic shift left [LSL]			
16 bits	LSL	32 bits	DLSL
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Logic shift right [LSR]			
16 bits	LSR	32 bits	DLSR
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D	Source data address	16 bits/32 bits, BIN
n	Arithmetic shift left/right times	16 bits/32bits, BIN

3. Suitable soft components

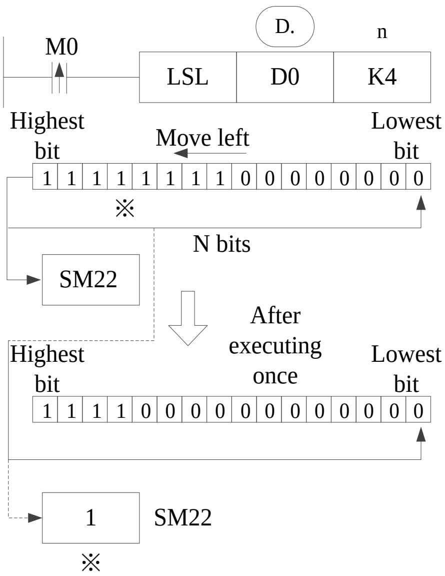
Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	D	●		●	●		●	●	●			
	n									●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

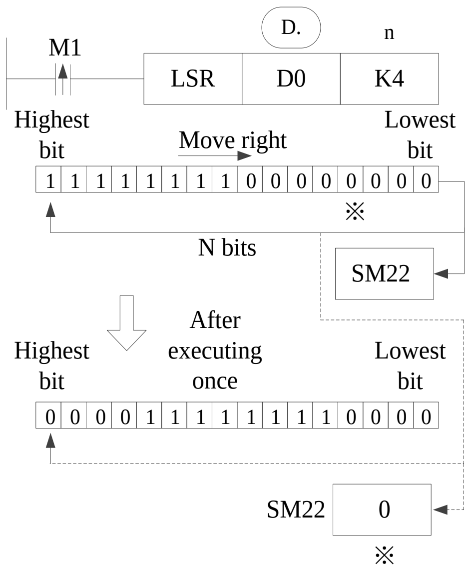
Description

After executing LSL once, the lowest bit is filled with 0; the last bit is stored in carry flag. LSL meaning and operation are the same to SHL.
After executing LSR once, the highest bit is filled with 0; the last bit is stored in carry flag. LSR and SHR are different, LSR add 0 in the highest bit when moving, SHR all bits are moved.

< Logic shift left >



< Logic shift right >



4-7-3． Rotation shift left [ROL], Rotation shift right [ROR]

1. Summary

Cycle shift left or right

Rotation shift left [ROL]			
16 bits	ROL	32 bits	DROL
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-
Rotation shift right [ROR]			
16 bits	ROR	32 bits	DROR
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
D	Source data address	16 bits/32 bits, BIN
n	Shift right or left times	16 bits/32 bits, BIN

3. Suitable soft components

Word

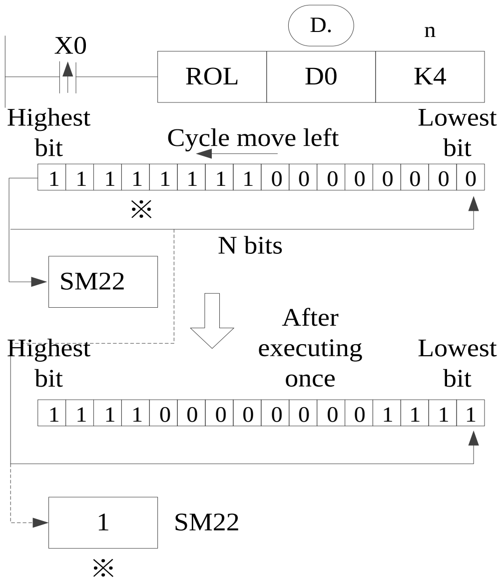
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●		●	●		●	●	●			
n									●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

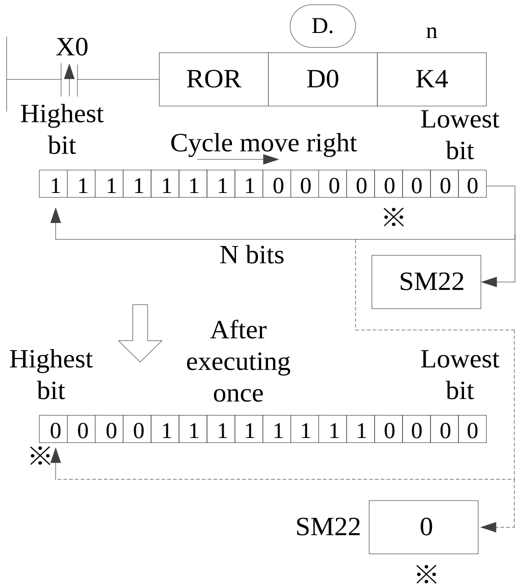
Description

When X0 changes from OFF to ON, the value will be cycle moved left or right, the last bit is stored in carry flag.

< Cycle shift left >



< Cycle shift right >



4-7-4. Bit shift left [SFTL]

1. Summary

Bit shift left

Bit shift left [SFTL]			
16 bits	SFTL	32 bits	DSFTL
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Types
S	Source soft element head address	bit
D	Target soft element head address	bit
n1	Source data quantity	16 bits /32 bits, BIN
n2	Shift left times	16 bits/32 bits, BIN

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
n1	●		●	●	●	●	●	●	●		
n2	●		●	●	●	●	●	●	●		

Bit

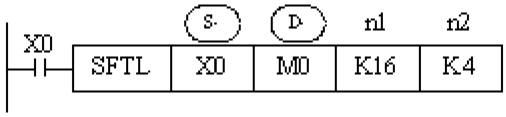
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S	●	●	●	●	●	●	
D		●	●	●	●	●	

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

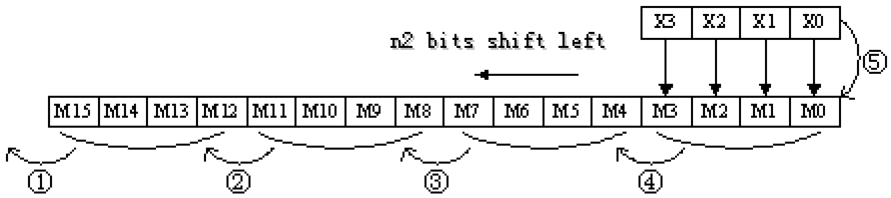
M includes M, HM, SM;   S includes S, HS;   T includes T, HT;   C includes C, HC.

Description

Move n2 bits left for the object which contains n1 bits.
When X0 changes from OFF to ON, the instruction will move n2 bits for the object.
For example, if n2 is 1, the object will move 1 bit left when the instruction executes once.



M15~M12→Overflow
M11~M 8→M15~M 12
M 7~M 4→M11~M8
M 3~M 0→M7~M4
X 3~X 0→M3~M0



4-7-5. Bit shift right [SFTR]

1. Summary

Bit shift right

Bit shift right [SFTR]			
16 bits	SFTR	32 bits	DSFTR
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element head address	bit
D	Target soft element head address	bit
n1	Source data quantity	16 bits/32 bits, BIN
n2	Shift right times	16 bits/32 bits, BIN

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
n1	●		●	●	●	●	●	●	●		
n2	●		●	●	●	●	●	●	●		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S	●	●	●	●	●	●	
D		●	●	●	●	●	

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

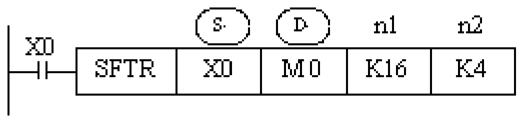
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Description

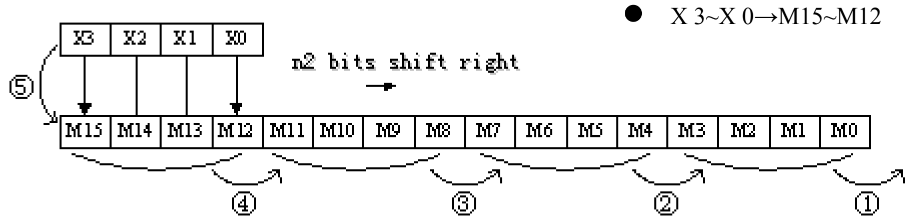
Move n2 bits right for the object which contains n1 bits.

When X0 changes from OFF to ON, the instruction will move n2 bits for the object.

For example, if n2 is 1, the object will move 1 bit right when the instruction executes once.



- M 3~M 0→Overflow
- M 7~M 4→M3~M0
- M11~M 8→M7~M4
- M15~M12→M11~M8
- X 3~X 0→M15~M12



4-7-6. Word shift left [WSFL]

1. Summary

Word shift left

Word shift left [[WSFL]			
16 bits	WSFL	32 bits	-
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element head address	16 bits, BIN
D	Target soft element head address	16 bits, BIN
n1	Source data quantity	16 bits, BIN
n2	Word shift left times	16 bits, BIN

3. Suitable soft components

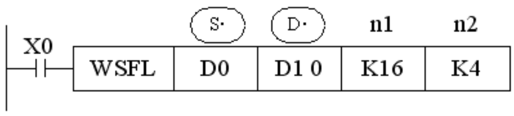
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n1	●		●	●		●	●	●	●		
n2	●		●	●		●	●	●	●		

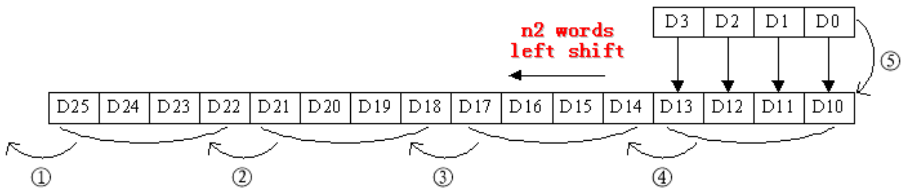
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

Move n2 words left for the object which contains n1 words.
When X0 changes from OFF to ON, the instruction will move n2 words for the object.



D25~D22→Overflow
D21~D18→D25~D22
D17~D14→D21~D18
D13~D10→D17~D14
D 3~D 0→D13~D10



4-7-7. Word shift right [WSFR]

1. Summary

Word shift right

Word shift right [WSFR]			
16 bits	WSFR	32 bits	-
Execution condition	rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element head address	16 bits, BIN
D	Target soft element head address	16 bits, BIN
n1	Source data quantity	16 bits, BIN
n2	Shift right times	16 bits, BIN

3. Suitable soft components

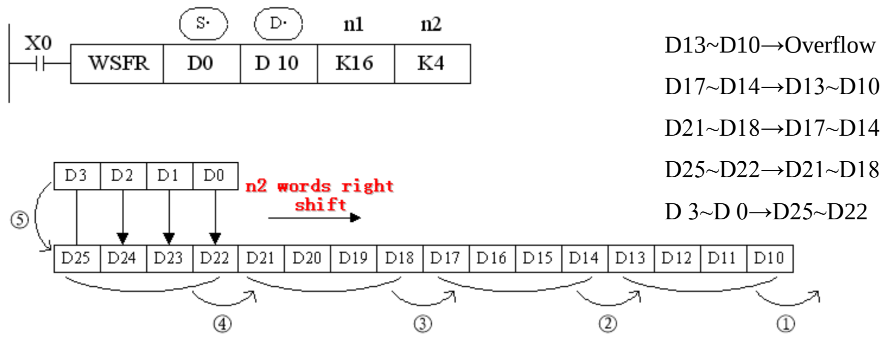
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n1	●		●	●		●	●	●	●		
n2	●		●	●		●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

Move n2 words right for the object which contains n1 words.
When X0 changes from OFF to ON, the instruction will move n2 words for the object.



4-8. Data Convert

Mnemonic	Function	Chapter
WTD	Single word integer converts to double word integer	4-8-1
FLT	16 bits integer converts to float point	4-8-2
DFLT	32 bits integer converts to float point	4-8-2
FLTD	64 bits integer converts to float point	4-8-2
INT	Float point converts to integer	4-8-3

BIN	BCD convert to binary	4-8-4
BCD	Binary converts to BCD	4-8-5
ASCI	Hex. converts to ASCII	4-8-6
HEX	ASCII converts to Hex.	4-8-7
DECO	Coding	4-8-8
ENCO	High bit coding	4-8-9
ENCOL	Low bit coding	4-8-10
GRY	Binary converts to gray code	4-8-11
GBIN	Gray code converts to binary	4-8-12

4-8-1. Single word integer converts to double word integer [WTD]

1. Summary

Single word integer converts to double word integer [WTD]			
16 bits	WTD	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element address	16 bits, BIN
D	Target soft element address	32 bits, BIN

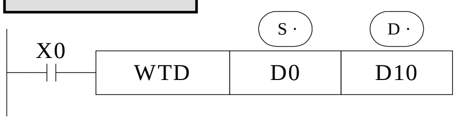
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD*	TD*	CD*	DX	DY	DM*	DSV	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			

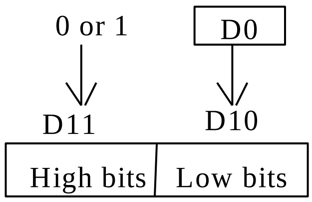
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D0) → (D11, D10)

Single Word Double Word



When single word D0 is positive integer, after executing this instruction, the high bit of double word D10 is 0.
When single word D0 is negative integer, after executing this instruction, the high bit of double word D10 is 1.
the high bit 0 and 1 is binary value.

4-8-2. 16 bits integer converts to float point [FLT]

1. Summary

16 bits integer converts to float point [FLT]					
16 bits	FLT	32 bits	DFLT	64 bits	FLTD
Execution condition	Normally ON/OFF, rising/falling edge		Suitable Models	XD, XL	
Hardware requirement	-		Software requirement	-	

2. Operands

Operands	Function	Data Type
S	Source soft element address	16 bits/32 bits/64 bits,BIN
D	Target soft element address	32 bits/64 bits,BIN

3. Suitable soft components

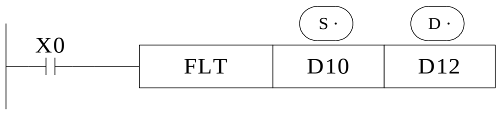
Word

Operand	System									Constant	Module	
	D*	FD	ED	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●								●		
D	●											

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

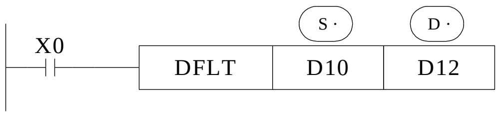
Description

<16 bits>



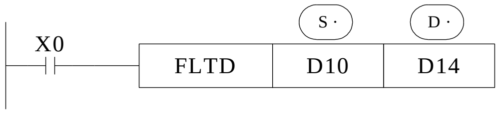
(D10) → (D13,D12)
BIN integer Binary float point

<32 bits >



(D11,D10) → (D13,D12)
BIN integer Binary float point

<64 bits>



(D13,D12,D11,D10) → (D15,D14)
BIN integer Binary float point

Convert BIN integer to binary floating point. As the constant K, H will auto convert by the floating operation instruction, so this FLT instruction can't be used.
The inverse transformation instruction is INT.
FLTD can change the 64 bits integer to 32 bits floating value.

D0 is integer 20, after executing the instruction, D10 is floating value 20.

Note: Before using floating number operation instructions such as EADD, ESUB, EMUL, EDIV, EMOV and ECMP, make sure that all operation parameters are floating number.

4-8-3. Float point converts to integer [INT]

1. Summary

Floating point converts to integer [INT]			
16 bits	INT	32 bits	DINT
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element address	16 bits/32 bits, BIN
D	Target soft element address	16 bits/32 bits, BIN

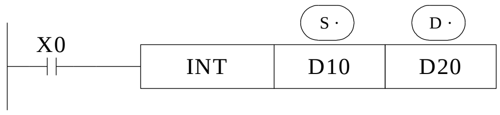
3. Suitable soft components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S	●	●									
	D	●										

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS; the word combined by bits.

Description

<16 bits>

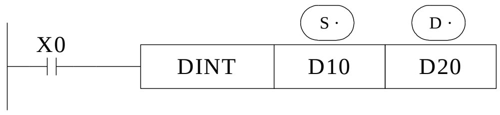


(D11,D10) → (D20)

Binary Float BIN integer

Give up the data after the decimal dot

<32 bits>



(D11,D10) → (D20,D21)

Binary Float BIN integer

Give up the data after the decimal dot

The binary source number is converted into a BIN integer and stored at the destination device.
Abandon the value behind the decimal point.
The inverse instruction is FLT.
When the result is 0, the flag bit is ON.
When converting, less than 1 and abandon it, zero flag is ON.
The result is over below data, the carry flag is ON.
16 bits operation: -32,768~32,767
32 bits operation: -2,147,483,648~2,147,483,647

For example, if D0 is floating value 130.2, after executing INT, D10 value is integer 130.

4-8-4. BCD convert to binary [BIN]

1. Summary

BCD convert to binary [BIN]			
16 bits	BIN	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element address	BCD
D	Target soft element address	16 bits/32 bits, BIN

3. Suitable soft components

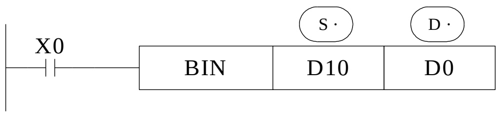
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			

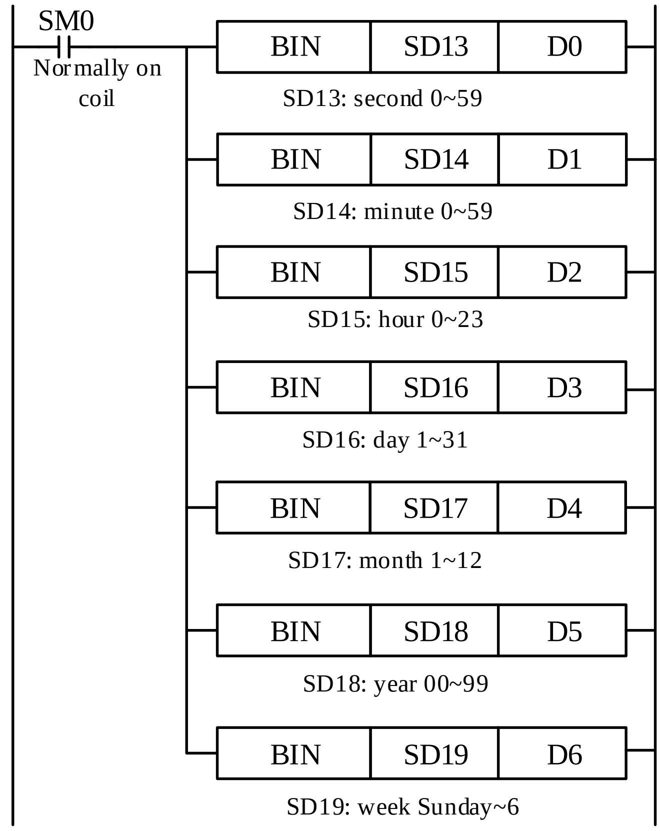
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

Source (BCD) → destination (BIN)



If source data is not BCD code, SM409 will be ON (Operation error), SD409=4 (error occurs).
As constant K automatically converts to binary, so it’s not suitable for this instruction.
For example: all the information stored in the clock information register SD13~SD19 of PLC is BCD code, but we are used to using decimal value. The time information can be converted from BCD code information to binary:



4-8-5. Binary convert to BCD [BCD]

1. Summary

Convert binary data to BCD code

Binary convert to BCD [BCD]			
16 bits	BCD	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element address	16 bits, BIN
D	Target soft element address	BCD code

3. Suitable soft components

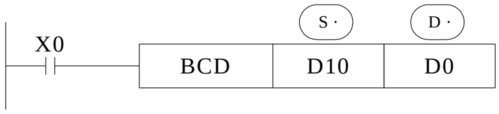
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

source (BIN)→destination (BCD)



This instruction can change the binary value to BCD code.

4-8-6. Hex converts to ASCII [ASCI]

1. Summary

Hex. convert to ASCII [ASCI]			
16 bits	ASCI	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Source soft element address	2 bits, HEX
D	Target soft element address	ASCII code
n	Transform character quantity	16 bits, BIN

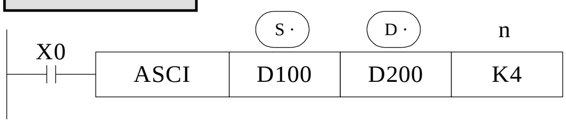
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n	●		●	●		●	●	●	●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



Transform the source Hex data to ASCII code, and store in D·. The transformation chacters are n.

D· Will store one ASCII code.

The convert process is this

Assign start device:
(D100)=0ABCH
(D101)=1234H
(D102)=5678H

[0]=30H [1]=31H
[5]=35H [A]=41H
[2]=32H [6]=36H
[B]=42H [3]=33H
[7]=37H [C]=43H
[4]=34H [8]=38H

n / D	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 down	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200 up		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201 down			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201 up				[C]	[B]	[A]	[0]	[4]	[3]
D202 down					[C]	[B]	[A]	[0]	[4]
D202 up						[C]	[B]	[A]	[0]
D203 down							[C]	[B]	[A]
D203 up								[C]	[B]
D204 down									[C]

4-8-7. ASCII convert to Hex.[HEX]

1. Summary

ASCII converts to Hex. [HEX]			
16 bits	HEX	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Date type
S	Source soft element address	ASCII
D	Target soft element address	2 bits, HEX
n	ASCII Character quantity	16 bits, BIN

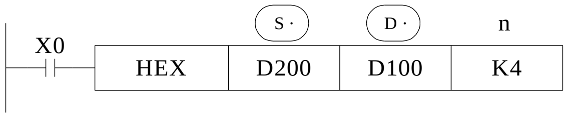
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n									●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



Convert the high 8 bits and low 8 bits in source (S·) to HEX data. Move 4 bits every time to destination (D·). The convert character number is assigned by n.

The convert process is the following:

(S·)	ASCII Code	HEX Convert
D200 down	30H	0
D200 up	41H	A
D201 down	42H	B
D201 up	43H	C
D202 down	31H	1
D202 up	32H	2
D203 down	33H	3
D203 up	34H	4
D204 down	35H	5

n (D·)	D102	D101	D100
1	Not change to be 0		··0H
2			·0AH
3			0ABH
4			0ABC H
5		··0H	ABC1 H
6		·0AH	BC12H
7		0ABH	C123H
8		0ABC H	1234H
9	··0H	ABC1H	2345H

n=k4

D200	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0
	41H? [A]								30H? [0]							
D201	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0
	43H? [C]								42H? [B]							
D100	0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
	0				A				B				C			

4-8-8. Coding [DECO]

Summary

Change any data or bit to 1.

Coding [DECO]			
16 bits	DECO	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	The source data address	16 bits, BIN
D	The decode result head address	16 bits, BIN
n	The decoding soft element bit quantity	16 bits, BIN

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
n									●		

Bit

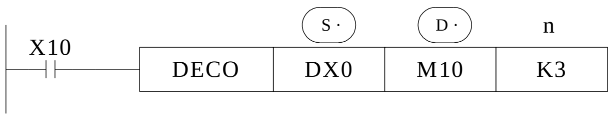
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D	●	●	●	●	●	●	

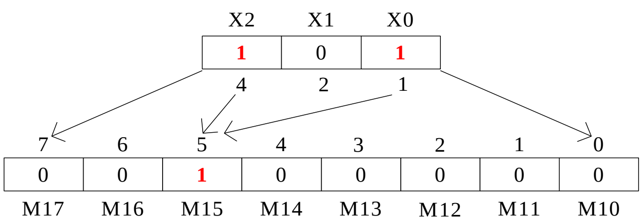
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T and HT; C includes C and HC.

Description

< When D· is bit unit > n≤16





N = 3, so the decoding object is the lower three bits in DX0, which are X2 ~ X0.

N = 3, so the decoding results need to be expressed by 2^3 = 8 bits, which are M17 ~ M10.

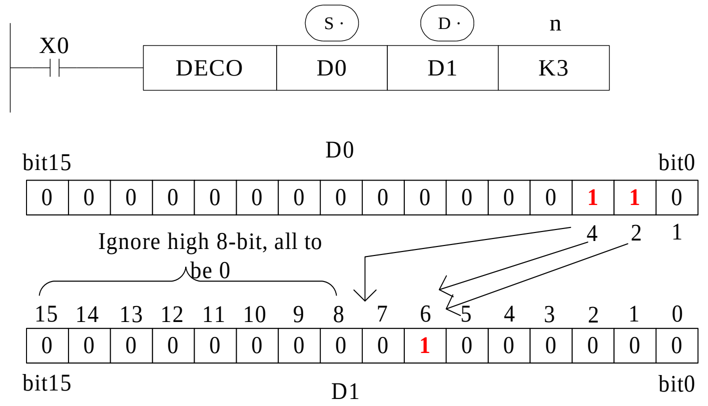
When X2 = 1, X1 = 0, X0 = 1, the value it represents is 4 + 1 = 5, so M15 in the fifth place from M10 changes to 1; when X2 ~ X0 is all zero, the value is 0, so M10 is 1 (M10 is the 0th place).

If n = 0, the instruction will not be executed. If n is the value out of 0 ~ 16, the instruction will not be executed.

When n = 16, if the decoding command D· is a bit soft component, the number of points is 2 ^ 16 = 65536.

When the driver input is OFF, the instruction is not executed, and the decoding output of the action is maintained.

< When D· is word device > n≤4



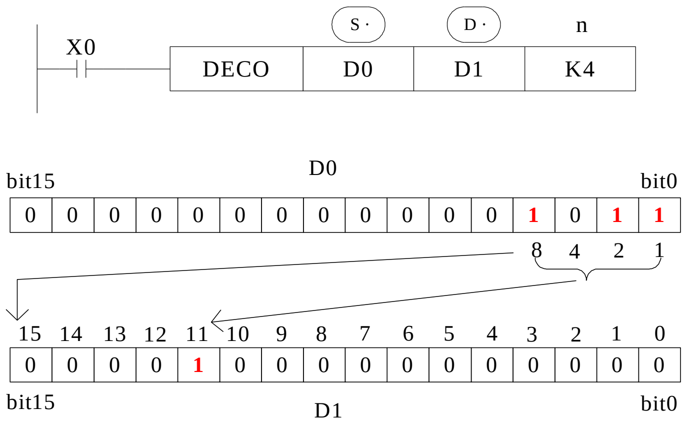
The low n-bit (n ≤ 4) of the source address is decoded to the target address. When n ≤ 3, the high 8-bit of the target turns to 0.

If n = 0, the instruction will not be executed. If n is out of 0 ~ 4, the instruction will not be executed.

N = 3, so the decoding object in D0 is bit2-bit0, and the maximum value it represents is 4 + 2 + 1 = 7.

N = 3, so in D1, 2^3 = 8 bits are needed to represent the decoding result, that is, bit7 ~ bit0. When bit2 and bit1 are both 1 and bit0 are 0, the value is 4+2=6, so bit6 in D1 is ON.

< D· is word soft component > n≤4



The low n-bit (n ≤4) of the source address is decoded to the target address. When n ≤ 3, the high 8-bit of the target turns to 0.

If n = 0, the instruction will not be executed. If n is out of 0 ~ 4, the instruction will not be executed.

N = 4, so the object of decoding in D0 is bit3 ~ bit0, which represents the maximum value of 8 + 4 + 2 + 1 = 15.

N = 4, so in D1, 2^4 = 16 bits are needed to represent the decoding result, that is, bit15 ~ bit0. When bit3, bit1 and bit0 are all 1 and bit2 is 0, the numerical value is 8+2+1=11, so bit11 in D1 is ON.

4-8-9. High bit coding [ENCO]

1. Summary

Find the highest bit which is 1.

High bit coding [ENCO]			
16 bits	ENCO	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Coding data address	16 bits, BIN
D	Coding result address	16 bits, BIN
n	The bit quantity of coding result	16 bits, BIN

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n									●		

Bit

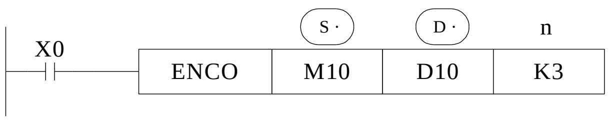
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S	●	●	●	●	●	●	

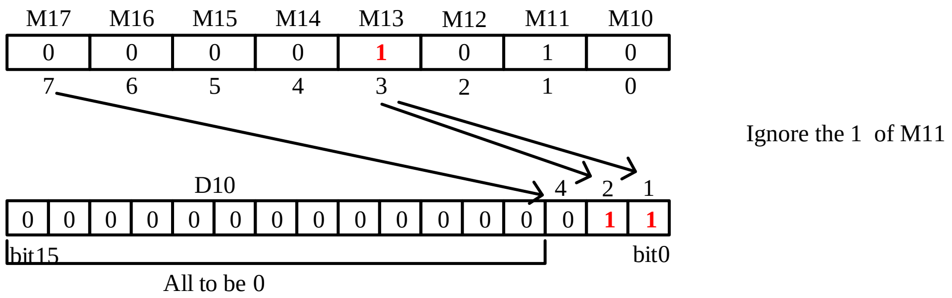
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T and HT; C includes C and HC.

Description

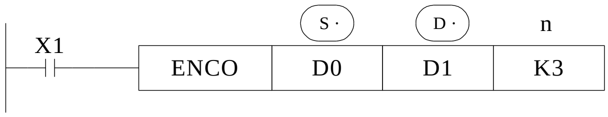
< When S· is bit device > n≤16

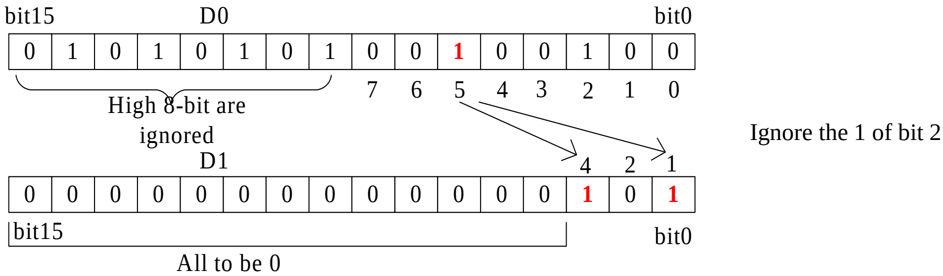




If the number of bits in the source address is 1, the low side is ignored, and if the source address is 0, the instruction will not be executed.
When the driving condition is OFF, the instruction is not executed and the coding output is unchanged.
When n = 16, if the encoding instruction is a bit element, its point number is 2 ^ 16 = 65536.
N = 3, the encoded object has 2^3 = 8 bits, which are M17 ~ M10, and the encoding results are stored in the lower three bits of D10, which are bit2 ~ bit0.
M13 and M11 are both 1. Ignoring M11, M13 is coded, bit2-bit0 represent 3, while bit0 and bit1 are 1.

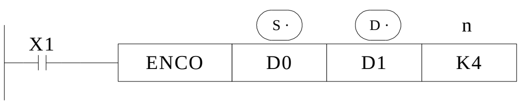
< When S· is word device n≤4

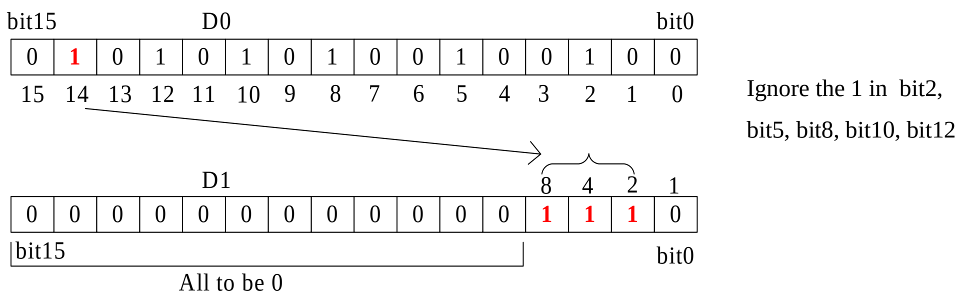




If multiple bits in the source address is 1, the low side is ignored, and if the source address is 0, the instruction will not be executed.
When the driver input is OFF, the instruction is not executed and the coding output is unchanged.
When n ≤ 3, the high 8 bits in D0 are neglected.
When n=3, the encoding object has 2^3 = 8 bits, that is, bit7 ~ bit0 in D0. The encoding result is stored in the lower 3 bits in D1, that is, bit2 ~ bit0.
When bit5 and bit2 in D0 are both 1, bit2 is ignored, and bit5 is coded, bit2-bit0 represent 5, bit2 and bit0 are 1.

< S· is word soft component > n≤4





If the number of bits in the source address is 1, the low side is ignored, and if the source address is 0, the instruction will not be executed.

When the driver input is OFF, the instruction is not executed and the coding output is unchanged.

N = 4, the encoded object has 2^4 = 16 bits, that is, bit15 ~ bit0 in D0. The encoding result is stored in the lower 4 bits in D1, that is, bit3 ~ bit0.

The highest bit of 1 in D0 is bit14, ignoring all low bits 1, and encoding bit14, bit3-bit0 represent 14, bit3, bit2 and bit1 are 1.

4-8-10. Low bit coding [ENCOL]

1. Summary

Find the position where the low bit is ON.

Low bit coding [ENCOL]			
16 bits	ENCOL	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need coding	16bit,BIN
D	Soft element address to save coding result	16bit,BIN
n	The bit quantity of coding result	16bit,BIN

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●			
D	●		●	●		●	●	●			
n									●		

Bit

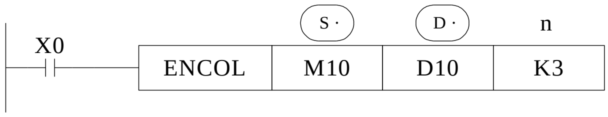
Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S	●	●	●	●	●	●	

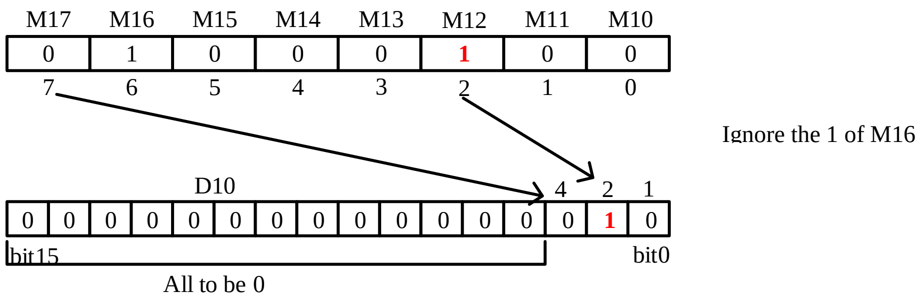
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM;   S includes S, HS;   T includes T and HT;   C includes C and HC.

Description

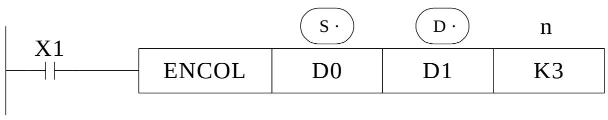
<if (S·) is bit device > n≤16

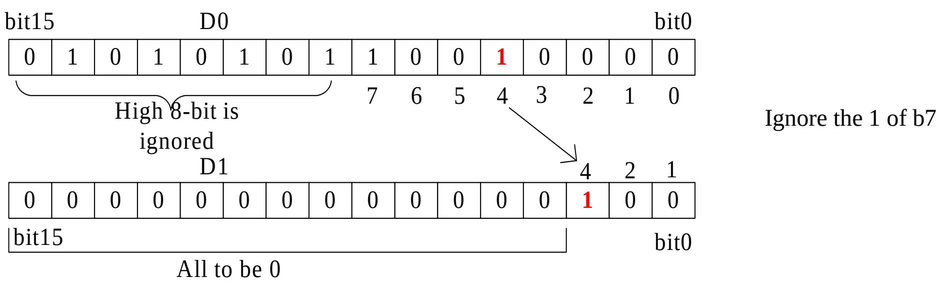




If the number of bits in the source address is 1, the high bit side is ignored, and if the source address is 0, the instruction will not be executed.
When the driving condition is OFF, the instruction is not executed and the coding output is unchanged.
When n = 16, if the (S·) of encoding instruction is a bit element, its point is 2 ^ 16 = 65536.
N = 3, the encoded object has 2^3 = 8 bits, which are M17 ~ M10, and the encoding results are stored in the lower three bits of D10, which are bit2 ~ bit0.
M12 and M16 are both 1. Ignoring M16, M12 is coded, bit2-bit0 represent 2, while bit1 is 1.

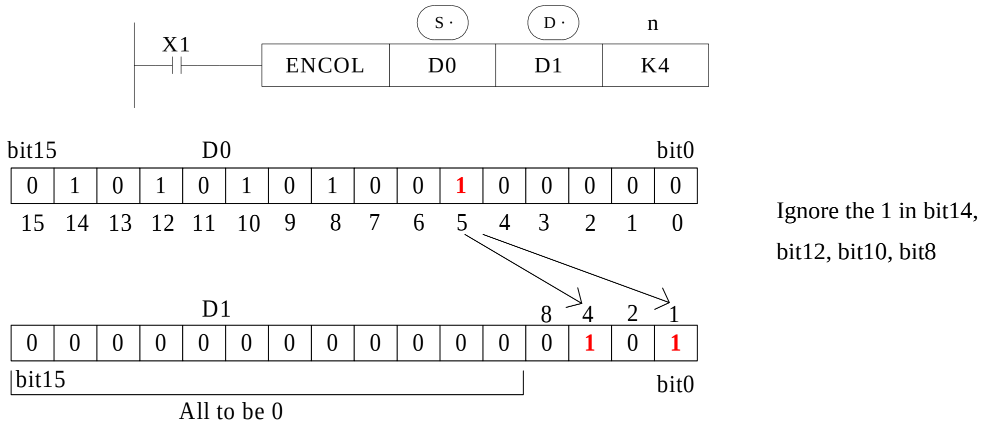
< if (S·) is word device> n≤4





If multiple bits in the source address is 1, the high bit side is ignored, and if the source address is 0, the instruction will not be executed.
When the driver input is OFF, the instruction is not executed and the coding output is unchanged.
When n ≤3, the high 8 bits in D0 are neglected.
The encoding object has 2^3 = 8 bits, that is, bit7 ~ bit0 in D0. The encoding result is stored in the lower 3 bits in D1, that is, bit2 ~ bit0.
When bit7 and bit4 in D0 are both 1, bit7 is ignored and bit4 is coded. Bit 2 is 1 when bit2-bit0 is expressed as 4.

< S· is word soft component > n≤4



If multiple bits in the source address is 1, the high bit side is ignored, and if the source address is 0, the instruction will not be executed.
When the driver input is OFF, the instruction is not executed and the coding output is unchanged.
N = 4, the encoded object has 2^4 = 16 bits, that is, bit15 ~ bit0 in D0. The encoding result is stored in the lower 4 bits in D1, that is, bit3 ~ bit0.
The lowest bit of 1 in D0 is bit5, ignoring all high bits 1, and encoding bit5 with bit3-bit0 as 5, bit2 and bit0 as 1.

4-8-11. Binary to Gray code [GRY]

1. Summary

Transform the binary data to gray code.

Binary to gray [GRY]			
16 bits	GRY	32 bits	DGRY
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need coding	16bits/32bits, BIN
D	Soft element address to save coding result	16bits/32bits, BIN

3. Suitable soft components

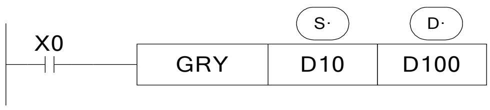
Word

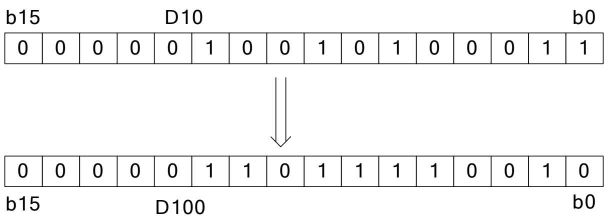
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

Source (BIN) → target (GRY)





Each bit of D10 will XOR with the bit on its left side. As the related gray code, the left bit will not change (the left bit is 0); the transformation result is stored in D100.

Transform the binary value to gray code.
GRY has 32 bits mode DGRY, which can transform 32 bits gray code.
(S·) Range is 0~32,767 (16 bits instruction); 0~2,147,483,647 (32 bits instruction).

4-8-12. Gray code to binary [GBIN]

1. Summary
Transform the gray code to binary data.

Gray code to binary [GBIN]			
16 bits	GBIN	32 bits	DGBIN
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need coding	16bits/32bits, BIN
D	Soft element address to save coding result	16bits/32bits, BIN

3. Suitable soft components

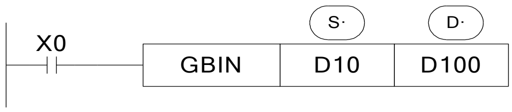
Word

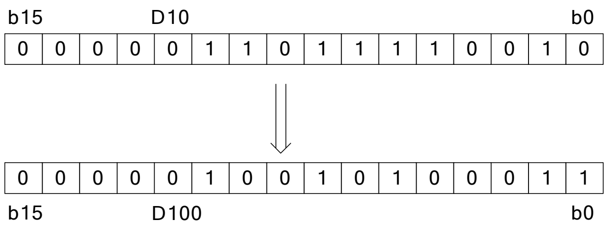
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●	●	●	●	●	●	●	●		
D	●		●	●		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

Source (GRY) → target (BIN)





From the left second bit of D10, XOR each bit with the value after decoding, as the bit value after decoding (the left bit will not change). The transformation value will be stored in D100.

Transform the gray code to binary value.
GBIN has 32 bits mode DBIN, which can transform 32 bits binary value.
S· Range is 0~32,767 (16 bits instruction); 0~2,147,483,647 (32 bits instruction).

4-9. Floating number Operation

Mnemonic	Function	Chapter
ECMP	Floating Compare	4-9-1
EZCP	Floating Zone Compare	4-9-2
EADD	Floating Add	4-9-3
ESUB	Floating Subtract	4-9-4
EMUL	Floating Multiplication	4-9-5

EDIV	Floating Division	4-9-6
ESQR	Floating Square Root	4-9-7
SIN	Sine	4-9-8
COS	Cosine	4-9-9
TAN	Tangent	4-9-10
ASIN	ASIN	4-9-11
ACOS	ACOS	4-9-12
ATAN	ATAN	4-9-13

4-9-1． Floating Compare [ECMP]

1. Summary

Floating Compare [ECMP]			
16 bits	-	32 bits	ECMP
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Soft element address need compare	32 bits, BIN
S2	Soft element address need compare	32 bits, BIN
D	Compare result	bit

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		

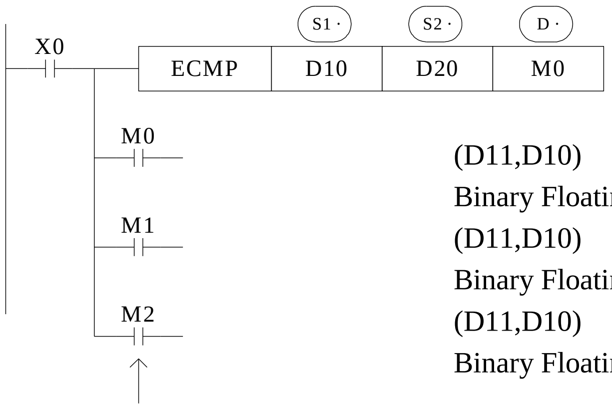
Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S and HS; C includes C and HC.

Description

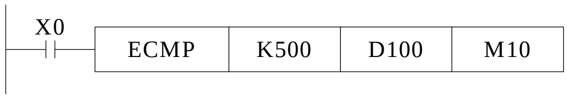
(D11, D10) : (D21, D20) → M0,M1,M2
Binary Floating Binary Floating



(D11,D10) > (D21<D20)
Binary Floating Binary Floating
(D11,D10) = (D21<D20)
Binary Floating Binary Floating
(D11,D10) < (D21<D20)
Binary Floating Binary Floating

When X0 is OFF, even ECMP doesn't run, M0~M2 will keep the status before X0 is OFF.

The instruction will compare the two source data S1 and S2. The result is stored in three bits from D.
If a constant K or H used as source data, the value is converted to floating value.



(K500) : (D101, D100) →M10,M11,M12
Binary converts to floating Binary floating

Note: Before the instruction is executed, the comparison data must be all floating numbers (if it is an integer, it can be converted by FLT instructions); otherwise, the execution result will be wrong.

4-9-2. Floating Zone Compare [EZCP]

1. Summary

Floating Zone Compare [EZCP]			
16 bits	-	32 bits	EZCP
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Soft element address need compare	32 bits, BIN
S2	Upper limit of compare data	32 bits, BIN
S3	Lower limit of compare data	32 bits, BIN

D	The compare result soft element address	bit

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		
S3	●	●			●	●	●	●	●		

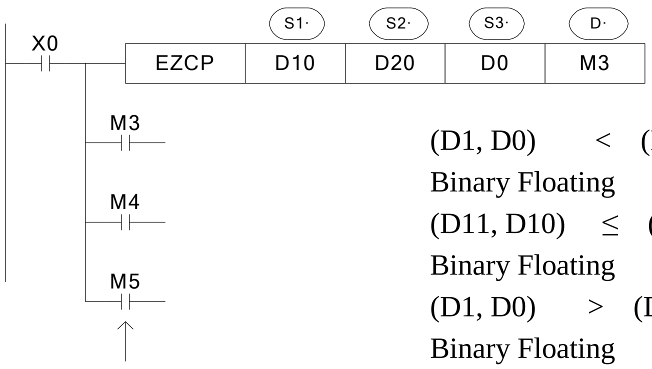
Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S and HS; C includes C and HC.

Description

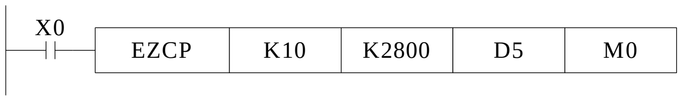
Compare the source data with the range



(D1, D0) < (D11, D10) ON
Binary Floating Binary Floating
(D11, D10) ≤ (D1, D0) ≤ (D21, D20) ON
Binary Floating Binary Floating Binary Floating
(D1, D0) > (D21, D20) ON
Binary Floating Binary Floating

When X0 is OFF, even EZCP doesn't run, M3~M5 will keep the status before X0 is OFF.

Compare the source data S3 to the upper and lower limit value of the range S1~S2.
The result will store in three coils starting from D.
Constant K and H will transform to binary floating value when they are source data.



(K10) : [D6,D5] : (K2800) → M0, M1, M2
Binary converts to Floating Binary Floating Binary converts to Floating

Please set S1≤ S2, when S2< S1, make S2 as the same value to S1.

Note: the compare value must be floating numbers, otherwise the result will be error.

4-9-3. Floating Addition [EADD]

1. Summary

Floating Add [EADD]			
16 bits	-	32 bits	EADD
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Addition operation data address	32 bits, BIN
S2	Addition operation data address	32 bits, BIN
D	Result address	32 bits, BIN

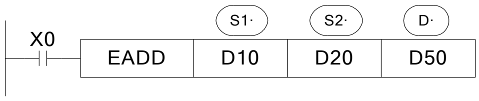
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		
D	●					●	●	●			

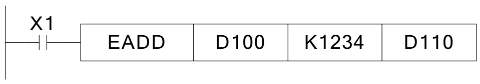
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D11, D10) + (D21, D20) → (D51, D50)

The two binary floating source data do addition operation, the result will be stored in target address.
If a constant K or H used as source data, the value is converted to floating point before the addition operation.



(K1234) + (D101, D100) → (D111, D110)
Binary converts to Floating Binary Floating Binary Floating

The source data and result address can be the same. Please note that when X0 is ON, the instruction will be executed in every scanning period.

Note: the add value must be floating numbers, otherwise the result will be error.

4-9-4. Floating Subtraction [ESUB]

1. Summary

Floating Sub [ESUB]			
16 bits	-	32 bits	ESUB
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Subtraction operation data address	32 bits, BIN
S2	Subtraction operation data address	32 bits, BIN
D	Result address	32 bits, BIN

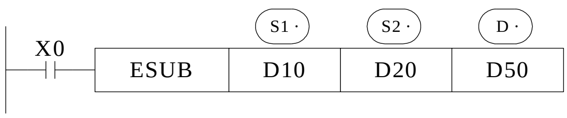
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		
D	●					●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D11, D10) − (D21,D20) → (D51,D50)
Binary Floating Binary Floating Binary Floating

The binary floating value S1 subtract S2, the result is stored in the target address.
If a constant K or H used as source data, the value is converted to floating point before the subtraction operation.

X1	ESUB	K1234	D100	D110

(K1234) － (D101, D100) → (D111, D110)
Binary converts to Floating   Binary Floating   Binary Floating

The source data and result address can be the same. Please note that when X0 is ON, the instruction will be executed in every scanning period.
Note: the operand value must be floating numbers, otherwise the result will be error.

4-9-5. Floating Multiplication [EMUL]

1. Summary

Floating Multiply [EMUL]			
16 bits	-	32 bits	EMUL
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Multiplication operation data address	32 bits, BIN
S2	Multiplication operation data address	32 bits, BIN
D	Result address	32 bits, BIN

3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		
D	●					●	●	●			

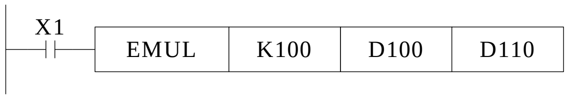
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description

X0	EMUL	D10 (S1·)	D20 (S2·)	D50 (D·)

(D11, D10) × (D21, D20) → (D51, D50)
Binary Floating   Binary Floating   Binary Floating

The floating value of S1 is multiplied with the floating value point value of S2. The result of the multiplication is stored at D as a floating value.
If a constant K or H used as source data, the value is converted to floating point before the multiplication operation.



(K100) × (D101, D100) → (D111, D110)
Binary converts to Floating Binary Floating Binary Floating

Note: the operand value must be floating numbers, otherwise the result will be error.

4-9-6. Floating Division [EDIV]

1. Summary

Floating Divide [EDIV]			
16 bits	-	32 bits	EDIV
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S1	Division operation data address	32 bits, BIN
S2	Division operation data address	32 bits, BIN
D	Result address	32 bits, BIN

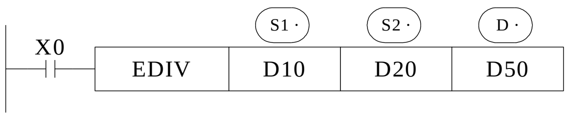
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		
D	●					●	●	●			

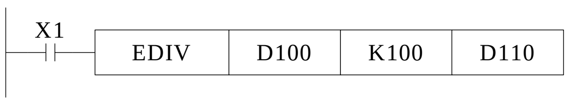
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D11, D10) ÷ (D21, D20) → (D51, D50)
Binary Floating Binary Floating Binary Floating

The floating point value of S1 is divided by the floating point value of S2. The result of the division is stored in D as a floating point value.
If a constant K or H used as source data, the value is converted to floating point before the division operation.



(D101, D100) ÷ (K100) → (D111, D110)
Binary converts to Floating Binary Floating Binary Floating

The source data S2 is 0, the calculation will be error. The instruction will not work.
Note: the operand value must be floating numbers, otherwise the result will be error.

4-9-7. Float Square Root [ESQR]

1. Summary

Floating Square Root [ESQR]			
16 bits	-	32 bits	ESQR
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

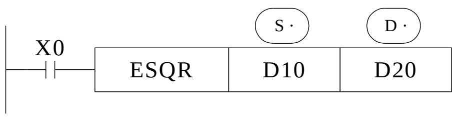
Operands	Function	Data Type
S	The soft element address need to do square root	32 bits, BIN
D	The result address	32 bits, BIN

3. Suitable soft components

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S	●	●			●	●	●	●	●		
	D	●					●	●	●			

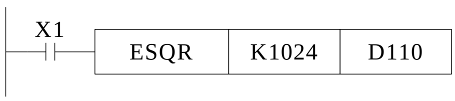
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D11, D10) → (D21, D20)
Binary Floating Binary Floating

A square root is performed on the floating point value S; the result is stored in D
If a constant K or H used as source data, the value is converted to floating point before the operation.



(K1024) → (D111, D110)
Binary converts to Floating Binary Floating

When the result is zero, zero flag activates.
Only when the source data is positive will the operation be effective. If S is negative then an error occurs and error flag SM409 is set ON, SD409=7, the instruction can't be executed.
Note: the operand value must be floating numbers, otherwise the result will be error.

4-9-8. Sine [SIN]

1. Summary

Floating Sine[SIN]			
16 bits	-	32 bits	SIN
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	The soft element address need to do sine	32 bits, BIN
D	The result address	32 bits, BIN

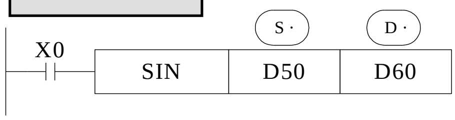
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

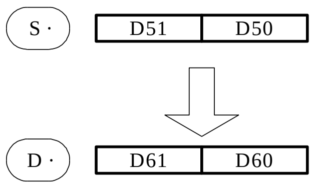
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D51, D50) → (D61, D60) SIN
Binary Floating   Binary Floating

This instruction performs the mathematical SIN operation on the floating point value in S (angle RAD). The result is stored in D.



RAD value (angle × π/180)
Assign the binary floating value

SIN value
Binary Floating

Note: the operand value must be floating numbers, otherwise the result will be error.

4-9-9. Cosine [COS]

1. Summary

Floating Cosine [COS]			
16 bits	-	32 bits	COS
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need to do cos	32 bits, BIN
D	Result address	32 bits, BIN

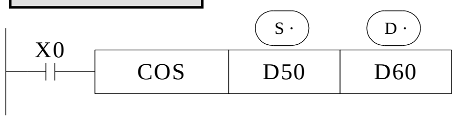
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

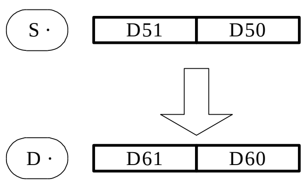
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D51,D50) RAD → (D61,D60) COS
Binary Floating Binary Floating

This instruction performs the mathematical COS operation on the floating point value in S (angle RAD). The result is stored in D.



RAD value (angle × π/180)
Assign the binary floating value
COS value
Binary Floating

Note: Before the instruction is executed, the data in parameter S must be floating number; otherwise, the execution result will be wrong.

4-9-10. TAN [TAN]

1. Summary

TAN [TAN]			
16 bits	-	32 bits	TAN
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement	-	Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need to do tan	32bit,BIN
D	Result address	32bit,BIN

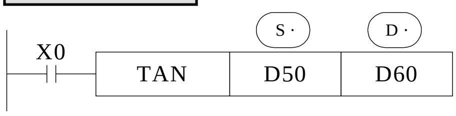
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

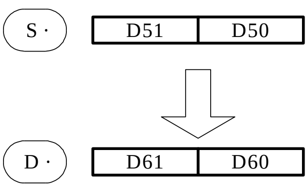
Description



(D51,D50) RAD → (D61,D60) TAN

Binary Floating Binary Floating

This instruction performs the mathematical TAN operation on the floating point value in S. The result is stored in D.



RAD value (angle × π/180)

Assign the binary floating value

TAN value

Binary Floating

Note: Before the instruction is executed, the data in parameter S must be floating number; otherwise, the execution result will be wrong.

4-9-11． ASIN [ASIN]

1. Summary

ASIN [ASIN]			
16 bits	-	32 bits	ASIN
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement		Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need to do arcsin	32 bits, BIN
D	Result address	32 bits, BIN

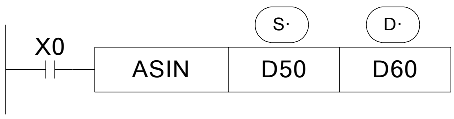
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

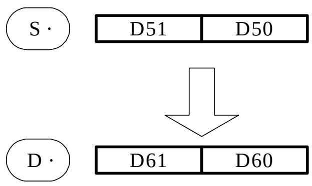
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D51, D50) ASIN → (D61, D60) RAD
Binary Floating Binary Floating

This instruction performs the mathematical ASIN operation on the floating point value in S. The result is stored in D.



ASIN value
Binary Floating
RAD value (angle×π/180)
Assign the binary floating value

Note: Before the instruction is executed, the data in parameter S must be floating number; otherwise, the execution result will be wrong.

4-9-12． ACOS [ACOS]

1. Summary

ACOS [ACOS]			
16 bits	-	32 bits	ACOS
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement		Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need to do arccos	32 bits, BIN
D	Result address	32 bits, BIN

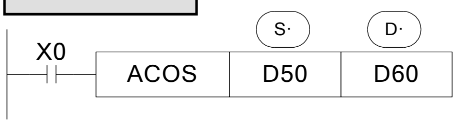
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

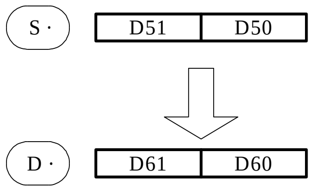
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D51,D50) ACOS → (D61,D60) RAD
Binary Floating Binary Floating

Calculate the arcos value(radian), save the result in the target address



ACOS value
Binary Floating
RAD value (angle×π/180)
Assign the binary floating value

Note: Before the instruction is executed, the data in parameter S must be floating number; otherwise, the execution result will be wrong.

4-9-13. ATAN [ATAN]

1. Summary

ATAN [ATAN]			
16 bits	-	32 bits	ACOS
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement		Software requirement	-

2. Operands

Operands	Function	Data Type
S	Soft element address need to do arctan	32 bit, BIN
D	Result address	32 bit, BIN

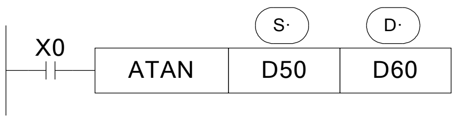
3. Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S	●	●			●	●	●	●	●		
D	●					●	●	●			

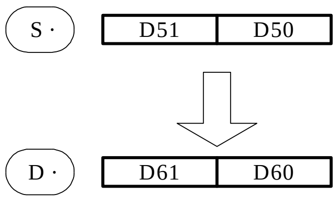
*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



(D51,D50) ATAN → (D61,D60) RAD
Binary Floating Binary Floating

Calculate the arctan value (radian), save the result in the target address



ATAN value
Binary Floating
RAD value (angle×π/180)
Assign the binary floating value

Note: Before the instruction is executed, the data in parameter S must be floating number; otherwise, the execution result will be wrong.

4-10. RTC Instructions

Mnemonic	Function	Chapter
TRD	Clock data read	4-10-1
TWR	Clock data write	4-10-2
TCMP	Clock compare	4-10-3

※1: To use the instructions, The Model should be equipped with RTC function;
※2: There are some errors in the clock of XD/XL series PLC, which is about ±5 minutes per month. It can be calibrated regularly by HMI or in the PLC program.

4-10-1. Read the clock data [TRD]

1. Instruction Summary

Read the clock data:

Read the clock data: [TRD]			
16 bits	TRD	32 bits	-
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement		Software requirement	-

2. Operands

Operands	Function	Data Type
D	Register address to save clock data	16 bits, BIN

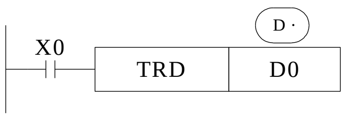
3. Suitable Soft Components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●		●	●							

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



The current time and date of the real time clock are read and stored in the 7 data devices specified by the head address D.

Read PLC's real time clock according to the following format.

Read the special data register (SD013~SD019).

Special data register for real time clock t

Unit	Item	Clock data		Unit	Item
SD018	Year	0-99	→	D0	Year
SD017	Month	1-12	→	D1	Month
SD016	Date	1-31	→	D2	Date
SD015	Hour	0-23	→	D3	Hour
SD014	Minute	0-59	→	D4	Minute
SD013	Second	0-59	→	D5	Second
SD019	Week	0 (Sun.)-6 (Sat.)	→	D6	Week

The RTC (real time clock) value is in BCD code format (SD013 to SD019). Please choose hex format to monitor the RTC value in XDPpro software. The value can be transformed to decimal format by BIN instruction. After reading the RTC by TRD instruction, the value will show in decimal format.
After reading the RTC by TRD, the value becomes decimal value.
after executing TRD instruction, D0 to D6 are occupied.

4-10-2. Write Clock Data [TWR]

1. Instruction Summary
Write the clock data:

Write clock data [TWR]			
16 bits	-	32 bits	TWR
Execution condition	Normally ON/OFF, rising/falling edge	Suitable Models	XD, XL
Hardware requirement		Software requirement	-

2. Operands

Operands	Function	Data Type
S	Write the clock data to the register	16 bits, BIN

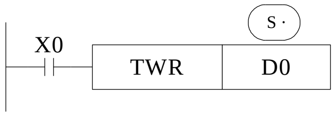
3. Suitable Soft Components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D	●		●	●	●	●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



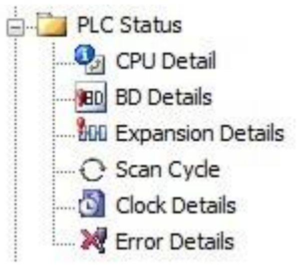
Write the RTC value to the PLC.
Write the set clock data into PLC’s real time clock.
In order to write real time clock, please set the 7 registers value from D0 to D6.

Data for clock setting

Unit	Item	Clock data		Unit	Item
D0	Year	0-99	→	SD018	Year
D1	Month	1-12	→	SD017	Month
D2	Date	1-31	→	SD016	Date
D3	Hour	0-23	→	SD015	Hour
D4	Minute	0-59	→	SD014	Minute
D5	Second	0-59	→	SD013	Second
D6	Week	0 (Sun.)-6 (Sat.)	→	SD019	Week

Special data register for real time clock t

After executing TWR instruction, the time in real time clock will immediately change to be the new time. It is a good idea to set the time few minutes late as the current time, and then drive the instruction when the real time reaches this value.
Note: when choosing secret download program advance mode in XDPpro software, the RTC only can be changed through TWR instruction.
There is another method to write the RTC. In the XDPpro software, please click the clock details in project bar on the left. Then click write into the current time.the PC will auto-write the current time to the PLC.



4-10-3. Clock compare [TCMP]

1. Instruction Summary
Compare three continuous clocks time.

Clock compare [TCMP]			
16 bits	TCMP	32 bits	-
Condition	Normally ON/OFF, rising/falling edge	Suitable model	XD, XL
Hardware	-	Software	-

2. operand

Operand	Function	Model
S1	The first clock soft component address	16 bits, BIN
S2	The second clock soft component address	16 bits, BIN
S3	The third clock soft component address	16 bits, BIN
S4	PLC real time clock information first address	16 bits, BIN
D2	The compare result first address	bit

3. suitable soft component

Word

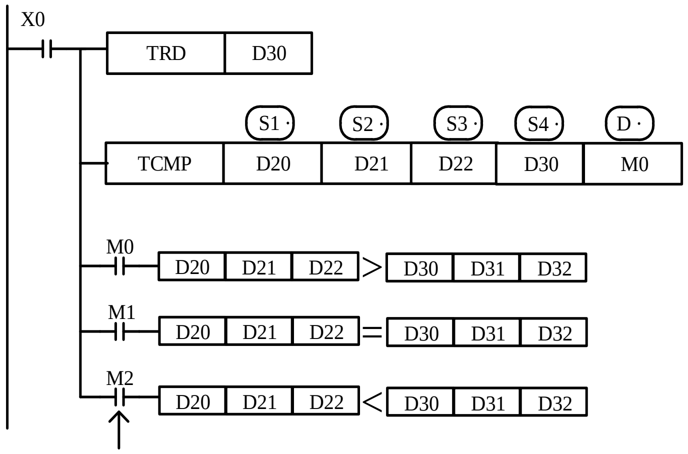
Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●			●	●	●	●	●		
S2	●	●			●	●	●	●	●		
S3	●	●			●	●	●	●	●		
S4	●	●			●	●	●	●	●		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●	●	●			

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Description

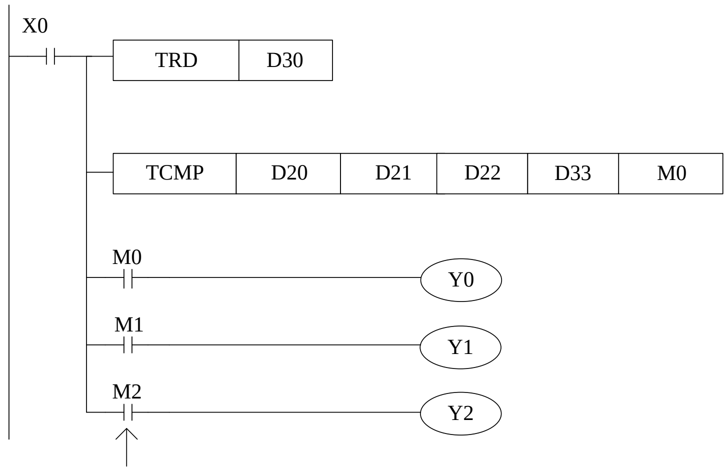


Even X0=OFF to stop instruction TCMP, M0~M2 still keep the state before X0 become OFF.

TRD will read the present clock information in D30~D36 (year, month, day, hour, minute, second, week).
X0 from OFF to ON, TCMP worked. Compare the three registers starting from S4 to three registers S1, S2, S3 (year, month, day). When S1, S2, S3 is larger than S4 clock, M0 is ON. When S1, S2, S3 is equal to S4 clock, M1 is ON. When S1, S2, S3 is smaller than S4 clock, M2 is ON.

For example, the present clock is 15:32:49 7,30,2014 Wednesday. D30=14, D31=7, D32=30, D33=15, D34=32, D35=49, D36=3. If the setting time is 1,6,2015, D20=15, D21=1, D22=6, Then M0=ON. If the setting time is 7,31,2014, D20=14, D21=7, D22=31, then M1=ON. If the setting time is 6,31,2014, D20=14, D21=6, D22=31, then M2=ON.
Note: if S4 is D33, it means hour, minute, second, then S1, S2, S3 mean hour, minute, second. S4 can start from year, month, day, hour; cannot start from minute, second. The week cannot compare.

For example:



The present clock is 15:32:49 7,30,2014 Wednesday. So D30=14, D31=7, D32=30, D33=15, D34=32, D35=49, D36=3. If the setting time is 15:32:49, D20=15, D21=32, D22=49, so Y1=ON. If the setting time is 17:32:49, D20=17, D21=32, D22=49, so Y0=ON. If the setting time is 2:32:5, D20=2, D21=32, D22=5, so Y2=ON.

5 HIGH SPEED COUNTER (HSC)

This chapter will introduce high speed counter's functions, including high speed count model, wiring method, read/write HSC value, reset etc.

Instructions List for HSC

Instruction name	Function	Instruction	Chapter
HSC read/write			
DMOV	HSC read	DMOV HSC0 D0	5-6-1
DMOV	HSC write	DMOV D4000 HSC0	5-6-2
CNT	No 24-segments single phase	CNT HSC0 K1000	5-7-1
CNT_AB	No 24-segments AB phase	CNT_AB HSC0 K1000	5-7-2
CNT	24-segments single phase	CNT HSC0 K1000 D0	5-7-3
CNT_AB	24-segments AB phase	CNT_AB HSC0 K1000 D0	5-7-4
RST	HSC reset	RST HSC0	5-8

5-1. Functions Summary

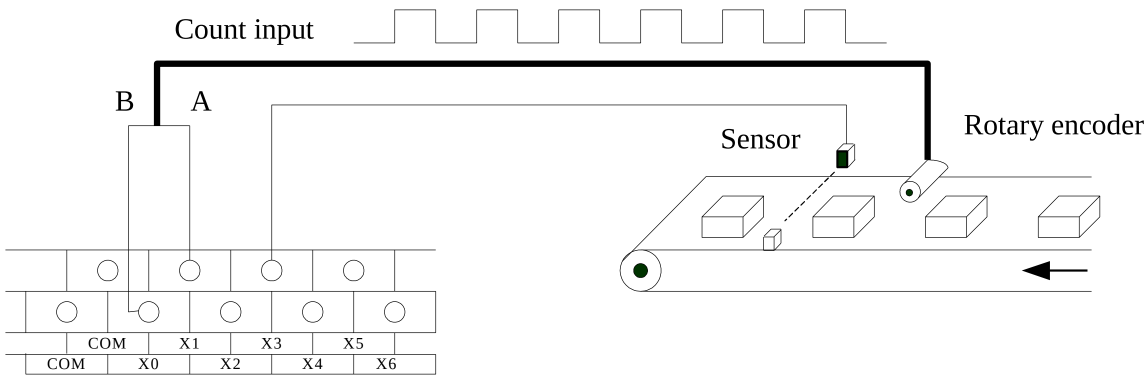
XD, XL series PLC has HSC (High Speed Counter) function which will not affect by the scanning cycle. Via choosing different counter, test the high speed input signals with detect sensors and rotary encoders. The highest testing frequency can reach 80 KHz.

Note:

(1) The high-speed counting input of XD/XL series PLC can only receive collector open-circuit signal (OC), but can not receive differential signal, so it is necessary to select the encoder of collector open-circuit signal (OC).

(2) When the counting frequency is higher than 25Hz, please select a high-speed counter.

(3) The XD1/XL1 series does not support high-speed counting.

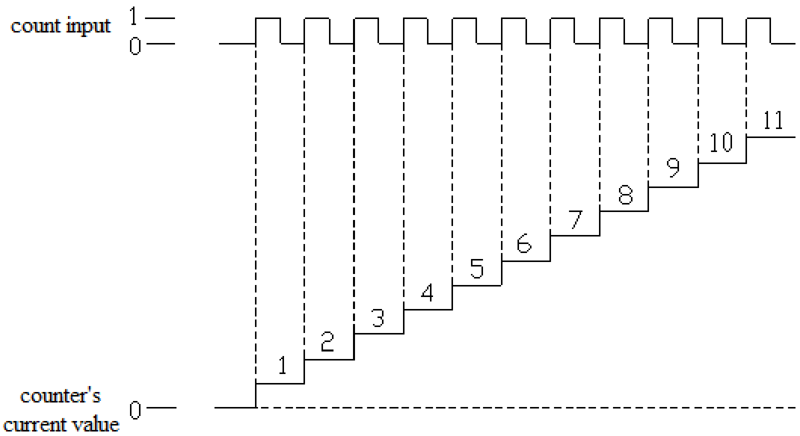


5-2. HSC Mode

XD, XL series high speed counter has two working mode: increasing mode and AB phase mode.

Increasing Mode

Under this mode, the count value increase at each pulse's rising edge;

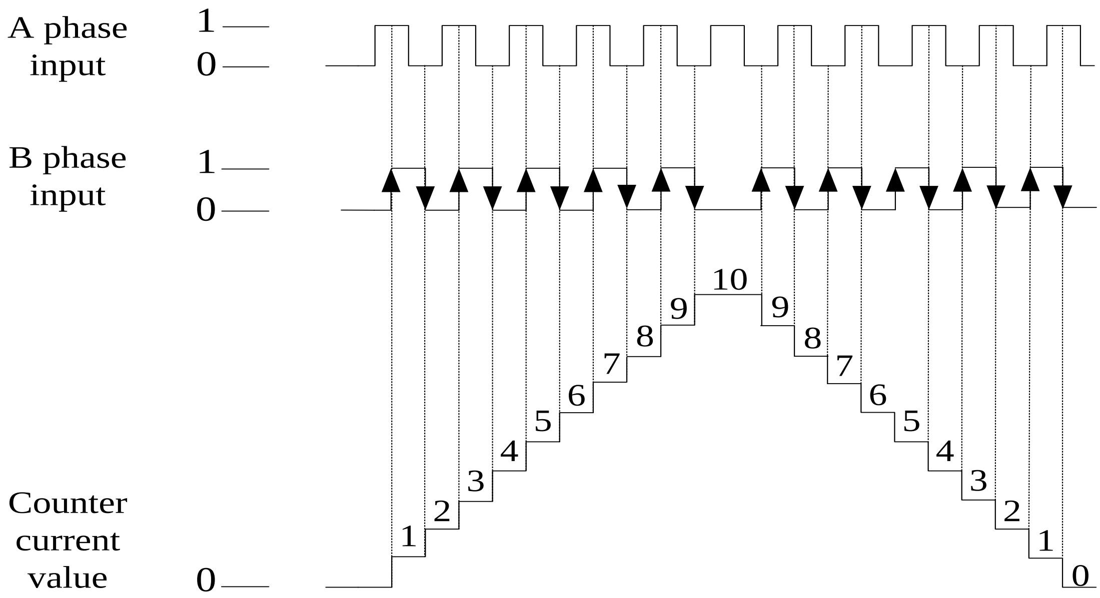


AB Phase Mode

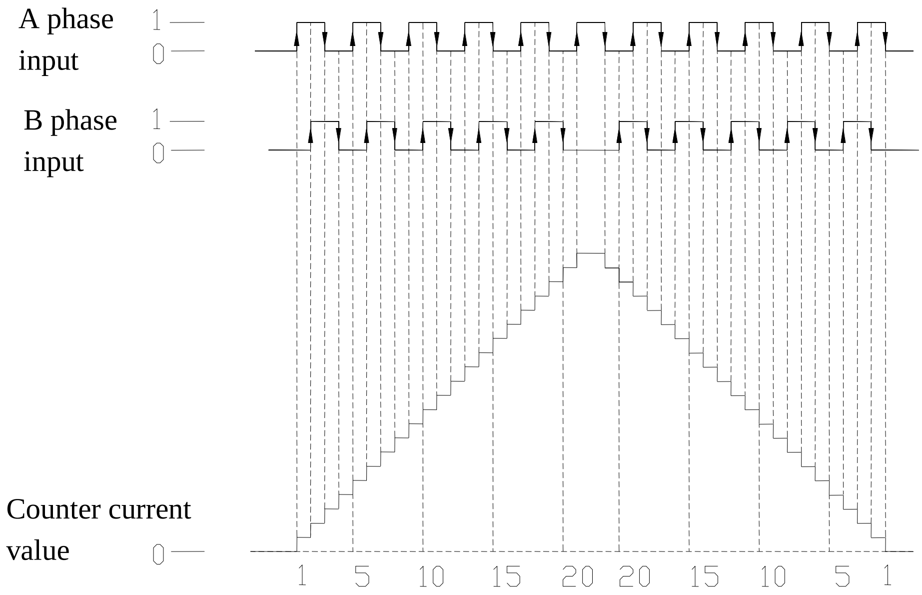
Under this mode, the HSC value increase or decrease according to two differential signal (A phase and B phase). According to the multiplication, we have 1-time frequency and 4-time frequency, but the default count mode is 4-time mode.

1-time frequency and 4-time frequency modes are shown below:

1-time Frequency



4-time Frequency



5-3. HSC Range

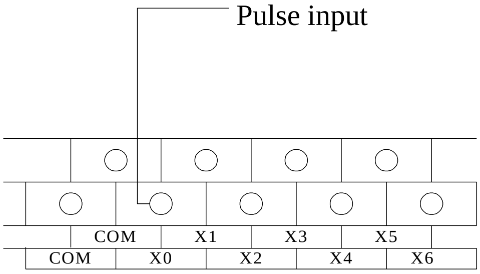
HSC's count range is: -2,147,483,648 ~ +2,147,483,647. If the count value overflows this range, then overflow or underflow appears;

Overflow means the count value jumps from +2,147,483,647 to -2,147,483,648, then continue counting; underflow means the count value jumps from -2,147,483,648 to +2,147,483,647 then continue counting.

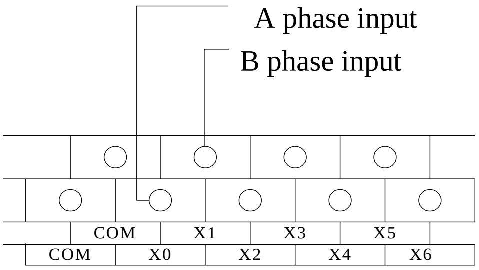
5-4. HSC Input Wiring

For the counter’s pulse input wiring, things differ with different PLC model and counter model; several typical input wiring diagrams are shown below: (take XD3-60 HSC0 as the example):

Increasing mode (counter HSC0)



AB phase mode (counter HSC0)



5-5. HSC ports assignment

Each letter’s Meaning:

U	A	B	Z
Pulse input	A phase input	B phase input	Z phase pulse catching

X can use as normal input terminals when there are no high speed pulses input. In the following table, Frequency doubling 2 means 2 frequency doubling; 4 means 4 frequency doubling; 2/4 means 2 and 4 frequency doubling.
Note: Z phase signal counting function is in developping.

XD2-16												
	Increasing mode							AB phase mode				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Max frequency	10K	10K	10K					5K	5K	5K		
Frequency doubling								2/4	2/4	2/4		
Counter interruption	√	√	√					√	√	√		
X000	U							A				
X001								B				

X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		

XD2-24/32, XD3-16/24/32, XD5-16/24/32, XL3-16												
	Increasing mode							AB phase mode				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Max frequency	80K	10K	10K					50K	5K	5K		
Frequency doubling								2/4	2/4	2/4		
Counter interruption	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		
X011												

XD2-48/60, XD3-48/60, XD5-48/60												
	Increasing mode							AB phase mode				
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC0	HSC2	HSC4	HSC6	HSC8
Max frequency	80K	80K	10K					50K	50K	5K		
Frequency doubling								2/4	2/4	2/4		
Counter interruption	√	√	√					√	√	√		
X000	U							A				
X001								B				
X002								Z				
X003		U							A			
X004									B			
X005									Z			
X006			U							A		
X007										B		
X010										Z		

XD5-24T4/32T4/48T4/60T4, XD5E-30T4, XDM-24T4/32T4/60T4/60T4L, XDC-24/32/48/60T XL5-32T4, XL5E-32T4, XLME-32T4												
	Increasing mode						AB phase mode					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Max frequency	80K	80K	80K	80K			50K	50K	50K	50K		
Frequency doubling							2/4	2/4	2/4	2/4		
Counter interruption	√	√	√	√			√	√	√	√		
X000	U						A					
X001							B					
X002							Z					
X003		U						A				

X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		

XD5-48T6/60T6												
	Increasing mode						AB phase mode					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Max frequency	80K	80K	80K	80K	80K	80K	50K	50K	50K	50K	50K	50K
Frequency doubling							2/4	2/4	2/4	2/4	2/4	2/4
Counter interruption	√	√	√	√	√	√	√	√	√	√	√	√
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		
X014					U						A	
X015											B	
X016											Z	
X017						U						A
X020												B
X021												Z

XD5-60T10, XDM-60T10, XD5E-60T10, XDME-60T10												
	Increasing mode											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Max frequency	80K	80K	80K	80K	80K	80K	80K	80K	80K	80K		
Frequency doubling												
Counter interruption	√	√	√	√	√	√	√	√	√	√		
X000	U											
X001												
X002												
X003		U										
X004												
X005												
X006			U									
X007												
X010												
X011				U								
X012												
X013												
X014					U							
X015												
X016												

X017						U						
X020												
X021												
X022							U					
X023												
X024												
X025								U				
X026												
X027												
X030									U			
X031												
X032												
X033										U		
X034												

XD5-60T10, XDM-60T10, XD5E-60T10, XDME-60T10												
	AB phase mode											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Max frequency	50K	50K	50K	50K	50K	50K	50K	50K	50K	50K		
Frequency doubling	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4		
Counter interruption	√	√	√	√	√	√	√	√	√	√		
X000	A											
X001	B											
X002	Z											
X003		A										
X004		B										
X005		Z										
X006			A									
X007			B									
X010			Z									
X011				A								
X012				B								
X013				Z								
X014					A							
X015					B							
X016					Z							
X017						A						
X020						B						
X021						Z						
X022							A					
X023							B					
X024							Z					
X025								A				
X026								B				
X027								Z				
X030									A			
X031									B			
X032									Z			
X033										A		
X034										B		
X035										Z		

5-6. AB phase counting frequency doubling setting

For AB phase counting, the frequency doubling can be set in special FLASH data registers SFD321, SFD322, SFD323... SFD330, when the value is 2, it is 2 frequency doubling, 4 is 4 frequency doubling.

Register name	Function	Setting value	Meaning
SFD320	HSC0 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD321	HSC2 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD322	HSC4 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD323	HSC6 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD324	HSC8 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD325	HSC10 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD326	HSC12 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD327	HSC14 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD328	HSC16 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling
SFD329	HSC18 frequency doubling	2	2 frequency doubling
		4	4 frequency doubling

Note: After the SFD register is modified, it is necessary to restart the high-speed counter (i.e. disconnect and reboot the drive condition) in order to make the new configuration effective!

5-7. HSC instruction

This section introduces the usage of single-phase high-speed counting instruction (CNT), AB-phase high-speed counting instruction (CNT_AB), reset of high-speed counting, reading and writing of high-speed counting.

5-7-1. Single phase HSC [CNT]

Instruction Summary

Single phase HSC instruction.

Single phase HSC [CNT]			
16 bits Instruction	-	32 bits Instruction	CNT
Execution condition	Normally ON/OFF coil	Suitable models	XD, XL
Hardware requirement		Software requirement	-

Operands

Operands	Function	Type
S	Specify HSC code (Eg. HSC0)	32 bits, BIN
D	Specify the compare value (Eg. K100, D0)	32 bits, BIN

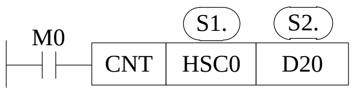
Suitable Soft Components

word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM	DS*	K/H	ID	QD
S1	Only can be HSC										
S2	●								●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

FUNCTIONS AND ACTIONS



- When M0 is on, HSC0 counts X0 signal in single phase mode, compares the high-speed counting value with the value set in register D20. When the high-speed counting value is equal to the set value, HSC0 coil is set on immediately, and the counting value is accumulated in HSCD0 (double words).
- If the driving condition M0 is not disconnected, HSC0 will remain on state and continue counting, and the counting value in HSCD0 will continue to accumulate.
- If the driving condition M0 is disconnected, HSC0 will remain on state and the counting value in HSCD0 will remain unchanged.
- During the counting process, if M0 is disconnected and connected again, the values in HSCD0 will continue to accumulate after the last counting value.

- In the counting process, if the setting value in D20 changes and the current counting value is less than the new setting value, then the new setting value is compared.

5-7-2. AB phase HSC [CNT_AB]

Instruction Summary

AB phase HSC instruction.

AB phase HSC [CNT_AB]			
16 bits Instruction	-	32 bits Instruction	CNT_AB
Execution condition	Normally ON/OFF coil	Suitable models	XD, XL(exclude XD1, XL1)
Hardware requirement		Software requirement	-

Operands

Operands	Function	Type
S	Specify HSC code (Eg. HSC0)	32 bits, BIN
D	Specify the compare value (Eg. K100, D0)	32 bits, BIN

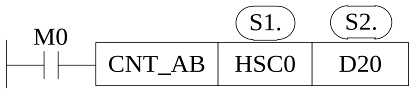
Suitable Soft Components

word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM	DS*	K/H	ID	QD
S1	Only can be HSC										
S2	●								●		

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

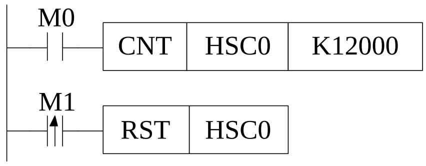
FUNCTIONS AND ACTIONS



- When M0 is on, HSC0 counts X0, X1 signal in AB phase mode, compares the high-speed counting value with the value set in register D20. When the high-speed counting value is equal to the set value, HSC0 coil is set on immediately, and the counting value is accumulated in HSCD0 (double words).
- If the driving condition M0 is not disconnected, HSC0 will remain on state and continue counting, and the counting value in HSCD0 will continue to accumulate.
- If the driving condition M0 is disconnected, HSC0 will remain on state and the counting value in HSCD0 will remain unchanged.
- During the counting process, if M0 is disconnected and connected again, the values in HSCD0 will continue to accumulate after the last counting value.
- In the counting process, if the setting value in D20 changes and the current counting value is less than the new setting value, then the new setting value is compared.

5-7-3. HSC reset [RST]

The reset mode of high-speed counter is software reset mode.



As shown above, when M0 is ON, HSC0 begins to count the pulse input of X0 port; when M1 changes from OFF to ON, HSC0 is reset, and the count value in HSCD0 (double words) is cleared.

5-7-4. Read HSC value [DMOV]

Instruction Summary

Read HSC value to the specified register;

Read HSC value [DMOV]			
16 bits Instruction	-	32 bits Instruction	DMOV
Execution condition	Normally ON/OFF, rising/falling edge	Suitable models	XD, XL (exclude XD1, XL1)
Hardware requirement		Software requirement	-

Operands

Operands	Function	Type
S	Specify HSC code	32 bits, BIN
D	Specify the read/written register	32 bits, BIN

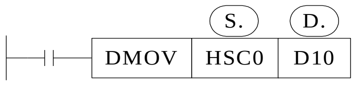
Suitable Soft Components

word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM	DS*	K/H	ID	QD
S	Only can be HSC										
D	●										

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

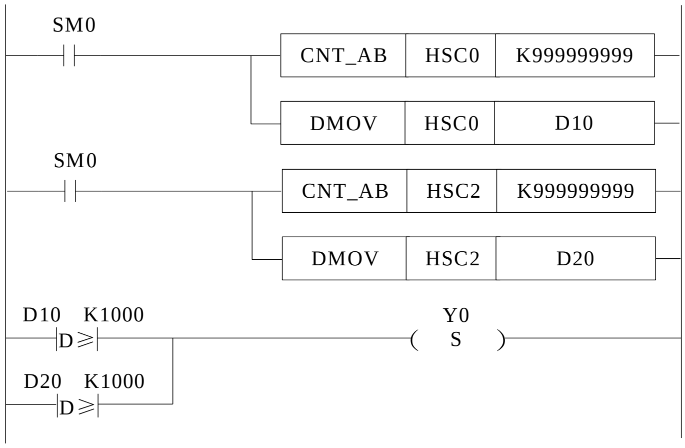
FUNCTIONS AND ACTIONS



When the trigger condition is established, the high-speed count value in the accumulative register HSCD0 (double words) corresponding to HSC0 of the high-speed counter is read into the data register D10 (double words).

High-speed counter can not directly participate in any application instructions or data comparison instructions (such as DMUL, LD > etc.) except DMOV, but can only be carried out after reading and writing into other registers.
As high speed counter is double words counter, so it must use 32-bit instruction DMOV. DMOV often uses together with high speed counter.

Program example:



5-7-5. Write HSC value [DMOV]

Instruction Summary

Write the specified register value into HSC;

Write HSC value [DMOV]			
16 bits Instruction	-	32 bits Instruction	DMOV
Execution condition	Normally ON/OFF, rising/falling edge	Suitable models	XD, XL (exclude XD1, XL1)
Hardware requirement		Software requirement	-

operands

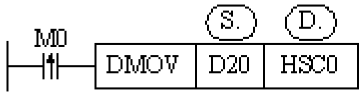
Operands	Function	Type
S	Specify HSC code	32 bits, BIN
D	Specify the read/written register	32 bits, BIN

suitable soft components

word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM	DS*	K/H	ID	QD
	S	●								●		
	D	Only can be HSC										

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

FUNCTIONS AND ACTIONS



When the trigger condition is established, The value in the double-word data register D20 is written into the accumulative register HSCD0 (double-word) corresponding to the HSC0 of the high-speed counter, and the original data is replaced.
High-speed counter can not directly participate in any application instructions or data comparison instructions (such as DMUL, LD > etc.) except DMOV, but can only be carried out after reading and writing into other registers.
As high speed counter is double words counter, so it must use 32-bit instruction DMOV. DMOV often uses together with high speed counter.

5-7-6. The difference between HSC and normal counter

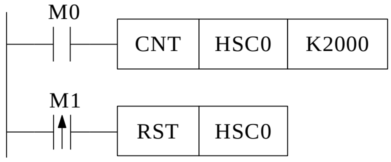
Although the instructions of high-speed counter use "CNT" in the same way as those of ordinary counter, their functions are quite different.
When M0 is changed from OFF to ON once, the value of common counter is added 1.
The high-speed counter trigger condition must be in the normally closed state when counting, which is equivalent to the high-number counter being activated, but the value of the high-number counter does not change. Only when the corresponding external signal input terminal receives the signal, the high-number counter counts. If the external signal input terminal has signal input and its trigger condition is not closed, the high-number counter will not count.
The difference is shown in the following table:

Counter type	Instruction format	Function
Normal counter	M0 — CNT C0 K2000	Count the OFF to ON times of M0, when the counting value reaches 2000, C0 is ON.
High-speed counter	M0 — CNT HSC0 K2000	When M0 is ON, count the X0 input signal, when the counting value reaches 2000, HSC0 is ON, M0 should be always ON when counting.

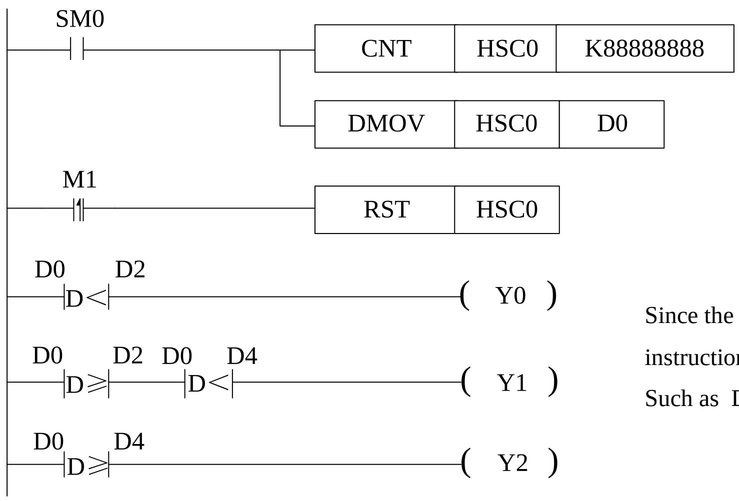
5-8. HSC Example

The following takes XD3-60 as an example to show the programming method of HSC.

Single-phase incremental mode



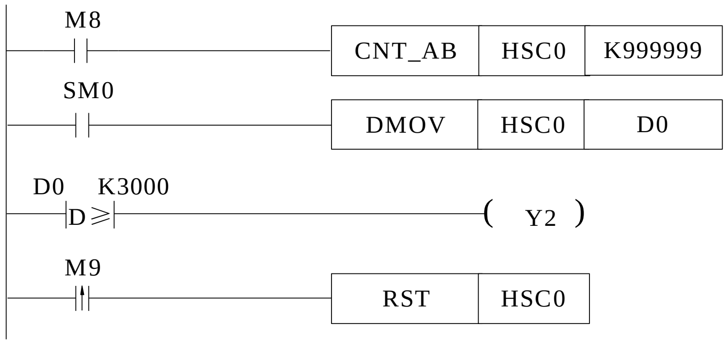
When the M0 is ON, HSC0 counts the rising edge of the OFF to ON of the input X0 port at high speed.
When M1 rising edge comes, reset HSC0 high-speed counter and HSCD0 (double word).



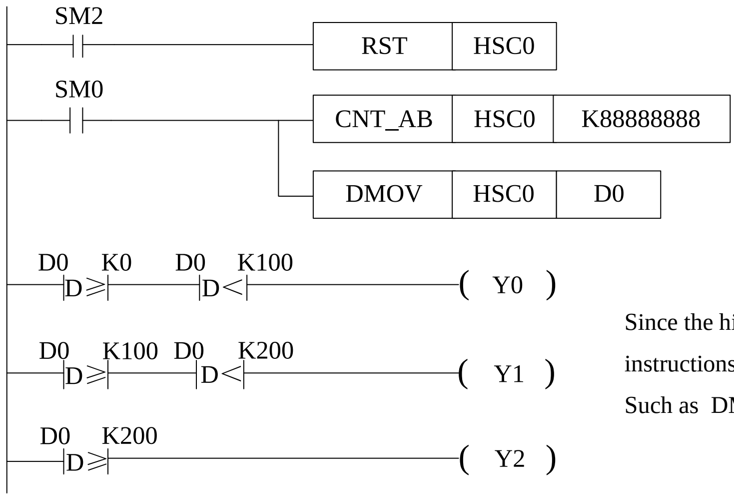
Since the high-speed count value is 32-bit, the instructions here are all 32-bit instructions. Such as DMOV，DLD<, DLD≥

- When SM0 is on, HSC0 counts X0 port in single-phase incremental mode, the setting value is K888888, and reads the high-speed counting value to D0 (double-word) in real time.
- When D0 (double words) is less than D2 (double words), Y0 is ON, when D0 (double words) is equal to or larger than D2 (double words) and less than D4 (double words), Y1 is ON. when D0 (double words) is equal to or larger than D4 (double words), Y2 is ON.
- When M1 rising edge is coming, reset HSC0 and HSCD0(double words).
- As the high speed counter is double words counter, please use double words instruction DLD < and DLD ≥.

AB phase input mode



- When M8 is ON, HSC0 starts to count. The signal inputs from X0 (A phase) and X1 (B phase).
- When SM0 is ON, the value in HSCD0 (double words) related to HSC0 is written to D0 (double words) in real-time.
- When the present counting value is over 3000, Y2 is ON.
- When the rising edge of M9 is coming, reset HSC0 and HSCD0 (double words).



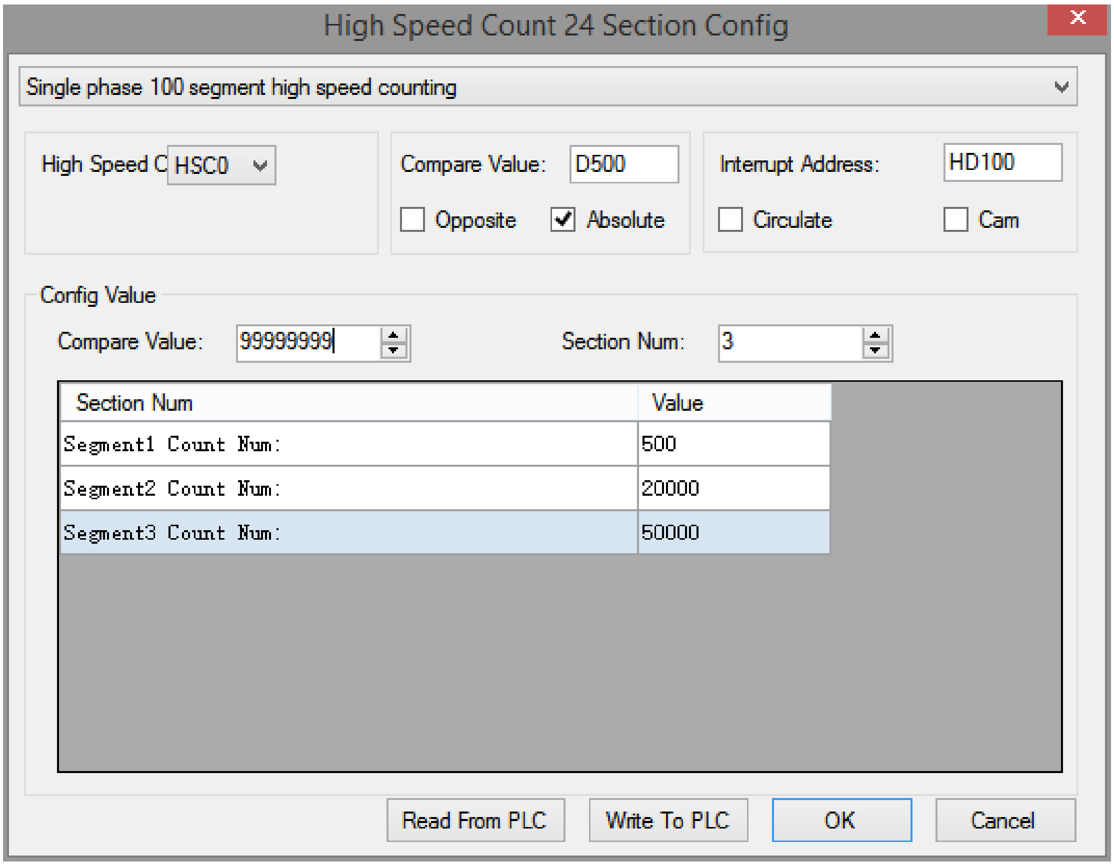
Since the high-speed count value is 32-bit, the instructions here are all 32-bit instructions. Such as DMOV，DLD<, DLD≥

- When the rising edge of the original forward pulse coil SM2 comes, that is, at the beginning of each scanning cycle, HSC0 is reset and the counting value in HSCD0 is cleared.
- When coil SM0 is on, HSC0 begins to count X0 and X1 ports in AB phase mode. The setting value of counting is K888888. At the same time, the counting value in HSCD0 (double words) is written into D0 (double words) in real time.
- When the counting value in D0 (double words) is greater than K0 and less than K100, the output coil Y0 is ON; when the counting value in D0 (double words) is greater than or equal to K100 and less than K200, the output coil Y1 is ON; and when the counting value in D0 (double words) is greater than or equal to K200, the output coil Y2 is ON.
- Since the high-speed counter is a double words counter, it is necessary to use the double words comparison instruction DLD ≥ and DLD < for comparison.

5-7. HSC interruption

5-7-1. Function overview and panel configuration

For XD/XL series PLC, some high-speed counters (referring to the high-speed counting input port allocation table of chapter 5-5 of each type of PLC) have a set value of 32 bits in 1-100 sections. When the difference of high-speed counting equals to the set value of corresponding 100 sections, the interruption will occur according to the corresponding interruption mark.
If the set value of N segment is set, there must be interrupt mark and interrupt program corresponding to N segment. The interruption marks corresponding to each high-speed counter are shown in chapter 5-9-4.
When using high-speed counting interrupt function, instructions can be written directly (see chapters 5-9-2 and 5-9-3), or can be configured by software panel. Please click HCNT in the XDPPro software, it will show below window.

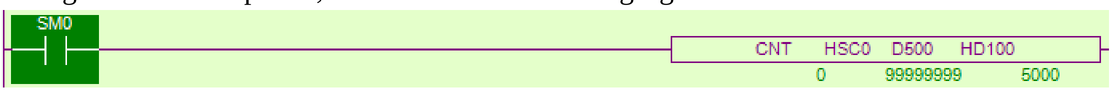


In this panel, we can configure the parameters related to high speed count interruption. Take the settings in above figure as an example to explain each parameter function.

Parameter		Function
Single phase 100 segment high speed counting	single phase 100 segments high speed counting	High Speed Counting in Single Phase Incremental Mode
	100 segments AB phase high speed counting	High Speed Counting in AB phase mode

High Speed C HSC0	HSC0~HSC18(32-bit）	High-speed counter number corresponding to high-speed input port
Compare Value: D500	Free to specify	HSC0 is ON when the count value is equal to the value in the register.
Compare Value: 99999999	Free to specify	When it counts to the compare value, HSC0 is ON, the compare value can be set here or put in compare reigster D500
☐ Opposite ☑ Absolute	Relative	It will produce the interruption of segment N when the counting value = segment N-1 interruption counting value + segment N setting value.
	Absolute	It will produce the interruption when the counting value is equal to setting value.
Interrupt Address: HD100	Free to specify	The set values of 100 segments of high-speed counting interrupts are stored in the registers starting from HD100, and the set values are stored in the double-word registers HD100, HD102, HD104….
☐ Circulate ☐ Cam	Interruption cycle	It must be used in relative mode. When all interrupts are over, high-speed counting interrupts can still be generated circularly.
	CAM	It must be used in absolute mode. When the counting value equals any set value, interruption occurs.
Section Num: 3	1~100 optional	If set to 3, it means execute three high-speed counting interrupts
Value	Free to specify	Each segment corresponds to an interrupt count value, which is written to the address block starting from HD100; the interrupt time is determined by the relative/absolute count mode

For detailed usage of the above parameters, please see the following chapters.
After writing to the PLC and clicking "OK", the high-speed count interrupt instruction configuration is completed, as shown in the following figure:



5-9-2．Single phase 100-segment HSC [CNT]

Summarization

Single phase 100-segment HSC instruction.

Single phase 100-segment HSC [CNT]			
16-bit instruction	-	32-bit instruction	CNT
Execution condition	Normal ON/OFF	Suitable model	XD, XL (exclude XL1, XD1)
Hardware requirements	-	Software requirements	-

Operand

Operand	Function	Type
S1	Set the HSC (for example: HSC0)	32 bits, BIN
S2	Set the compare value (eg. K100, D0)	32 bits, BIN
S3	Set the 100-segment setting value	32 bits, BIN

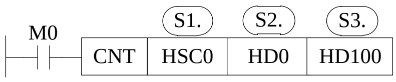
Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM	DS*	K/H	ID	QD
S1	Only can be HSC										
S2	●								●		
S3	●										

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



- When the high-speed counter HSC0 counts in single-phase mode, high-speed counting value is compared to data block starting from HD100 (such as HD102, HD102, HD104 and other double-word registers), it will immediately produce the corresponding high-speed counting interrupt when the condition is met, each section of the corresponding interrupt marks please refer to chapter 5-9-4.
- During the high-speed counting process, it is invalid to modify the set value of 100 segments.
- In the process of high-speed counting, the driving condition M0 can not be disconnected. If M0 is disconnected and then rebooted, no interruption will occur. The high-speed counter must be reset first, and then set ON M0 again to produce interruption.
- When the interrupt is finished in a single execution, if it needs to start the interruption again, the high-speed counter must be reset first, and then the driving condition must be ON again.
- In interrupt loop mode, interrupts can be generated in sequence as long as M0 remains

on state.

5-9-3. AB phase 100-segment HSC [CNT_AB]

Summarization

AB phase 100-segment HSC instruction.

AB phase 100-segment HSC [CNT_AB]			
16 bits instruction	-	32 bits instruction	CNT_AB
Execution condition	Normal ON/OFF	Suitable model	XD, XL (exclude XL1, XD1)
Hardware requirements	-	Software requirements	-

Operand

Operand	Function	Type
S1	Set the HSC (such as:HSC0)	32 bits, BIN
S2	Set the compare value (such as: K100, D0)	32 bits, BIN
S3	Set the 100-segment setting value	32 bits, BIN

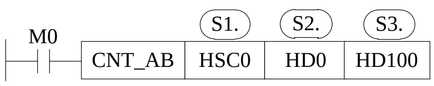
Suitable soft components

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM	DS*	K/H	ID	QD
S1	Only can be HSC										
S2	●								●		
S3	●										

*Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Description



- When the high-speed counter HSC0 counts in AB phase mode, high-speed counting value is compared to data block starting from HD100 (such as HD102, HD102, HD104 and other double-word registers), it will immediately produce the corresponding high-speed counting interrupt when the condition is met, each section of the corresponding interrupt marks please refer to chapter 5-9-4.
- During the high-speed counting process, it is invalid to modify the set value of 100 segments.
- In the process of high-speed counting, the driving condition M0 can not be disconnected. If M0 is disconnected and then rebooted, no interruption will occur. The high-speed counter must be reset first, and then set ON M0 again to produce interruption.

- When the interrupt is finished in a single execution, if it needs to start the interruption again, the high-speed counter must be reset first, and then the driving condition must be ON again.
- In interrupt loop mode, interrupts can be generated in sequence as long as M0 remains on state.

5-9-4. Interruption flag of HSC

The 100 segments interruption flags of each HSC are in the following table. For example, the 100 segments interruption flags of HSC0 are I2000, I2001, I2002….. I2099.

HSC	Interruption flag					
	Segment 1	Segment 2	Segment 3	……	Segment N	Segment 100
HSC0	I2000	I2001	I2002	……	I（2000+N-1）	I2099
HSC2	I2100	I2101	I2102	……	I（2100+N-1）	I2199
HSC4	I2200	I2201	I2202	……	I（2200+N-1）	I2299
HSC6	I2300	I2301	I2302	……	I（2300+N-1）	I2399
HSC8	I2400	I2401	I2402	……	I（2400+N-1）	I2499
HSC10	I2500	I2501	I2502	……	I（2500+N-1）	I2599
HSC12	I2600	I2601	I2602	……	I（2600+N-1）	I2699
HSC14	I2700	I2701	I2702	……	I（2700+N-1）	I2799
HSC16	I2800	I2801	I2802	……	I（2800+N-1）	I2899
HSC18	I2900	I2901	I2902	……	I（2900+N-1）	I2999

5-9-5. Setting value meaning in absolute or relative mode

The setting value meaning is different in absolute and relative mode. Relative/absolute mode can be set in the software panel. It can also be modified by special Flash register SFD330. (Note: Driving conditions must be OFF and ON again to make the configuration effective.)
0: Relative mode;
1: Absolute mode.

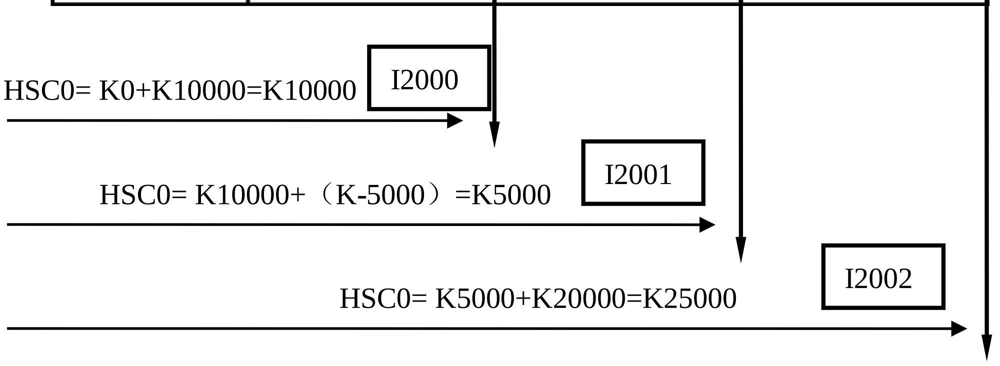
- **Relative mode**

In relative mode, the set value of high-speed counting 100 segments is relative cumulative value. When the set value of counting equals the sum of the interruption count value of N-1 segment and the set value of N segment, the segment N interrupt is generated.
N interrupt markers correspond to N interrupt settings. The N+1 interrupt settings register is reserved for other purposes.

Example1:
The current value of HSC0 is 0, segment one preset value is 10000, the preset value in segment 2 is －5000, the preset value in segment 3 is 20000. When starting to count, when the counter's current value is 10000, it generates the segment 1 interruption I2000; when the counter's current value is 5000, it generates the segment 2 interruption I2001; when the counter's current value is 25000, it generates the segment 3 interruption I2002.

See graph below:

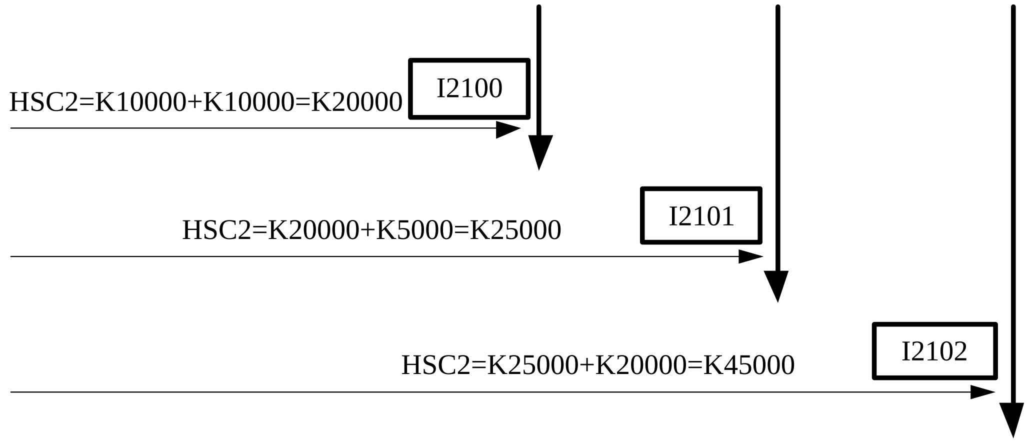
HSC0	HD0	HD1	HD2	HD3	HD4	HD5
K0	K10000		K-5000		K20000	



Example 2:
HSC2 current value is 10000, the segment one preset value is 10000, the preset value of segment 2 is 5000, the preset value of segment 3 is 20000. When starting to count, when the counter's current value is 20000, it generates the segment 1 interruption I2100; when the counter's current value is 25000, it generates the segment 2 interruption I2101; when the counter's current value is 45000, it generates the segment 3 interruption I2102.

See graph below:

HSC2	HD0	HD1	HD2	HD3	HD4	HD5
K10000	K10000		K5000		K20000	



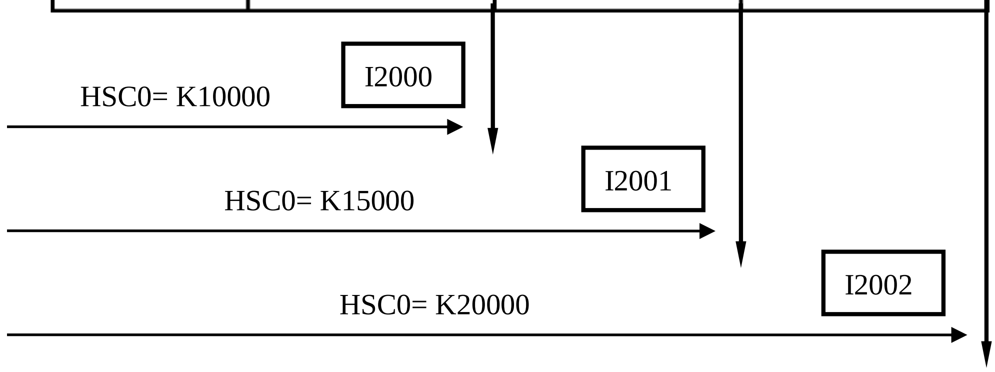
- **Absolute Mode**

In absolute mode, interruption occurs when the count value equals the set value of each section of the counter. N interrupt markers correspond to N interrupt settings. The N+1 interrupt settings register is reserved for other purposes.

Example 1:

The current value of counter HSC0 is 0, the setting value of segment 1 is 10000, the setting value of segment 2 is 15000, and the setting value of segment 3 is 20000. When it starts counting, if the current value of the counter is 10000, the segment 1 interruption I2000 is generated; when the current value of the counter is 15000, the segment 2 interruption I2001 is generated; when the current value of the counter equals 20000, the segment 3 interruption I2002 is generated.

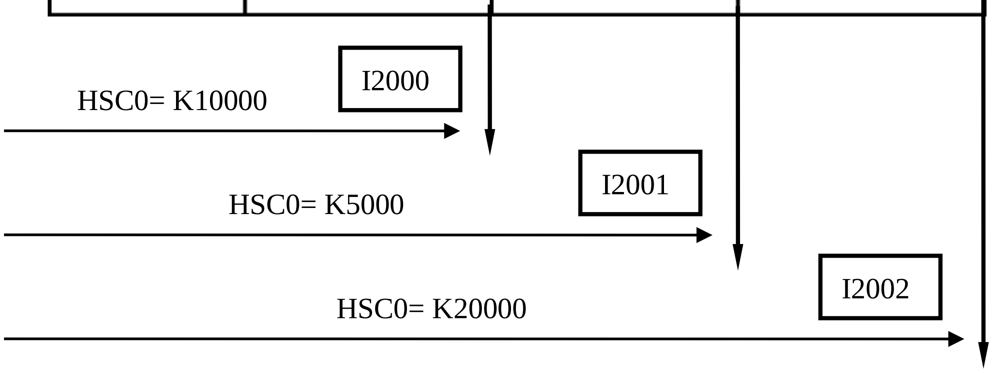
HSC0	HD0	HD1	HD2	HD3	HD4	HD5
K0	K10000		K15000		K20000	



Example 2:

The current value of counter HSC2 is 5000, segment 1 set value is 10000, segment 2 set value is 5000, and segment 3 set value is 20000. When it starts counting, if the current value of the counter is 10000, segment 1 interrupt I2100 is generated; when the current value of the counter is 5000, segment 2 interrupt I2101 is generated; when the current value of the counter equals 20000, segment 3 interrupt I2102 is generated.

HSC0	HD0	HD1	HD2	HD3	HD4	HD5
K0	K10000		K5000		K20000	



Note: When absolute counting is performed in non-cam mode, counting interrupts are generated sequentially, i.e.,segment 1 interruption, segment 2 interruption, segment 3 interruption... When a segment interrupt occurs, no interrupt occurs even if the count value reaches the set value of the segment again.

As in the example above, if the count value is increased from 4000 to 5000 and 10000 after the interruption of segment 1 and 2, the interruption of segment 1 and 2 will not occur again, and the interruption of segment 3 will occur when the count value continues to increase to 20000.

5-9-6. HSC interruption cycle mode

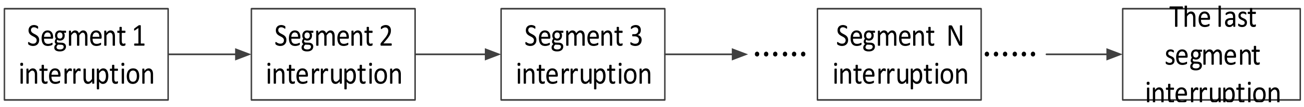
Mode 1: Single loop (normal mode)

The HSC interruption will not happen after it ends. The following conditions can start the interruption again.

reset the HSC

Reboot the HSC activate condition

The interruption is generated as the following sequence when single loop execution:



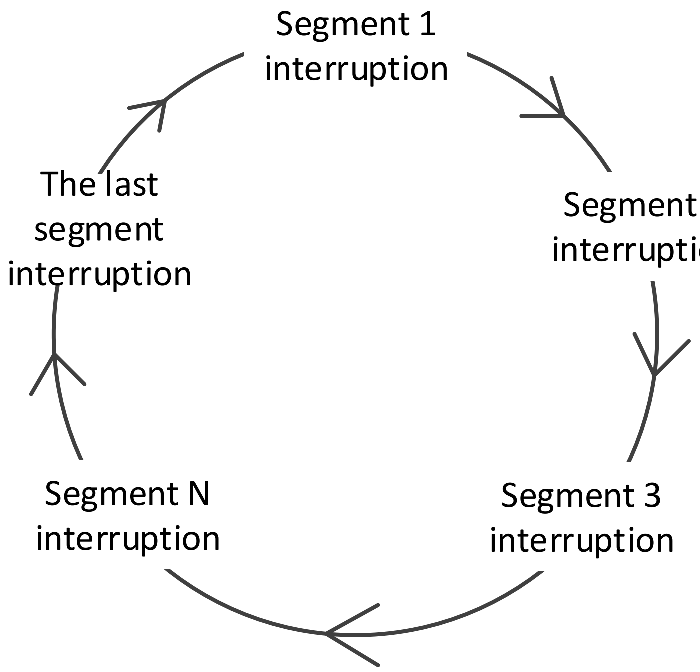
Mode 2: Continuous loop

Continous loop interruption is only suitable for relative counting mode. In continuous loop mode, the interruption will start again after it is completed. This mode is especially suitable for the following application:

continuous back-forth movement.

Generate cycle interruption according to the fixed pulse.

When continuous loop interruption is performed (without cam function enabled), interrupts occur in the following order:



Via setting SFD331, users can switch between single loop mode or continuous loop mode. The detailed assignment is show below:
(Note: the settings will be effective after setting OFF and ON the driving condition again)

Address	HSC	Setting
Bit0	100 segments HSC interruption cycle (HSC0)	0: single loop 1: continuous loop
Bit1	100 segments HSC interruption cycle (HSC2)	
Bit2	100 segments HSC interruption cycle (HSC4)	
Bit3	100 segments HSC interruption cycle (HSC6)	
Bit4	100 segments HSC interruption cycle (HSC8)	
Bit5	100 segments HSC interruption cycle (HSC10)	
Bit6	100 segments HSC interruption cycle (HSC12)	
Bit7	100 segments HSC interruption cycle (HSC14)	
Bit8	100 segments HSC interruption cycle (HSC16)	
Bit9	100 segments HSC interruption cycle (HSC18)	

5-9-7. CAM function of high speed counter interruption

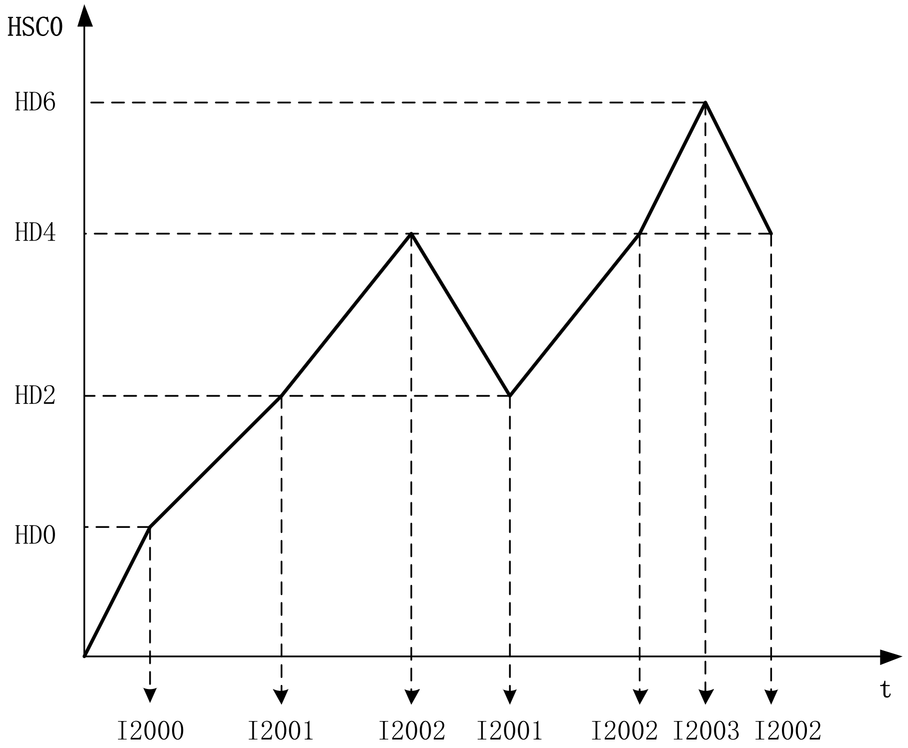
High-speed counting cam: After setting all interruption set value, the high-speed counting cam function is selected. When the high-speed counting value is equal to any of the interruption set value, the corresponding high-speed counting interruption (the same as the 100-segment high-speed counting interruption marker) is executed immediately. When the high-speed counting value changes repeatedly, the same high-speed interruption of the cam can be executed repeatedly.

High-speed counting cam not only can fully realize the cyclic sequence interruption function of ordinary electronic cam, but also can generate multiple times of positive and negative single point interruption in single cycle. It is widely used in control systems of high-speed winding machine and packaging machine.

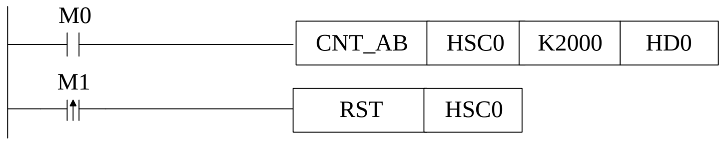
Note: CAM function is only fit for absolute counting mode.

Cam function can be set by configuration panel in XINJE PLC software, or by special Flash register SFD332: (Note: Drive condition must be set OFF and ON again to make configuration effective)
0: No cam function enabled
1: Enable Cam Function

Example:
Four values are stored in four consecutive double-word registers starting with register HD0. When HSC0 starts to count, if the HSC0 count value equals any of the four registers, the corresponding interrupt signal will be generated immediately. As shown in the following figure:



5-9-8. Interruption using notes and parameter address



LD M0 //HSC trigger condition M0 (also interruption counting condition)
CNT_AB HSC0 K2000 HD0 //HSC and 100-segment head address setting
LDP M1 //HSC reset trigger condition
RST HSC0 //HSC and 100-segment reset (also reset the interruption)

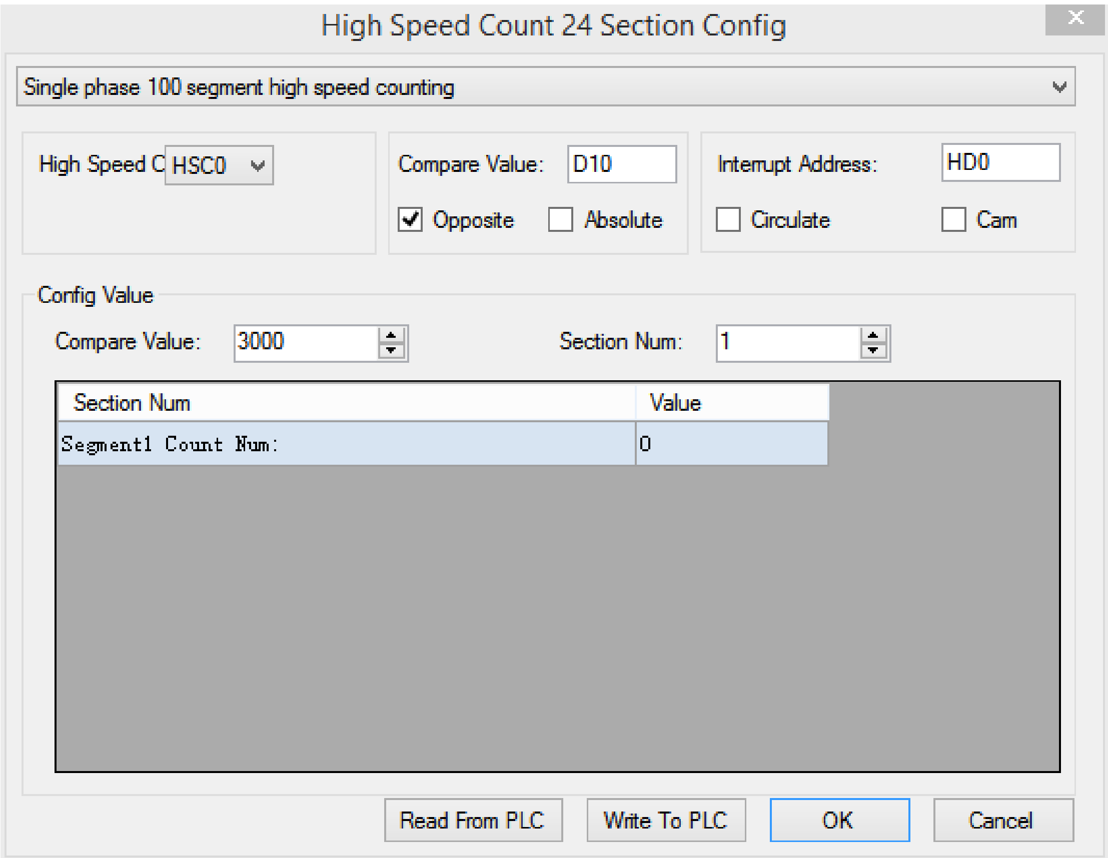
As shown in the above example (note: the interrupt subprogram is omitted, see the application example in chapter 5-9-9). The data register HD0 sets the region starting address for the set value of 100 segments, and then stores the set value of 100 segments in double-word form. Attention should be paid to using high-speed counting interrupts:

- The register after the last segment no needs to set 0, but should be reserved and cannot be used for other purpose. For example, it has 3 segments, segment 1 is HD0, segment 2 is HD2, segment 3 is HD4, then HD6 is reserved.
- It is not allowed to set the interrupt setting value without writing the interrupt program. Otherwise, errors will occur.
- 100-segment interrupt of high speed counter generate in turn, that is, if the first interrupt does not occur, the second interrupt will not occur.
- In high speed counting process, if the present counting value is changed by DMOV, ADD instruction (DMOV K1000 HSCD0), the interruption value will not change at this time. Please do not change the HSCD value when the high speed counter is running.

Some parameters can be modified in special Flash registers, as shown in the following table:

Parameter	Register address	Setting value
Counting mode	SFD330	0: relative 1: absolute
Execution mode	SFD331	0: execution once 1: interruption cycle
CAM function	SFD332	0: not enable 1: enable cam function

The above parameters can also be configured by the configuration panel in the following way: Move the mouse over the high-speed counting instruction and right-click it. Select "CNT_AB Instruction Parameter Configuration" from the drop-down menu. A configuration panel will appear to configure the parameters in this window. As shown in the following figure:



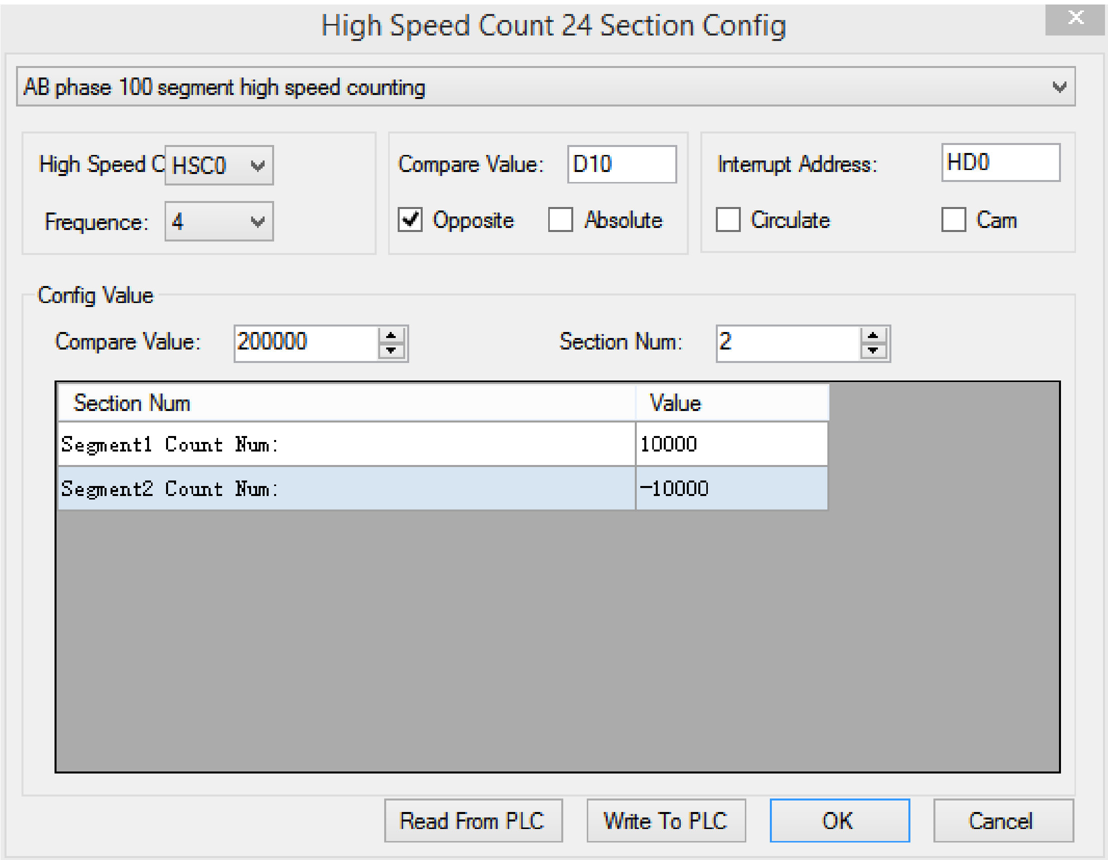
5-9-9. Application of HSC interruption

Application 1:

When M0 is ON, HSC0 starts counting. The counting value is stored in the address starting from HD0. When it reaches the set value, the interruption is produced. When the rising edge of M1 is coming, clear the HSC0.

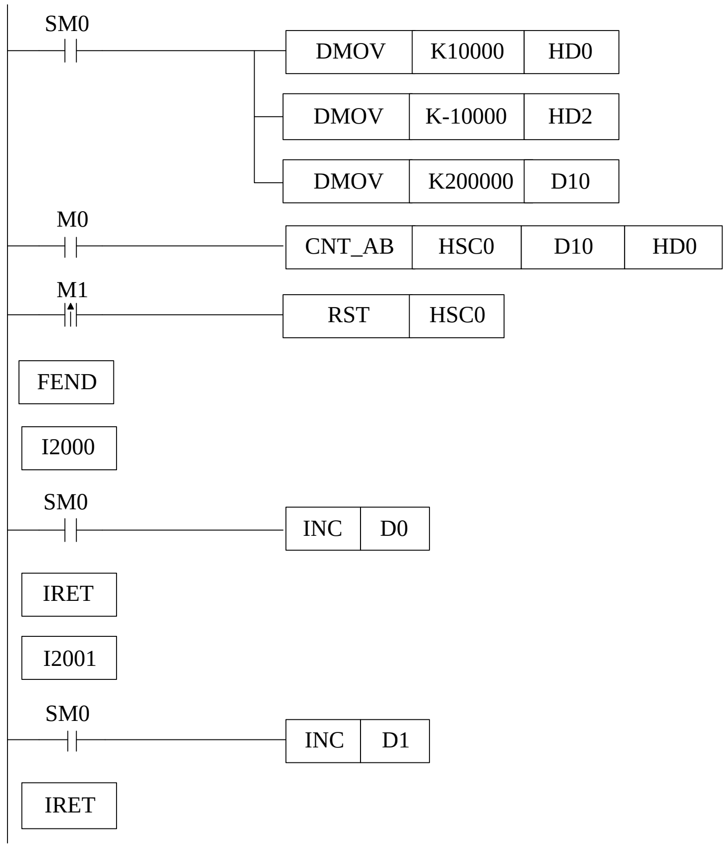
Method 1:

Configure the parameters through XDPpro software:



Configure item	**Function**
High speed counter	Choose HSC, the range is from HSC0 to HSC18
Frequency	Choose the HSC frequency doubling (2 or 4)
Compare value	The value can be register or constant, in this example, when the counting value reaches compare value, HSC0 is ON. here the compare value is 200000 which is saved in D10.
Relative and absolute	The HSC is relative mode or absolute mode
Interrupt address	The starting registers to store 100 segments interruption preset value
Circulate	100 segments interruption mode is cycle or not
Cam	The cam function is executed when any set value of 100-segment high speed counting interruption equals the counting value.

Method 2: make the program

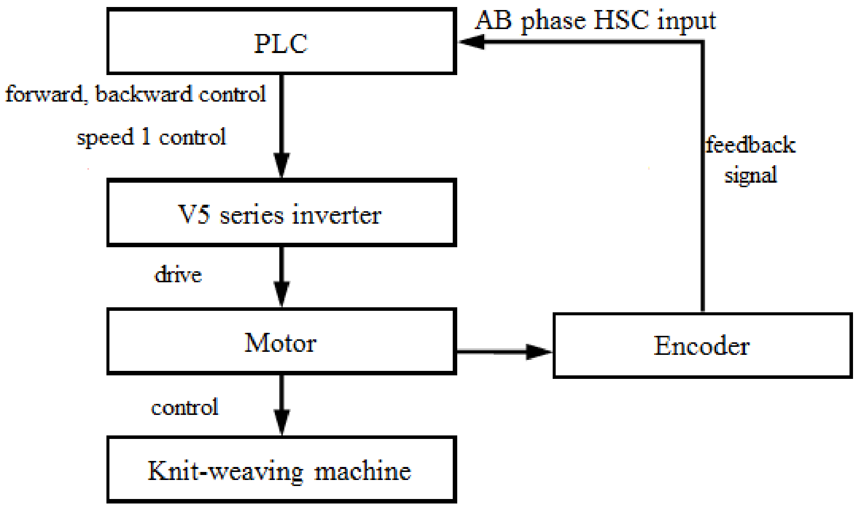


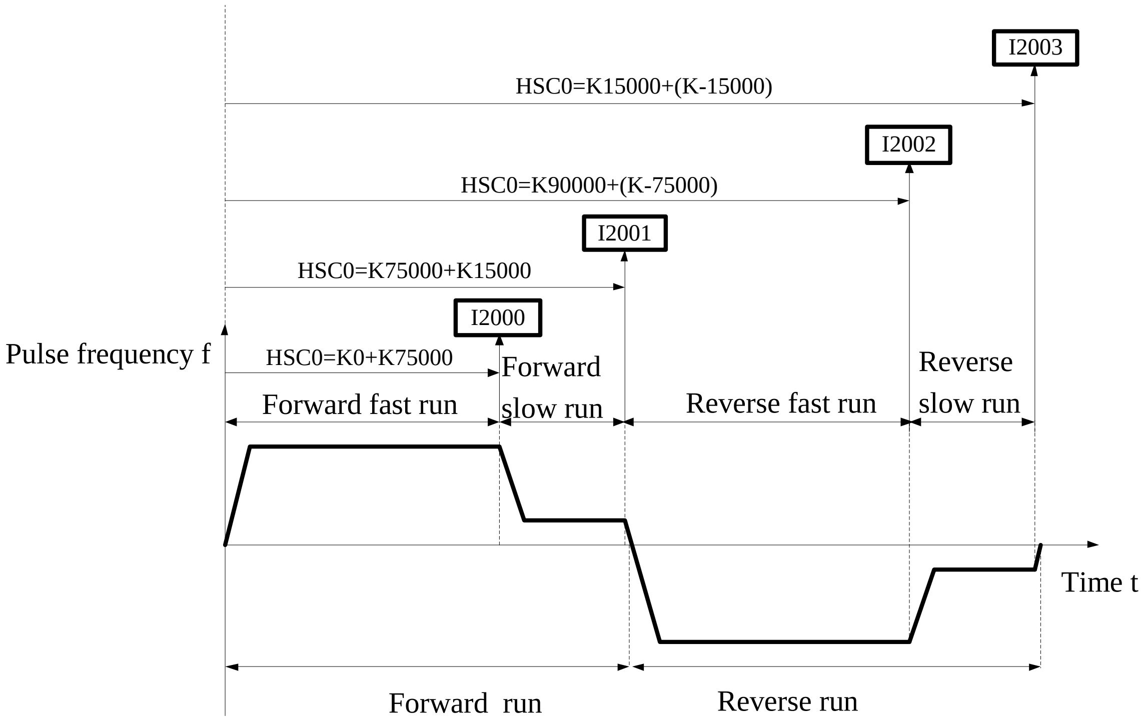
Instruction:

```
LD      SM0                     //SM0 is normally ON coil
DMOV K10000 HD0                 //segment one preset value HD0 is 10000
DMOV K-10000 HD2                //segment 2 preset value HD2 is -10000
DMOV K200000   D10              //set HSC compare value
 LD     M0                      //HSC activate condition M0
CNT_AB  HSC0 D10 HD0            //HSC interruption instruction
LDP     M1                      //HSC reset condition M1
RST     HSC0                    //reset HSC and 100 segments interruption
FEND                            //the main program end
I2000                           //segment one interruption flag
LD      SM0                     //SM0 is normally ON coil
INC     D0                      //D0= D0+1
IRET                            //interruption return flag
I2001                           //segment 2 interruption flag
LD      SM0                     //SM0 is normally ON coil
INC     D1                      //D1= D1+1
IRET                            //interruption return flag
```

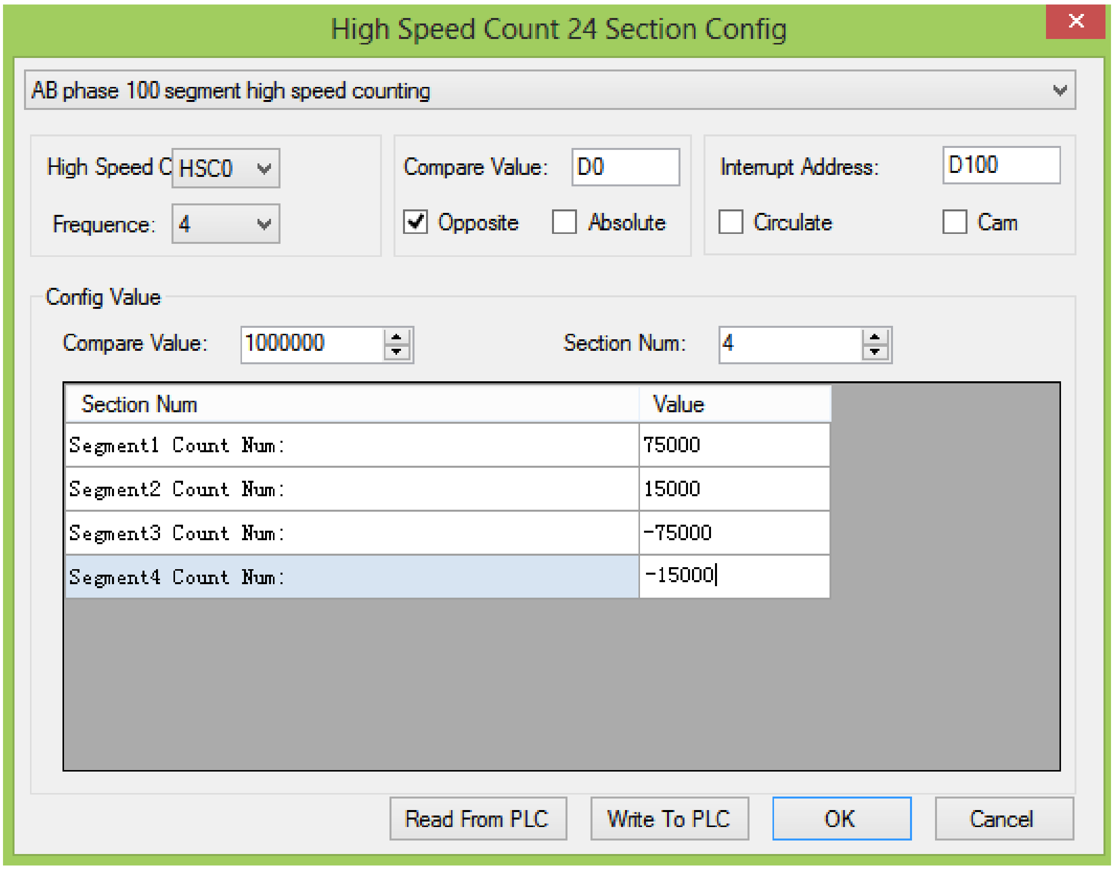
Application 2: knit-weaving machine (continuous loop mode)

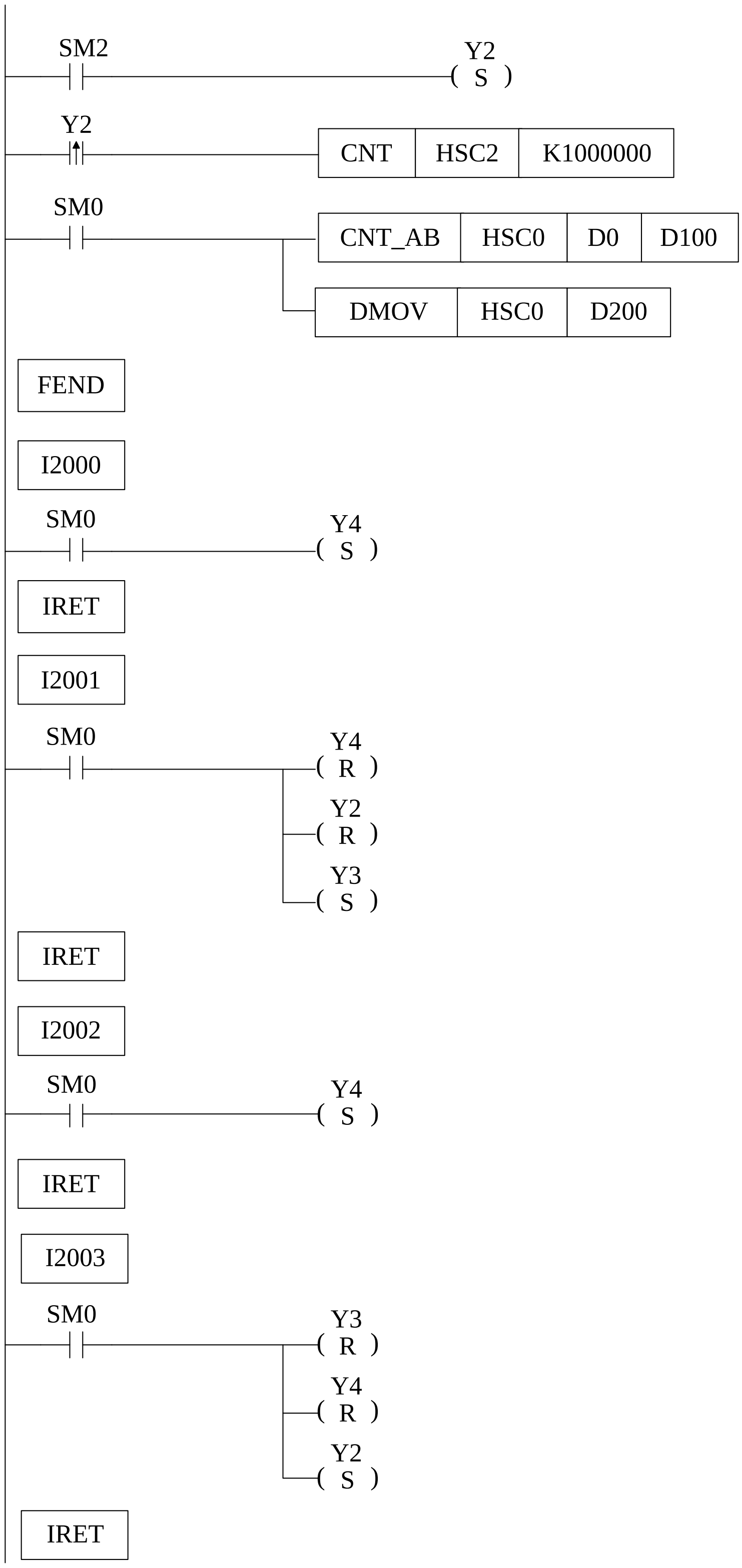
The machine principle: Control the inverter via PLC, thereby control the motor. Meantime, via the feedback signal from encoder, control the knit-weaving machine and the precise position.





Below is PLC program: Y2 represents forward output signal; Y3 represents reverse output signal; Y4 represents output signal of speed 1; HSC2: Back-forth times accumulation counter; HSC0: AB phase HSC;


High Speed Count 24 Section Config
AB phase 100 segment high speed counting
High Speed C
HSC0
Frequence:
4
Compare Value:
D0
Opposite
Absolute
Interrupt Address:
D100
Circulate
Cam
Config Value
Compare Value:
1000000
Section Num:
4
Section Num
Value
Segment1 Count Num:
75000
Segment2 Count Num:
15000
Segment3 Count Num:
-75000
Segment4 Count Num:
-15000
Read From PLC
Write To PLC
OK
Cancel


SM2
Y2
S
Y2
CNT
HSC2
K1000000
SM0
CNT_AB
HSC0
D0
D100
DMOV
HSC0
D200
FEND
I2000
SM0
Y4
S
IRET
I2001
SM0
Y4
R
Y2
R
Y3
S
IRET
I2002
SM0
Y4
S
IRET
I2003
SM0
Y3
R
Y4
R
Y2
S
IRET

Instruction List:

```
LD      SM2                        //SM2 is initial ON coil
SET     Y2                         //set ON Y2 (forward run)
LDP     Y2                         // Back-forth times activate condition Y2
CNT  HSC2  K1000000                //HSC2 starts counting
LD      SM0                        //SM000 is normal ON coil
CNT_AB  HSC0  D0  D100             //HSC 100 segments first address
DMOV  HSC0  D200                   //read HSC0 counting value to D200
FEND                               //main program end
I2000                              //Interruption 1 flag
LD      SM0                        //SM0 is normal ON coil
SET     Y4                         //set ON Y4 (run at speed 1)
IRET                               //interruption return
I2001                              //interruption 2 flag
LD      SM0                        //SM0 is normal ON coil
RST     Y4                         //reset Y4 (stop running at speed 1)
RST     Y2                         //reset Y2 (stop forward running)
SET     Y3                         //set ON Y3 (reverse running)
IRET                               //interruption return
I2002                              //interruption 3 flag
LD      SM0                        //SM0 is normal ON coil
SET     Y4                         //set ON Y4 (run at speed 1)
IRET                               //interruption return
I2003                              //interruption 4 flag
LD      SM0                        //SM0 is normal ON coil
RST     Y3                         //reset Y3 (stop reverse running)
RST     Y4                         //reset Y4 (stop running at slow speed)
SET     Y2                         //set on Y2 (forward running)
IRET                               //interruption return
```

6 Communication Function

This chapter mainly includes: basic concept of communication, Modbus communication and free communication.

Relative Instruction

Mnemonic	Function	Circuit and soft components	Chapter
MODBUS Communication			
COLR	Coil Read	COLR S1 S2 S3 D1 D2	6-2-3
INPR	Input coil read	INPR S1 S2 S3 D1 D2	6-2-3
COLW	Single coil write	COLW D1 D2 S1 S2	6-2-3
MCLW	Multi-coil write	MCLW D1 D2 D3 S1 S2	6-2-3
REGR	Register read	REGR S1 S2 S3 D1 D2	6-2-3
INRR	Input register read	INRR S1 S2 S3 D1 D2	6-2-3
REGW	Single register write	REGW D1 D2 S1 S2	6-2-3
MRGW	Multi-register write	MRGW D1 D2 D3 S1 S2	6-2-3
Free Communication			
SEND	Send data	SEND D10 D100 K2	6-3-4
RCV	Receive data	RCV D20 D200 K2	6-3-4
Read and write serial port data			
CFGCR	Read serial port	CFGCR HD0 K7 K2	6-5-1
CFGCW	Write serial port6-3-4	CFGCW HD0 K8 K2	6-5-1

6-1. Summary

XD, XL series PLC main units can fulfill your requirement on communication and network. They not only support Modbus RTU, but also support Modbus ASCII and field bus X-NET. XD, XL series PLC offer multiple communication methods, with which you can communicate with the devices (such as printer, instruments etc.) that have Modbus communication protocol.

6-1-1. COM port

COM Port

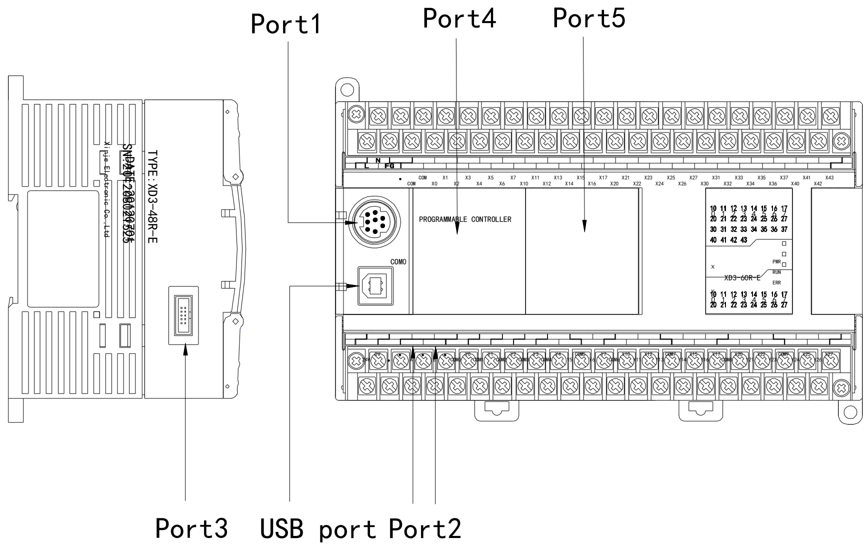
XD, XL series PLC have multiple communication ports, such as USB port, Ethernet port, port0~port5, port2-RS232, port2-RS485.

×not support √support

	USB	RJ45	Port0	Port1	Port2	Port2-RS232	Port2-RS485	Port3	Port4	Port5
XD1	×	×	√	√	√	×	×	×	×	×
XD2	×	×	√	√	√	×	×	√	√	√
XD3	√	×	×	√	√	×	×	√	√	√
XD5	√	×	×	√	√	×	×	√	√	√
XDM	√	×	×	√	√	×	×	√	√	√
XDC	×	×	×	√	×	√	√	√	√	√
XD5E	√	√	×	√	√	×	×	√	√	√
XDME	√	√	×	√	√	×	×	√	√	√
XL1	×	×	×	√	√	×	×	×	×	×
XL3	√	×	×	√	√	×	×	√	×	×
XL5	√	×	×	√	√	×	×	√	×	×
XL5E	×	√	×	√	√	×	×	√	×	×

Note: In the series of " √ " PLCs, there may be some models that do not support USB port or Port2-Port5. See Appendix 5 for details.

The distribution of XD series communication ports is as follows:



Note:

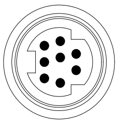
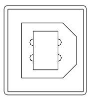
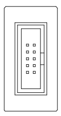
(1) The USB port of some models is RJ45 port or Port0 port or Port2-RS232;

(2) Port 1 port of some models is RJ45 port.

(2) Port2 port of some models is Port2-RS485 port or RJ45 port.

(3) The left-most output terminal of XD5E is USB port or RS232 port.

The definitions and functions of each communication port are as follows:

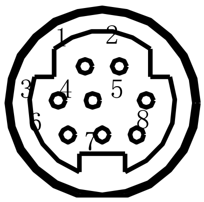
Port	Appearance	Definition	protocol	Function
Port0		RS232 port	X-NET	Download program, set the port parameters through software or xinje config tool
Port1		RS232 port	Modbus RTU Modbus ASCII Free communication X-NET	Download program and connect external devices, set the port parameters through software or xinje config tool
Port2-RS232		RS232 port	Modbus RTU Modbus ASCII Free communication X-NET	Download program and connect external devices, set the port parameters through software or xinje config tool
Port2-RS485	A, B port	RS485 port	Modbus RTU Modbus ASCII Free communication X-NET	Download program and connect external devices, set the port parameters through software or xinje config tool
Port2	A, B port	RS485 port		
USB 口		USB port	X-NET	High speed download port, please install the USB driver first
RJ45		Ethernet port	TCP/IP communication based on Ethernet	High speed stable download/upload program and data, remote monitoring, communicate with TCP IP device in LAN, set the port parameters through software or xinje config tool
Port3		Left extension ED port (for extending RS232/RS485 port)	Modbus RTU Modbus ASCII Free communication X-NET	connect external devices, set the port parameters through software or xinje config tool
Port4		Above extension BD port/ RS232/RS485/Optical fiber port (see below details)	Modbus RTU Modbus ASCII Free communication X-NET	connect external devices, set the port parameters through software or xinje config tool
Port5				

Note:

(1) Port2-RS232 and Port2-RS485 of XDC series can not be used simultaneously; when configuring in programming software, the port number is COM2, just like Port2.

(2) If the parameters of Port1 can not be online after modification, the problem can be solved by "stop PLC when reboot", initialization after successful stopping, and then re-power-on; if not necessary, it is better not to modify the communication parameters of Port1.
(3) The communication function of X-NET is not within the scope of this manual. Please refer to the "X-NET User Manual".
(4) The content of Ethernet communication is not within the scope of this manual. Please refer to the User Manual of TCP IP Communication Based on Ethernet.

1. RS232 port (port0, port1, port2-RS232)



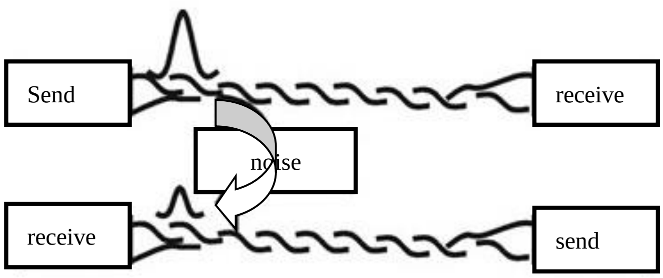
2：PRG
4：RxD
5：TxD
6：VCC
8：GND

Mini Din 8-pin plug (holes)

2. RS485 port (port2, port2-RS485)

About RS485 port, A is “+” signal、B is “-” signal. XL series PLC RS485 port is put outside. SG terminal is signal ground. The terminal diagram is shown as below:

Please use twisted pair cable for RS485. (See below diagram). But shielded twisted pair cable is better and the single-ended connects to the ground.

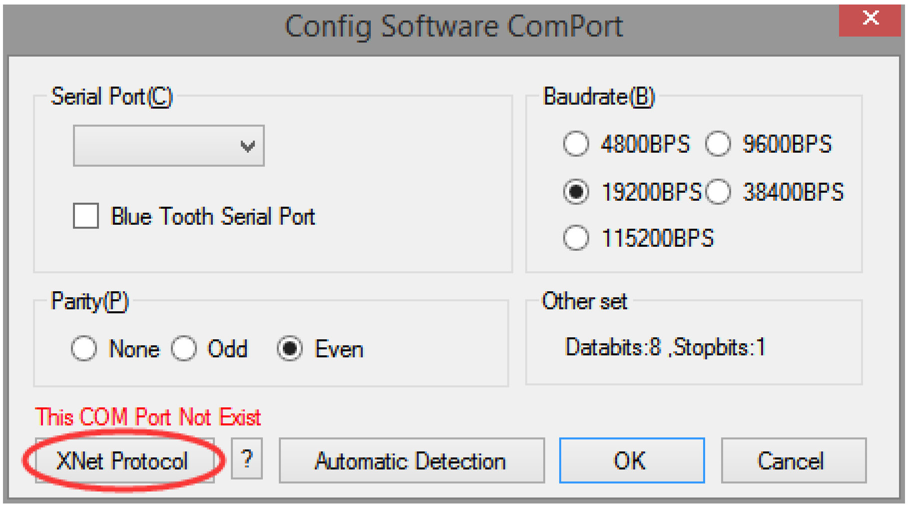


3. USB port

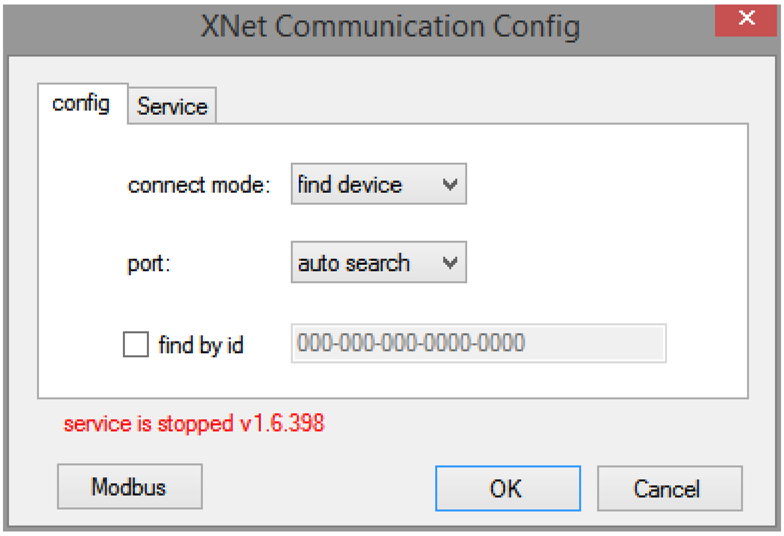
When downloading programs and data through the USB port, the USB driver and XINJEConfig tool must be installed first. Because the current USB driver has been built in the XINJEConfig software, the USB driver will be installed automatically after the XINJEConfig software is installed.

After installing the xinje config tool and usb driver, please switch to Xnet mode in the PLC software:

(1) Open XDPPro software, click option/software serial port config

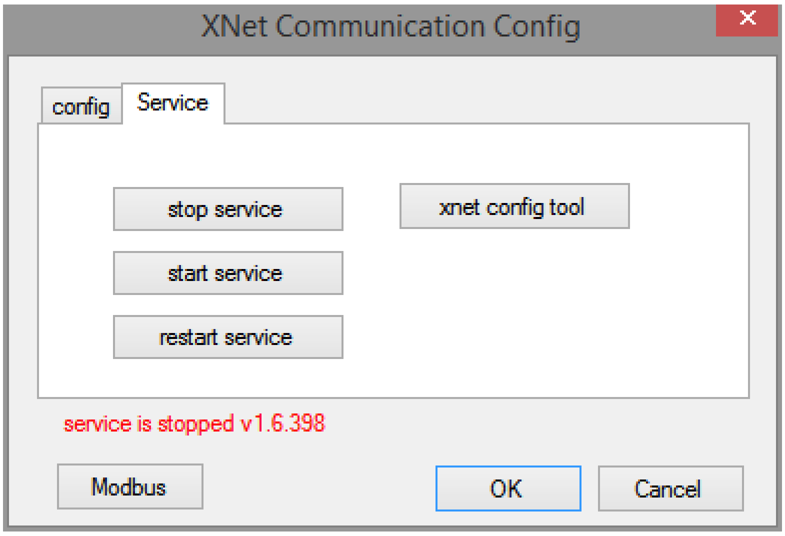


(2) Click Xnet protocol to switch to xnet mode. Then click ok to confirm.



Note:

(1) If it shows the error "find device: error2 cannot find device", you can click "Restart Service" to try to reconnect, or restart the programming software and PLC to reconnect. If you still can't connect, you need to check whether the PLC is power on, whether the USB download cable is connected properly, whether the USB driver and XINJEConfig software are installed properly.

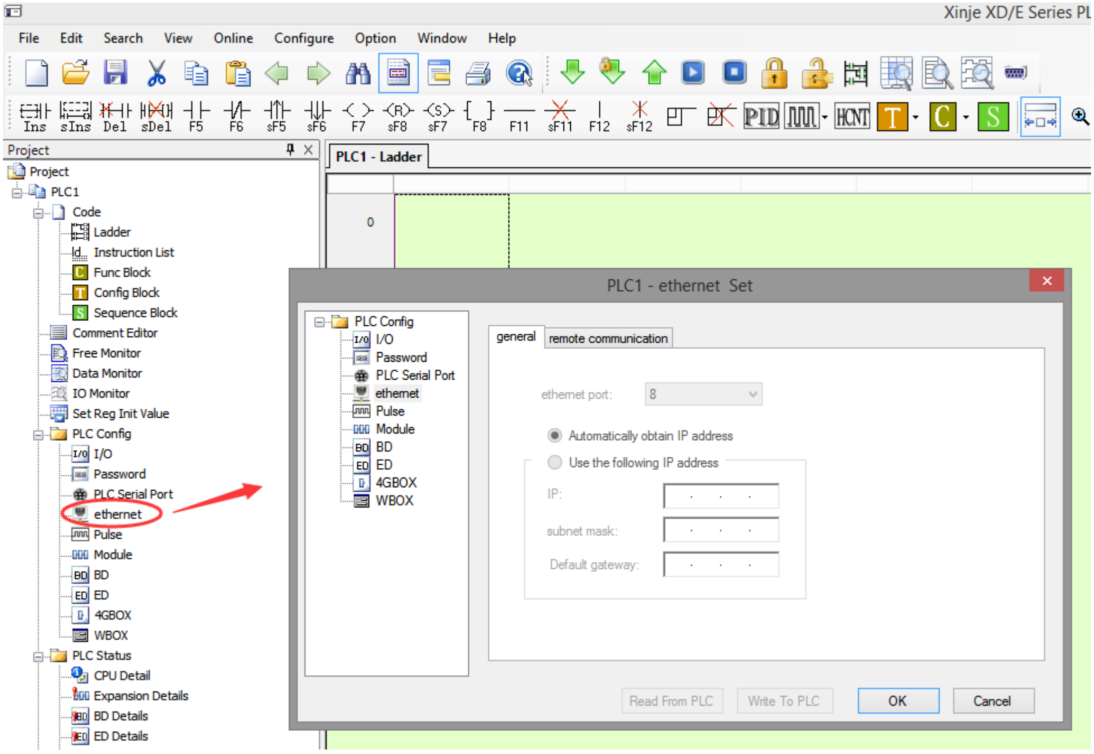


4. **Ethernet port (RJ45)**

RJ45 port is unique for Ethernet PLC, supports TCP/IP Ethernet communication, the port is faster and more stable than USB communication, the data monitoring real-time ability is better, program downloading and uploading is faster. The connection mode of Ethernet communication itself has obvious advantages over RS485 and USB. In many situations of PLC communication, users can communicate with any PLC on the spot through only one switch.

In addition to its application in LAN, Ethernet also supports the remote search, monitoring and operation of PLC, download functions, and communication with other TCP IP devices in the network through the Internet.

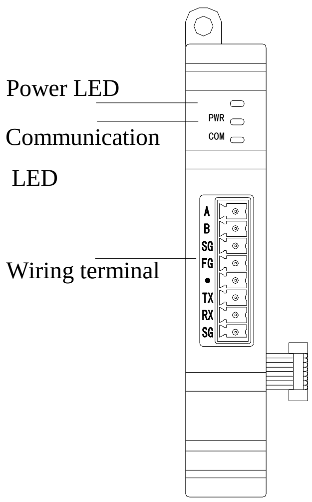
RJ45 port can be configured in "PLC Config-Ethernet" of XINJE PLC programming software, or through XINJEConfig tool. Refer to the relevant manual for details.



5. Left extension ED port (port3)

The left extension ED port can connect ED card to extend RS232 and RS485 port. The ED models include XD-NES-ED (can extend one RS232 and one RS485 port, but the two cannot communicate at the same time).

XD-NES-ED



Each part name is shown as below:

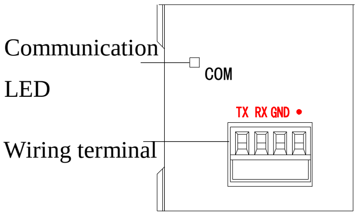
Name		Function
Power LED		The light is ON when the ED module power on
Communication LED		The light is ON when ED module communication is normal
Wiring terminal	A	RS485+
	B	RS485-
	SG	Ground
	FG	Connect to ground terminal
	-	Empty
	TX	RS232 send
	RX	RS232 receive
	SG	Ground

6. Above extension BD port (port4, port5)

The above extension port can connect BD card which contains RS232 mode (XD-NS-BD), RS485 mode (XD-NE-BD) and optical fiber mode (XD-NO-BD).

XD series 24/32 I/O PLC can extend one BD card, XD series 48/60 I/O PLC can extend 2 BD cards, XD series 16 I/O PLC cannot extend BD card.

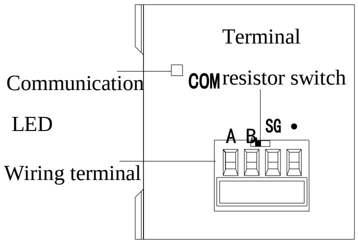
(1) XD-NS-BD



Each part name is shown as below:

Name		Function
Communication LED		Not support this function
Wiring terminal	TX	Signal send
	RX	Signal receive
	GND	Ground
	●	Empty

(2) XD-NE-BD

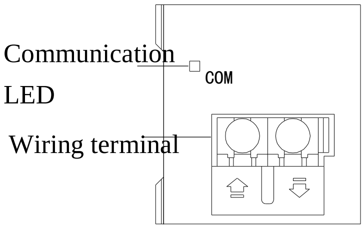


Each part name is shown as below:

Name		Function
Communication LED		The light is flashing when the BD card communication is successful
Wiring terminal	A	485+
	B	485-
	S	Signal ground
	●	Empty
Terminal resistor switch		To choose whether to use terminal resistor (120Ω)

XD-NE-BD has the switch to select whether it is terminal. The switch default setting is OFF which means not install terminal resistor. If XD-NE-BD is at the head or end of the bus, it needs to install 120Ω terminal resistor at the both side and turn on the switch (right).

(3) XD-NO-BD



Each part name is shown as below:

Name	Function
Communication LED	Not support this function
Wiring terminal	The left side is signal input terminal, the right side is signal output terminal

6-1-2. Communication parameters

Communication Parameters

Station	Modbus station number: 1~254
Baud Rate	300bps~9Mbps
Data Bit	5, 6, 7, 8, 9
Stop Bit	1, 1.5, 2
Parity	Even, Odd, even, empty, mask

The default parameters: Station number is 1, baud rate is 19200bps, 8 data bits, 1 stop bit, even parity.
There are many ways to set the parameters of PLC communication port:
There are two ways to set Modbus communication parameters: (1) setting parameters by programming software; (2) setting parameters by XINJEConfig tool, refer to chapter 6-2-6 for details.
Free format communication parameters can be set by programming software, refer to chapter 6-3-2 for details.
X-NET communication parameters can be set by Xinje Config tool. Refer to X-NET fieldbus manual for details.

Note: For the A, B terminal on the PLC body, 1Mbps and higher baud rate is only fit for X-NET communication mode.

6-2. MODBUS communication

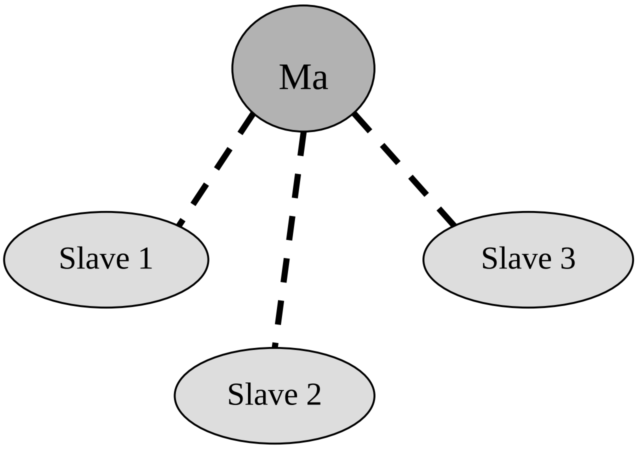
6-2-1. Function overview

XD, XL series PLC support both Modbus master and Modbus slave.

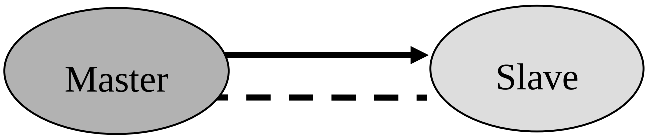
Master mode: When PLC is set to be master, it can communicate with other slave devices which have MODBUS-RTU or MODBUS-ASCII protocol via Modbus instructions; it also can change data with other devices.
For example: Xinje XD3 series PLC can control inverter by Modbus.

Slave mode: When PLC is set to be slave, it can only response with other master devices.

Master and slave: In RS485 network, there can be one master and several slaves at one time (see below diagram). The master station can read and write any slave station. Two slave stations cannot communicate with each other. Master station should write program and read or write one slave station; slave station has no program but only response the master station. （Wiring：connect all 485+, connect all 485-）



In RS232 network (see below diagram), there can only be one master and one slave at one time.



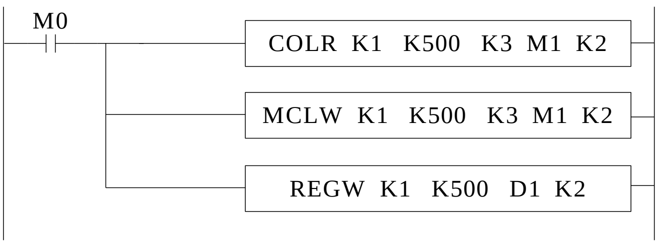
There is dotted line in the diagram. It means any PLC can be master station when all PLC in the network don't send data. As the PLC do not have unified clock standard, communication will fail when more than one PLC send data at one time. It is not recommended to use.

Note:

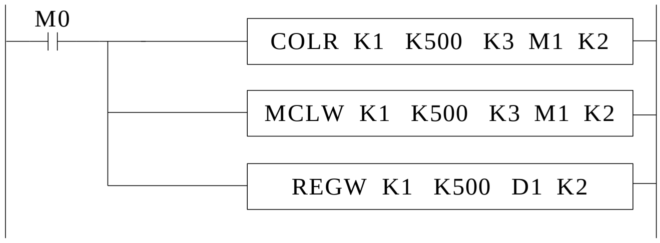
1. For XD/XL series PLC, RS232 and RS485 only support half-duplex.
2. For XC series PLC, if master PLC send one data to slave PLC, and master PLC send data again before slave PLC receiving the last one completely, slave PLC end data error may occur; For XD/XL series PLC, we solve this problem by adding waiting time before communication, which means the slave PLC will receive the next data only after some time the last data finished.

6-2-2．Changing of Modbus instruction

Modbus instruction handling mode has changed in XD/XL series PLC, users can write Modbus instructions directly in program, the protocol station will queue up Modbus requests, which is not the same task with communication; It means users can use one triggering condition to trigger multiple Modbus instructions at the same time. PLC will queue up Modbus requests according to protocol station, which will lead to communication error in XC series PLC.



XC series（×）



XD3 series（√）

Note: XD/XL series PLC sequence block has cancelled Modbus communication instructions, which is replaced by the current Modbus instruction handling mode.

6-2-3. Modbus communication address

The soft component's code in PLC corresponds with Modbus ID number, please see the following table:

XD1, XD2, XD3, XL1, XL3 series PLC Modbus address and internal soft component table:

type	component	Address	number	Modbus address (Hex)	Modbus address (decimal)
Coil bit	M	M0~M7999	8000	0~1F3F	0~7999
	X	X0~X77 (main unit)	64	5000~503F	20480~20543
		X10000~X10077 (#1 module)	64	5100~513F	20736~20799
		X10100~X10177 (#2 module)	64	5140~517F	20800~20863
		X10200~X10277 (#3 module)	64	5180~51BF	20864~20927
		X10300~X10377 (#4 module)	64	51C0~51FF	20928~20991
		X10400~X10477 (#5 module)	64	5200~523F	20992~21055
		X10500~X10577 (#6 module)	64	5240~527F	21056~21119
		X10600~X10677 (#7 module)	64	5280~52BF	21120~21183
		X10700~X10777 (#8 module)	64	52C0~52FF	21184~21247
		X11000~X11077 (#9 module)	64	5300~533F	21248~21311
		X11100~X11177 (#10 module)	64	5340~537F	21312~21375
		X20000~X20077(#1 BD)	64	58D0~590F	22736~22799
	Y	Y0~77(main unit)	64	6000~603F	24576~24639
		Y10000~Y10077 (#1 module)	64	6100~613F	24832~24895

		Y10100~Y10177 (#2 module)	64	6140~617F	24896~24959
		Y10200~Y10277 (#3 module)	64	6180~61BF	24960~25023
		Y10300~Y10377 (#4 module)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (#5 module)	64	6200~623F	25088~25151
		Y10500~Y10577 (#6 module)	64	6240~627F	25152~25215
		Y10600~Y10677 (#7 module)	64	6280~62BF	25216~25279
		Y10700~Y10777 (#8 module)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (#9 module)	64	6300~633F	25344~25407
		Y11100~Y11177 (#10 module)	64	6340~637F	25408~25471
		Y20000~Y20077(#1 BD)	64	68D0~690F	26832~26895
	S	S0~S1023	1024	7000~73FF	28672~29695
	SM	SM0~SM2047	2048	9000~97FF	36864~38911
	T	T0~T575	576	A000~A23F	40960~41535
	C	C0~C575	576	B000~B23F	45056~45631
	ET	ET0~ET31	32	C000~C01F	49152~49183
	SEM	SEM0~SEM31	32	C080~C09F	49280~49311
	HM※1	HM0~HM959	960	C100~C4BF	49408~50367
	HS※1	HS0~HS127	128	D900~D97F	55552~55679
	HT※1	HT0~HT95	96	E100~E15F	57600~57695
	HC※1	HC0~HC95	96	E500~E55F	58624~58719
	HSC※1	HSC0~HSC31	32	E900~E91F	59648~59679
Register word	D	D0~D7999	8000	0~1F3F	0~7999
	ID	ID0~ID99(main unit)	100	5000~5063	20480~20579
		ID10000~ID10099 (#1 module)	100	5100~5163	20736~20835
		ID10100~ID10199 (#2 module)	100	5164~51C7	20836~20935
		ID10200~ID10299 (#3 module)	100	51C8~522B	20936~21035
		ID10300~ID10399 (#4 module)	100	522C~528F	21036~21135
		ID10400~ID10499 (#5 module)	100	5290~52F3	21136~21235
		ID10500~ID10599 (#6 module)	100	52F4~5357	21236~21335
		ID10600~ID10699 (#7 module)	100	5358~53BB	21336~21435
		ID10700~ID10799 (#8 module)	100	53BC~541F	21436~21535
		ID10800~ID10899 (#9 module)	100	5420~5483	21536~21635

		ID10900~ID10999 (#10 module)	100	5484~54E7	21636~21735
		ID20000~ID20099 (#1 BD)	100	58D0~5933	22736~22835
	QD	QD0~QD99(main unit)	100	6000~6063	24576~24675
		QD10000~QD10099 (#1 module)	100	6100~6163	24832~24931
		QD10100~QD10199 (#2 module)	100	6164~61C7	24932~25031
		QD10200~QD10299 (#3 module)	100	61C8~622B	25032~25131
		QD10300~QD10399 (#4 module)	100	622C~628F	25132~25231
		QD10400~QD10499 (#5 module)	100	6290~62F3	25232~25331
		QD10500~QD10599 (#6 module)	100	62F4~6357	25332~25431
		QD10600~QD10699 (#7 module)	100	6358~63BB	25432~25531
		QD10700~QD10799 (#8 module)	100	63BC~641F	25532~25631
		QD10800~QD10899 (#9 module)	100	6420~6483	25632~25731
		QD10900~QD10999 (#10 module)	100	6484~64E7	25732~25831
		QD20000~QD20099 (#1 BD)	100	68D0~6933	26832~26931
	SD	SD0~SD2047	2048	7000~77FF	28672~30719
	TD	TD0~TD575	576	8000~823F	32768~33343
	CD	CD0~CD575	576	9000~923F	36864~37439
	ETD	ETD0~ETD31	32	A000~A01F	40960~40991
	HD※1	HD0~HD999	1000	A080~A467	41088~42087
	HSD※1	HSD0~HSD499	500	B880~BA73	47232~47731
	HTD※1	HTD0~HTD95	96	BC80~BCDF	48256~48351
	HCD※1	HCD0~HCD95	96	C080~C0DF	49280~49375
	HSCD※1	HSCD0~HSCD31	32	C480~C49F	50304~50335
	FD※2	FD0~FD5119	5120	C4C0~D8BF	50368~55487
	SFD※2	SFD0~SFD1999	2000	E4C0~EC8F	58560~60559
	FS※2	FS0~FS47	48	F4C0~F4EF	62656~62703

XD5, XDM, XDC, XD5E, XDME, XL5, XL5E, XLME series PLC Modbus address and internal soft component table:

Type	component	Address	numbers	Modbus address (hex)	Modbus address (decimal)
Coil bit	M	M0~M20479	20480	0~4FFFF	0~20479
	X	X0~X77(main unit)	64	5000~503F	20480~20543
		X10000~X10077 (#1 module)	64	5100~513F	20736~20799

		X10100~X10177 (#2 module)	64	5140~517F	20800~20863
		X10200~X10277 (#3 module)	64	5180~51BF	20864~20927
		X10300~X10377 (#4 module)	64	51C0~51FF	20928~20991
		X10400~X10477 (#5 module)	64	5200~523F	20992~21055
		X10500~X10577 (#6 module)	64	5240~527F	21056~21119
		X10600~X10677 (#7 module)	64	5280~52BF	21120~21183
		X10700~X10777 (#8 module)	64	52C0~52FF	21184~21247
		X11000~X11077 (#9 module)	64	5300~533F	21248~21311
		X11100~X11177 (#10 module)	64	5340~537F	21312~21375
		X11200~X11277 (#11 module)	64	5380~53BF	21376~21439
		X11300~X11377 (#12 module)	64	53C0~53FF	21440~21503
		X11400~X11477 (#13 module)	64	5400~543F	21504~21567
		X11500~X11577 (#14 module)	64	5440~547F	21568~21631
		X11600~X11677 (#15 module)	64	5480~54BF	21632~21695
		X11700~X11777 (#16 module)	64	54C0~54FF	21696~21759
		X20000~X20077 (#1 BD)	64	58D0~590F	22736~22799
	Y	Y0~77(main unit)	64	6000~603F	24576~24639
		Y10000~Y10077 (#1 module)	640	6100~613F	24832~24895
		Y10100~Y10177 (#2 module)	64	6140~617F	24896~24959
		Y10200~Y10277 (#3 module)	64	6180~61BF	24960~25023
		Y10300~Y10377 (#4 module)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (#5 module)	64	6200~623F	25088~25151
		Y10500~Y10577 (#6 module)	64	6240~627F	25152~25215
		Y10600~Y10677 (#7 module)	64	6280~62BF	25216~25279
		Y10700~Y10777 (#8 module)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (#9 module)	64	6300~633F	25344~25407
		Y11100~Y11177 (#10 module)	64	6340~637F	25408~25471

		Y11200~Y11277 (#11 module)	64	6380~63BF	25472~25535
		Y11300~Y11377 (#12 module)	64	63C0~63FF	25536~25599
		Y11400~Y11477 (#13 module)	64	6400~643F	25600~25663
		Y11500~Y11577 (#14 module)	64	6440~647F	25664~25727
		Y11600~Y11677 (#15 module)	64	6480~64BF	25728~25791
		Y11700~Y11777 (#16 module)	64	64C0~64FF	25792~25855
		Y20000~Y20077(#1 BD)	64	68D0~690F	26832~26895
	S	S0~S7999	8000	7000~8F3F	28672~36671
	SM	SM0~SM4095	4096	9000~9FFF	36864~40959
	T	T0~T4095	4096	A000~AFFF	40960~45055
	C	C0~C4095	4096	B000~BFFF	45056~45151
	ET	ET0~ET39	40	C000~C027	49152~49191
	SEM	SEM0~SEM127	128	C080~C0FF	49280~49407
	HM※1	HM0~HM6143	6144	C100~D8FF	49408~55551
	HS※1	HS0~HS999	1000	D900~DCEF	55552~56551
	HT※1	HT0~HT1023	1024	E100~E4FF	57600~58623
	HC※1	HC0~HC1023	1024	E500~E8FF	58624~59647
	HSC※1	HSC0~HSC36	40	E900~E927	59648~59687
Register word	D	D0~D20479	20480	0~4FFF	0~20479
	ID	ID0~ID99(main unit)	100	5000~5063	20480~20579
		ID10000~ID10099 (#1 module)	100	5100~5163	20736~20835
		ID10100~ID10199 (#2 module)	100	5164~51C7	20836~20935
		ID10200~ID10299 (#3 module)	100	51C8~522B	20936~21035
		ID10300~ID10399 (#4 module)	100	522C~528F	21036~21135
		ID10400~ID10499 (#5 module)	100	5290~52F3	21136~21235
		ID10500~ID10599 (#6 module)	100	52F4~5357	21236~21335
		ID10600~ID10699 (#7 module)	100	5358~53BB	21336~21435
		ID10700~ID10799 (#8 module)	100	53BC~541F	21436~21535
		ID10800~ID10899 (#9 module)	100	5420~5483	21536~21635
		ID10900~ID10999 (#10 module)	100	5484~54E7	21636~21735
		ID11000~ID11099 (#11 module)	100	54E8~554B	21736~21835
		ID11100~ID11199 (#12 module)	100	554C~55AF	21836~21935

		ID11200~ID11299 (#13 module)	100	55B0~5613	21936~22035
		ID11300~ID11399 (#14 module)	100	5614~5677	22036~22135
		ID11400~ID11499 (#15 module)	100	5678~56DB	22136~22235
		ID11500~ID11599 (#16 module)	100	56DC~573F	22236~22335
		ID20000~ID20099(#1 BD)	100	58D0~5933	22736~22835
	QD	QD0~QD99(main unit)	100	6000~6063	24576~24675
		QD10000~QD10099 (#1 module)	100	6100~6163	24832~24931
		QD10100~QD10199 (#2 module)	100	6164~61C7	24932~25031
		QD10200~QD10299 (#3 module)	100	61C8~622B	25032~25131
		QD10300~QD10399 (#4 module)	100	622C~628F	25132~25231
		QD10400~QD10499 (#5 module)	100	6290~62F3	25232~25331
		QD10500~QD10599 (#6 module)	100	62F4~6357	25332~25431
		QD10600~QD10699 (#7 module)	100	6358~63BB	25432~25531
		QD10700~QD10799 (#8 module)	100	63BC~641F	25532~25631
		QD10800~QD10899 (#9 module)	100	6420~6483	25632~25731
		QD10900~QD10999 (#10 module)	100	6484~64E7	25732~25831
		QD11000~QD11099 (#11 module)	100	64E8~654B	25832~25931
		QD11100~QD11199 (#12 module)	100	654C~65AF	25932~26031
		QD11200~QD11299 (#13 module)	100	65B0~6613	26032~26131
		QD11300~QD11399 (#14 module)	100	6614~6677	26132~26231
		QD11400~QD11499 (#15 module)	100	6678~66DB	26232~26331
		QD11500~QD11599 (#16 module)	100	66DC~673F	26332~26431
		QD20000~QD20099(#1 BD)	100	68D0~6933	26832~26931
	SD	SD0~SD4095	4096	7000~7FFF	28672~32767
	TD	TD0~TD4095	4096	8000~8FFF	32768~36863
	CD	CD0~CD4095	4096	9000~9FFF	36864~40959
	ETD	ETD0~ETD39	40	A000~A027	40960~40999
	HD※1	HD0~HD6143	6144	A080~B87F	41088~47231
	HSD※1	HSD0~HSD1023	1024	B880~BC7F	47232~48255
	HTD※1	HTD0~HTD1023	1024	BC80~C07F	48256~49279

	HCD[※1]	HCD0~HCD1023	1024	C080~C47F	49280~40303
	HSCD[※1]	HSCD0~HSCD39	40	C480~C4A7	50304~50343
	FD[※2]	FD0~FD8191	8192	C4C0~E4BF	50368~58559
	SFD[※2]	SFD0~SFD5999	6000	E4C0~FC2F	58560~64559
	FS[※2]	FS0~FS47	48	F4C0~F4EF	62656~62703

Note:

1. *1 is power-off retentive range, *2 is flash range.
2. The address is usually for Modbus-RTU and Modbus-ASCII communication when PLC works as lower computer, and upper computer: SCADA/screen/PLC......
3. If upper computer is PLC, then we write program according to Modbus-RTU or Modbus-ASCII protocol; if upper computer is SCADA or HMI, there will be two situations: 1. with xinje driver. E.g.: xinje HMI can use PLC soft components directly (Y0/M0). 2. without xinje driver. Please select Modbus-RTU or Modbus-ASCII protocol, then use the address in the above table to define the data vairable.
4. For Octonary I/O, calculate corresponding octonary I/O Modbus address. For example, Y0 modbus address is H6000, Y10 modbus address is H6008 (not H6010), Y20 modbus address is H6016 (not H6020).
5. when the modbus address is over K32767, it needs to use hex format to show it and add 0 before the address. For example, HD0 Modbus address is 41088 which cannot write in the software, please convert it to hex format H0A080.

6-2-4 Modbus data format

Modbus transmission mode:

There are two transmission modes: RTU and ASCII; It defines serial transmission of bit content in message domain; it decides how information to pack and decode; transmission mode (and port parameters) of all devices in Modbus serial links should be the same.

Modbus-RTU data structure

RTU mode:

Under Modbus RTU (remote terminal unit) mode, message has two 4-bit hexadecimal characters in every 8-bit byte. This mode has very high data density, higher throughput rate than Modbus ASCII. Every message should be sent by continuous characters.

RTU mode frame check domain: cycle redundancy check（CRC）.

RTU mode frame description:

Modbus station	Function code	data	CRC	
1 byte	1 byte	0~252 byte	2 byte	
			CRC low	CRC high

Format:

START	No input signal ≧ 10ms
Address（station no.）	Communication address: 8-bit binary
Function	Function code：8-bit binary
DATA（n - 1）	Data content：
……	

DATA 0	N*8-bit data, N≤8, max 8 bytes
CRC CHK Low	CRC check code
CRC CHK High	16-bit CRC check code is consist of two 8-bit binary
END	No input signal ≧ 10ms

2. Modbus address:

00H: All the Xinje XC series PLC broadcast—— slave stations don't response.

01H: Communicate with address 01H PLC.

0FH: Communicate with address 15H PLC.

10H: Communicate with address 16H PLC and so on. Up to 254（FEH）.

3. Function and DATA:

Function code	Function	Modbus instruction
01H	Read coil	COLR
02H	Read input coil	INPR(not support Xinje PLC)
03H	Read register	REGR
04H	Read input register	INRR
05H	Write coil	COLW
06H	Write register	REGW
10H	Write multi-register	MRGW
0FH	Write multi-coil	MCLW

(1) Take 06H function code as example (single register write), and introduce data format.

E.g.: upper computer write data to PLC H0002 (D2).

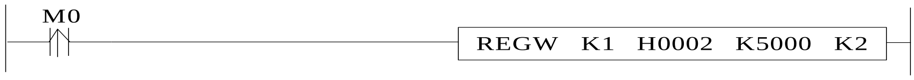
RTU mode:

Asking format		Response format	
ID	01H	ID	01H
Function code	06H	Function code	06H
Register ID	00H	Register ID	00H
	02H		02H
Data content	13H	Data contents	13H
	88H		88H
CRC CHECK High	25H	CRC CHECK High	25H
CRC CHECK Low	5CH	CRC CHECK Low	5CH

Explanation:

1. Address is PLC station no.
2. Function code is Modbus-RTU protocol read/write code.
3. Register address is the PLC modbus address, please see chapter 6-2-3.
4. Data content is the value in D2.
5. CRC CHECK High / CRC CHECK Low is high and low bit of CRC check value.

If 2 pieces of Xinje XD3 series PLC communicate with the other one, write K5000 to D2.



M0 is trigger condition (Rising edge). If communication fails, the instruction will try twice. If the third time communication fails, then communication ends.

The relationship between REGW and Modbus RTU protocol (other instructions are the same)

REGW	Function code 06H
K1	Station no.
H0002	Modbus address
K5000	Data contents 1388H
K2	PLC serial port

The complete communication datum are: 01H 06H 00H 02H 13H 88H (system take CRC checking automatically)

If monitor the serial port2 data by serial port debugging tool, the datum are: 01 06 00 02 13 88 25 5C

Note: The instruction doesn’t distinguish decimal, hex, binary, octal etc. For example, B10000, K16 and H10 are the same value, so the following instructions are the same.

REGW K1 B111110100 D1 K2

REGW K1 K500 D1 K2

REGW K1 H1F4 D1 K2

(2) Function code 01H/02H: read coil/read input coil

Eg. Read coil address 6000H (Y0). At this time, Y0 and Y1 are ON.

RTU mode:

Asking format		**Response format**	
Address	01H	Address	01H
Function code	01H/02H	Function code	01H/02H
Coil address	60H	Byte number	01H
	00H		
Coil number	00H	Data contents	03H
	02H		
CRC CHECK Low	A3H	CRC CHECK Low	11H
CRC CHECK High	CBH	CRC CHECK High	89H

As the status of Y0 and Y1 is ON, the data contents are 03H (0000 0011).

(3) Function code 03H: read register

Eg. Read two register starting from 03E8H (D1000, D1001).

RTU mode:

Asking format		**Response format**	
Address	01H	Address	01H
Function code	03H	Function code	03H
Register address	03H	Byte number	04H
	E8H		
Register number	00H	Data contents	12H
			2EH

	02H		04H
			E8H
CRC CHECK Low	44H	CRC CHECK Low	9DH
CRC CHECK High	7BH	CRC CHECK High	CCH

At this time, the data read from D1000 and D1001 are 122EH (4654) and 04E8H (1256).

(4) Function code 05H: write single coil

Eg. Set on the coil address 6000H (Y0).

RTU mode:

Asking format		Response format	
Address	01H	Address	01H
Function code	05H	Function code	05H
Coil address	60H	Coil address	60H
	00H		00H
Data contents (low byte is before high byte)	FFH	Data contents	FFH
	00H		00H
CRC CHECK Low	92H	CRC CHECK Low	92H
CRC CHECK High	3AH	CRC CHECK High	3AH

Note: when writing single coil, ON is 00FFH, OFF is 0000H; the low byte is before high byte for the data contents.

(5) Function code 0FH: write multiple coils

Eg. Write 16 coils start from address 6000H (Y0).

RTU mode:

Asking format		Response format	
Address	01H	Address	01H
Function code	0FH	Function code	0FH
Coil address	60H	Coil address	60H
	00H		00H
Coil number	00H	Coil number	00H
	10H		10H
Byte number	02H	-	-
Data contents (low byte is before high byte)	03H		
	01H		
CRC CHECK Low	43H	CRC CHECK Low	4AH
CRC CHECK High	16H	CRC CHECK High	07H

The data contents are 0103H, the binary format is 0000 0001 0000 0011, write in corresponding Y17~Y0, so Y0, Y1, Y10 are set ON.

Note: when writing the data contents, the low byte is before the high byte.

(6) Function code 10H: write multiple registers

Eg. Write 3 registers starting from address 0000H (D0).

RTU mode:

Asking format		**Response format**	
Address	01H	Address	01H
Function code	10H	Function code	10H
Register address	00H	Register address	00H
	00H		00H
Register number	00H	Register number	00H
	03H		03H
Byte number	06H	-	-
Data contents	00H		
	01H		
	00H		
	02H		
	00H		
	03H		
CRC CHECK Low	3AH	CRC CHECK Low	3AH
CRC CHECK High	81H	CRC CHECK High	81H

After executing, the value in D0, D1, D2 are 1, 2, 3.

Note: byte number = register number * 2.

Modbus-ASCII data structure

ASCII mode:

For Modbus ASCII（American Standard Code for Information Interchange）mode in serial links, every 8-bit byte is sent as two ASCII characters. When communication links and devices do not fit RTU mode timing monitor, we usually use the ASCII mode.

Note：One byte needs two characters, so ASCII mode has lower inefficiency than RTU mode.

E.g.: Byte 0X5B will be encoded as two characters: 0x35 and 0x42（ASCII code 0x35 ="5"， 0x42 ="B" ）.

ASCII mode frame check domain：Longitudinal Redundancy Checking (LRC)

ASCII mode frame description:

Start mark	Modbus no.	Function code	data	LRC	End mark	
1 character	2 characters	2 characters	0~252*2 characters	2 characters	2 characters	
0x3A					0x0D	0x0A

Format：

STX (3AH)	Start mark=3AH
Address code high bit	Communication position（no）：Consist of 2 ASCII codes
Address code low bit	
Function code high bit	Function code（command）：Consist of 2 ASCII codes
Function code low bit	
Instruction start ID	Command start bit：
Instruction start ID	

Instruction start ID	Consist of 4 ASCII codes
Instruction start ID	
Data length	Length from start to end： Consist of 4 ASCII codes
Data length	
Data length	
Data length	
LRC check high bit	LRC check code： Consist of 2 ASCII codes
LRC check low bit	
END high bit	End mark： END Hi=CR（0DH），END Lo=CR（0AH）
END low bit	

2. Communication address：

00H：All Xinje XC series PLC broadcast—— slave stations do not response.

01H：Communicate with address 01H PLC.

0FH：Communicate with address 15H PLC.

10H：Communicate with address 16H PLC.

And so on, up to 254（FEH）.

3. Function and DATA：

Function code	Function	Corresponding modbus
01H	Read coil	COLR
02H	Read input coil	INRR
03H	Read register	REGR
04H	Read input register	INRR
05H	Write single coil	COLW
06H	Write single register	REGW
10H	Write multiple registers	MRGW
0FH	Write multiple coils	MCLW

Take 06H function code（write single register）as example, and introduce data format（other functions are similar to this）：

E.g.：upper computer write data K5000(H1388) to PLC H0002 (D2).

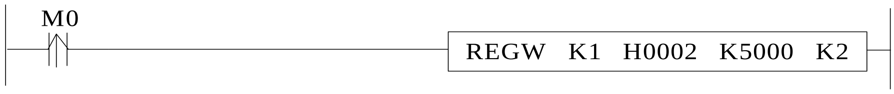
ASCII mode：

Start mark	3AH
ID	30H
	31H
Function code	30H
	36H
Register ID high byte	30H
	30H
Register ID low byte	30H
	32H
Data content high byte	31H
	33H
Data content low byte	38H
	38H

LRC	35H
	43H
End mark	0DH
	0AH

Description：
1. address is PLC station number.
2. Function code is Modbus-ASCII protocol read/write code.
3. Register ID is the PLC modbus communication ID, please see chapter 7-2-2.
4. Data content is the value in D2.
5. LRC CHECK Low / CRC CHECK High is low and high bit of CRC check value.

If two pieces of Xinje XD3 PLC communicate with each other, write K5000 to D2.



M0 is trigger condition (rising edge). When Xinje PLC communicates by Modbus, if communication fails, the instruction will try twice. If the third time communication fails, then communication ends.
The relationship between REGW and ASCII protocol (other instructions are similar to this):

REGW	Function code 06H
K1	Station number
H0002	Modbus ID
K5000	Data content is 1388H
K2	PLC communication serial port

Complete data string: 3AH 30H 31H 30H 36H 30H 30H 30H 32H 31H 33H 38H 38H 35H 43H (system take CRC checking automatically)
If monitor the serial port2 by serial port debugging tool, the datum are: 3AH 30H 31H 30H 36H 30H 30H 30H 32H 31H 33H 38H 38H 35H 43H 0DH 0AH
Note: The data does not distinguish decimal, binary, hexadecimal etc. For example, B10000, K16 and H10 are the same value, so the following instructions are the same.
REGW K1 B111110100 D1 K2
REGW K1 K500 D1 K2
REGW K1 H1F4 D1 K2

6-2-5. Communication Instructions

Modbus instructions include coil read/write, register read/write; below will introduce the details.

Instructions in details:
The operand definition in the instruction:
1. Remote communication station and serial port number.
E.g.： one PLC connects 3 inverters. PLC needs to write and read the parameters of inverter. The inverter station number is 1.2 and 3. So the remote communication number is 1.2 and 3.

2. Remote register/coil start ID number:
Assign remote coil/register number: the start coil/register ID of PLC read and write, it is normally used with ‘assigned coil/register number’.
E.g.：PLC read Xinje inverter’s output frequency (H2103), output current（H2104）, bus voltage（H2105）, then remote register/coil start ID is H2103, assigned coil number is K3.

3. Local receipt/send coil/register address：Coil/register in PLC used to exchange data with lower computer.
E.g.： write coil M0：write M0 status to assigned address in lower computer
Write register D0：write D0 value to assigned address
Read coil M1：read content in lower computer assigned address to M1
Read register D1：read content in lower computer assigned address to D1

4. communication condition:
The preconditions of Modbus communication can be normal open/closed coil and rising/falling edge. When the open/close coil triggers, Modbus instructions will always be executed. When the communication between multiple slave stations or the traffic is large, communication delay may occur. The oscillating coil can be used as triggering condition. When the rising/falling edge triggers, Modbus instructions will only be executed once, and only when the next rising/falling edge comes, Modbus instructions will be executed again.

Coil Read [COLR]

Instruction Summary

Read the specified station’s coil status to the local device;

Coil read [COLR]			
16 bits instruction	COLR	32 bits instruction	-
Execution condition	Normally ON/OFF coil	Suitable models	XD, XL
Hardware requirement	-	Software Requirement	-

Operands

Operands	Function	Type
S1	Specify the remote communication station no.	16 bits, BIN
S2	Specify the remote coil start address	16 bits, BIN
S3	Specify the coil quantity	16 bits, BIN
D1	Specify the local coil start address	bits
D2	Specify the serial port no.	16 bits, BIN

Suitable soft components

Word

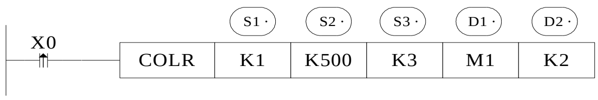
Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●				●		
S2	●	●		●	●				●		
S3	●	●		●	●				●		

Bit

Operands	System						
	X	Y	M*	S*	T*	C*	Dn.m
D1	●	●	●	●	●	●	

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM. M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Function



- Read the coil, Modbus function code 01H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- Operands S3: K1~K2000, the max coil quantity is 2000.
- When X0 is ON, COLR instruction is executed. When the instruction starts to execute, the Modbus read and write flag SM160 (serial port 2) is set on; when the execution is completed, SM160 (serial port 2) is set OFF. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

Input coil read [INPR]

Summary

Read the specified station’s input coil status to local device.

Input coil read[INPR]			
16 bits instruction	INPR	32 bits instruction	-
Execution condition	Normally ON/OFF, rising edge	Suitable models	XD, XL
Hardware requirement	-	Software requirement	-

Operands

Operands	Function	Type
S1	Specify remote communication station no.	16 bits, BIN
S2	Specify remote coil start address number	16 bits, BIN
S3	Specify coil number	16 bits, BIN
D1	Specify start address number of local receipt coils	bit
D2	Specify serial port number	16 bits, BIN

Suitable soft components

Word

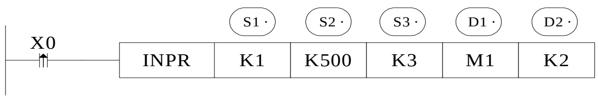
Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●				●		
S2	●	●		●	●				●		
S3	●	●		●	●				●		
D2									K		

Bit

Operands	System						
	X	Y	M*	S*	T*	C*	Dn.m
D1	●	●	●	●	●	●	

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS. M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Function



- Read input coil, Modbus function code is 02H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- Operand S3: K1~K2000, max input coil number is 2008.
- When X0 is ON, INPR instruction is executed, Modbus read write flag SM160(serial port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.
- This instruction cannot read XINJE PLC input coil.

Single Coil Write [COLW]

Summary

Write local device specified coil to remote station no’s coil.

Single Coil write [COLW]			
16 bits instruction	COLW	32 bits instruction	-
Execution Condition	Normally ON/OFF, edge triggering	Suitable Models	XD, XL
Hardware Requirement	-	Software Requirement	-

Operands

Operands	Function	Type
D1	Specify remote communication station number	16 bits， BIN
D2	Specify remote coil start address	16 bits， BIN
S1	Specify start address of local coil	bit
S2	Specify serial port number	16 bits， BIN

Suitable soft components

Word

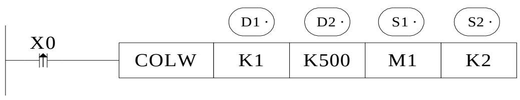
Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D1	●	●		●	●				●		
D2	●	●		●	●				●		
S2									K		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S1	●	●	●	●	●	●	

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS. M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Function



- Write single coil, Modbus function code is 05H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- When X0 is ON, COLW instruction is executed, Modbus read write flag SM160(serial

port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

Multiple coils write [MCLW]

Summary

Write local device multiple coils to remote station no's coil.

Multiple coils write [MCLW]			
16 bits instruction	MCLW	32 bits instruction	-
Execution Condition	Normally ON/OFF, edge triggering	Suitable models	XD, XL
Hardware Requirement	-	Software Requirement	-

Operands

Operands	Function	Type
D1	Specify remote communication station number	16 bits，BIN
D2	Specify remote coil start address	16 bits，BIN
D3	Specify coil number	16 bits，BIN
S1	Specify start address of local coils	bit
S2	Specify serial port number	16 bits，BIN

Suitable soft components

Word

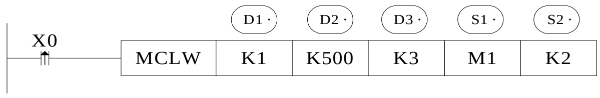
Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D1	●	●	●	●					●		
D2	●	●	●	●					●		
D3	●	●	●	●					●		
S2									K		

Bit

Operands	System						
	X	Y	M*	S*	T*	C*	Dn.m
S1	●	●	●	●	●	●	

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS. M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Function



- Write multiple coils, Modbus function code is 0FH.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- Operand D3: max coil number is 1976.
- When X0 is ON, MCLW instruction is executed, Modbus read write flag SM160(serial port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

Register read [REGR]

Summary

Read remote station no's register to local device.

Register read[REGR]			
16 bits instruction	REGR	32 bits instruction	-
Execution Condition	Normally ON/OFF, edge triggering	Suitable models	XD, XL
Hardware Requirement	-	Software Requirement	-

Operands

Operands	Function	Type
S1	Specify remote communication station number	16 bits，BIN
S2	Specify remote register start address	16 bits，BIN
S3	Specify register number	16 bits，BIN
D1	Specify start address of local register	16 bits，BIN
D2	Specify serial port number	16 bits，BIN

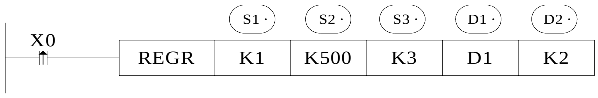
Suitable soft components

Word

Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●					●		
S2	●		●	●					●		
S3	●	●	●	●					●		
D1	●										
D2									K		

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Function



- Read register, Modbus function code is 03H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- Operand S3: max register number is 125.
- When X0 is ON, REGR instruction is executed, Modbus read write flag SM160(serial port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

Input register read [INRR]

Summary

Read remote station no’s input register to local device.

Input register read [INRR]			
16 bits instruction	INRR	32 bits instruction	-
Execution Condition	Normally ON/OFF, edge triggering	Suitable models	XD, XL
Hardware Requirement	-	Software Requirement	-

Operands

Operands	Function	Type
S1	Specify remote communication station number	16 bits，BIN
S2	Specify remote register start address	16 bits，BIN
S3	Specify register number	16 bits，BIN
D1	Specify start address of local register	16 bits，BIN
D2	Specify serial port number	16 bits，BIN

suitable soft components

Word

Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●	●	●					●		
S2	●	●	●	●					●		
S3	●	●	●	●					●		
D1	●										
D2									K		

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Function

		S1·	S2·	S3·	D1·	D2·
X0	INRR	K1	K500	K3	D1	K2

- Read input register, Modbus function code is 04H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- Operand S3: max register number is 125.
- When X0 is ON, INRR instruction is executed, Modbus read write flag SM160(serial port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

Single Register write [REGW]

summary

Write local device register to specified remote station no’s register.

Register write[REGW]			
16 bits instruction	REGW	32 bits instruction	-
Execution Condition	Normally ON/OFF, edge triggering	Suitable models	XD, XL
Hardware Requirement	-	Software Requirement	-

Operands

Operands	Function	Type
D1	Specify remote communication station number	16 bits，BIN
D2	Specify remote register start address	16 bits，BIN
S1	Specify start address of local register	16 bits，BIN
S2	Specify serial port number	16 bits，BIN

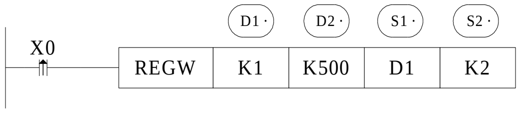
suitable soft components

Word

Operands	System								Constant	Module	
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD
D1	●	●	●	●					●		
D2	●	●	●	●					●		
S1	●										
S2									K		

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Function



- Write register, Modbus function code is 06H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- When X0 is ON, REGW instruction is executed, Modbus read write flag SM160(serial port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

Multiple registers write [MRGW]

Summary

Write local device multiple registers to remote station no’s registers.

Multi-register write [MRGW]			
16 bits instruction	MRGW	32 bits instruction	-
Execution Condition	Normally ON/OFF, edge triggering	Suitable models	XD, XL
Hardware Requirement	-	Software Requirement	-

Operands

Operands	Function	Type
D1	Specify remote communication station number	16 bits，BIN
D2	Specify remote register start address	16 bits，BIN
D3	Specify register number	16 bits，BIN
S1	Specify start address of local registers	16 bits，BIN
S2	Specify serial port number	16 bits，BIN

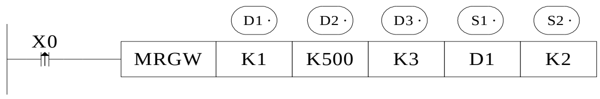
suitable soft components

Word

Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D1	●	●	●	●					●		
D2	●	●	●	●					●		
S1	●										
S2									K		

Notes: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

Function



- Write multiple registers, Modbus function code is 10H.
- Serial port: K0~K5. K0: Port0 (RS232), K1: Port1(RS232), K2: Port2(RS485), K3: Port3(left extension port), K4: Port4(above extension port 1), K5: Port5(above extension port 2).
- Operand D3: the max register number is 123.
- When X0 is ON, MRGW instruction is executed, Modbus read write flag SM160(serial port2) is set ON, SM160 is set OFF when the execution is completed. If a communication error occurs and the number of resend is set, it will be automatically resend. Users can check the relevant registers to determine the cause of the error. The execution result of Modbus read and write instructions of serial port 2 is in SD160.

6-2-6. Modbus serial port configuration

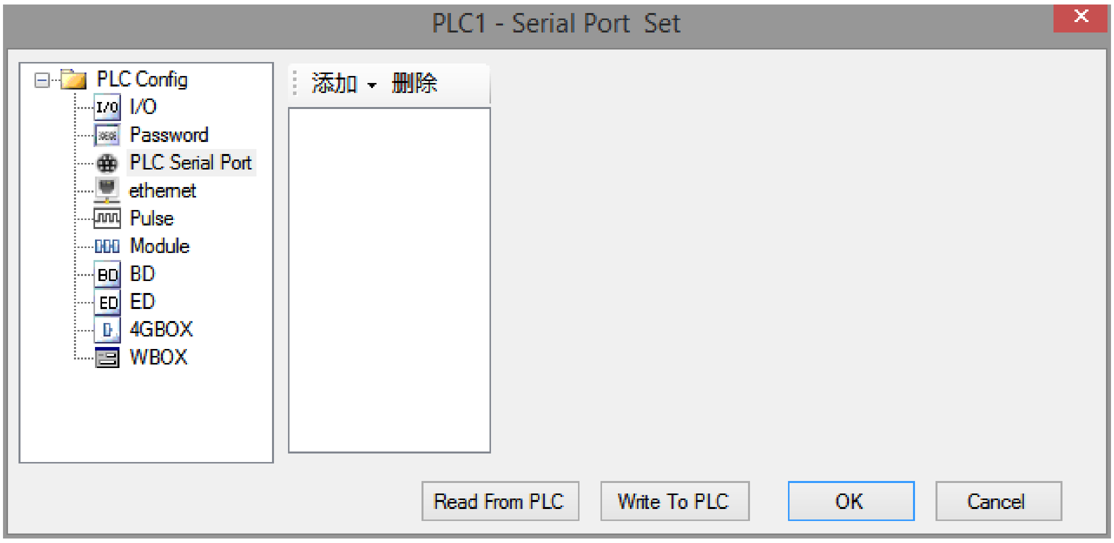
There are two ways to set Modbus communication parameters: 1. setting parameters by programming software; 2. setting parameters by XINJEConfig tool;

1. Set parameters by programming software

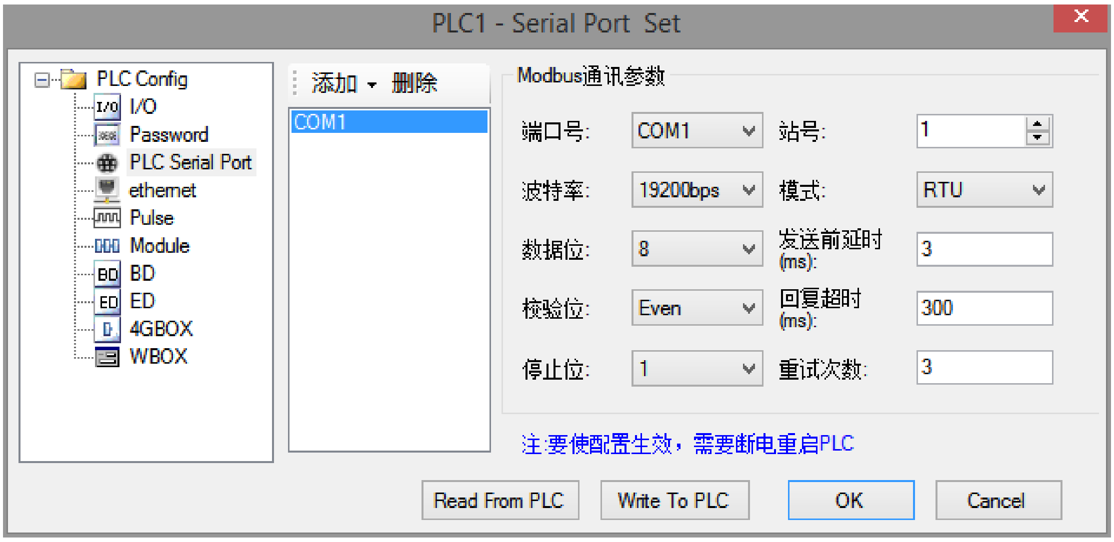
When using programming software to configure the parameters of PLC serial port, the version below V3.4 must use XNET communication mode, and the version above V3.4 can also use Modbus communication mode (RS232 port).

(1) Use the USB download cable to connect the PLC with the computer. Here the USB download cable is the HMI download cable, as shown below, the software must switch to XNet communication mode.

(2) Open the programming software, click configure/PLC comm port settings. It will show below figure:



(3) Click add, it will show two modes, modbus mode and free mode, please select modbus mode, it will show below figure.



Port No.: It refers to Port of PLC, COM0 refers to Port 0 (RS232), COM1 refers to Port 1 (RS232), COM2 refers to Port 2 (RS485) or Port 2-RS232 (RS485) or Port 2-RS485 (RS485), COM3 refers to Port 3 (left extended ED port), COM4 refers to Port 4 (upper extended BD port 1), COM5 refers to Port 5 (upper extended BD port 2).

The baud rate, data bit, parity bit, stop bit should be same to the communication device.

Station number: if the PLC is master, the station no. is defaulted 1, if the PLC is slave, it needs to set different station no.

Two communication modes: RTU, ASCII.

Delay before sending: Waiting time before PLC sends data. In the original XC series PLC, if the master PLC communicates with the slave PLC, the master PLC sends data to the slave PLC. If the master PLC sends data to the slave PLC after the first time, and the slave PLC has not yet had time to receive the data, then the master PLC sends data to the slave PLC again, which easily leads to the error of the slave PLC; In XD series PLC, it has send delay to solve

the problem. That is, after receiving data from the slave station, it must delay a certain time to receive the next communication data, so as not to cause the above problems.
Reply overtime (ms): it refers to the time when the PLC can not receive the response after sending the request and wait for sending again.
Retry times: It refers to the number of times that the PLC can not receive the reply, and each reply needs a reply timeout time.

(4) After setting, click write to PLC, then cut off the PLC power supply and power on again to make the settings effective.

Note: V3.4 version of the XD series of PLC download and upload serial configuration data must use XNET communication mode, that is, using USB port to download and upload configuration data. If the following prompt appears, you need to check whether the serial port parameters you configured are downloaded from the USB port to the PLC.
Note: Versions V3.4 and above can be configured in Modbus communication mode (RS232 port); Versions V3.4 and below XD series PLC must use X-NET communication mode when downloading and uloading serial configuration data, that is, downloading and uploading configuration data through USB port.

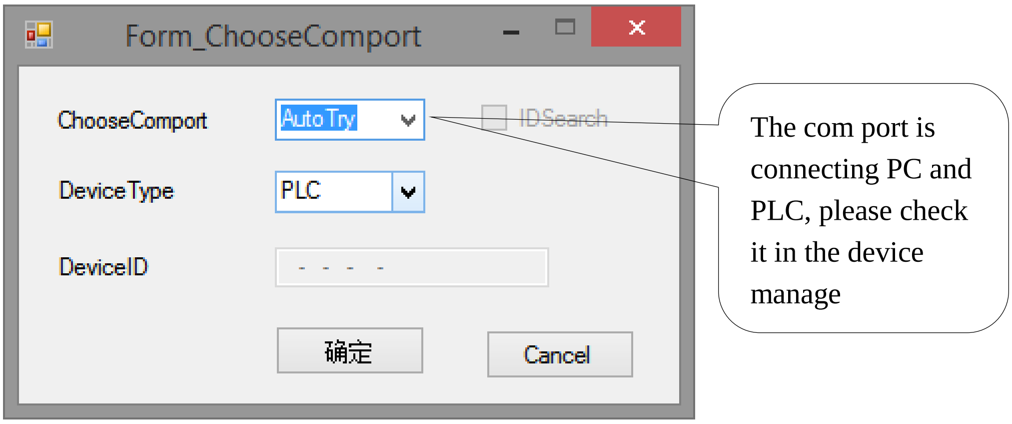
2. Set the parameters by using XINJEConfig tool

When using configuration tool XINJEConfig to configure parameters of PLC serial port, the XINJEConfig tools of V1.6.308 and below must use USB port. The XINJEConfig tool for V1.6.309 and above can also be configured using RS232 port.

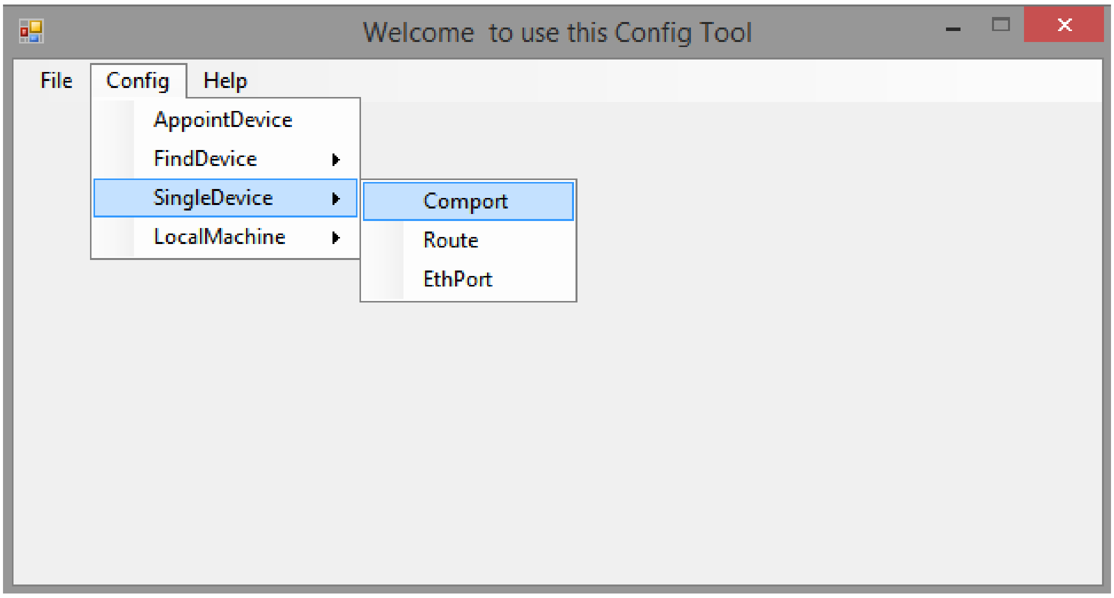
(1) Use the USB download cable to connect the PLC with the computer. Here the USB download cable is the HMI download cable, as shown below.

(2) Open xinjeconfig tool

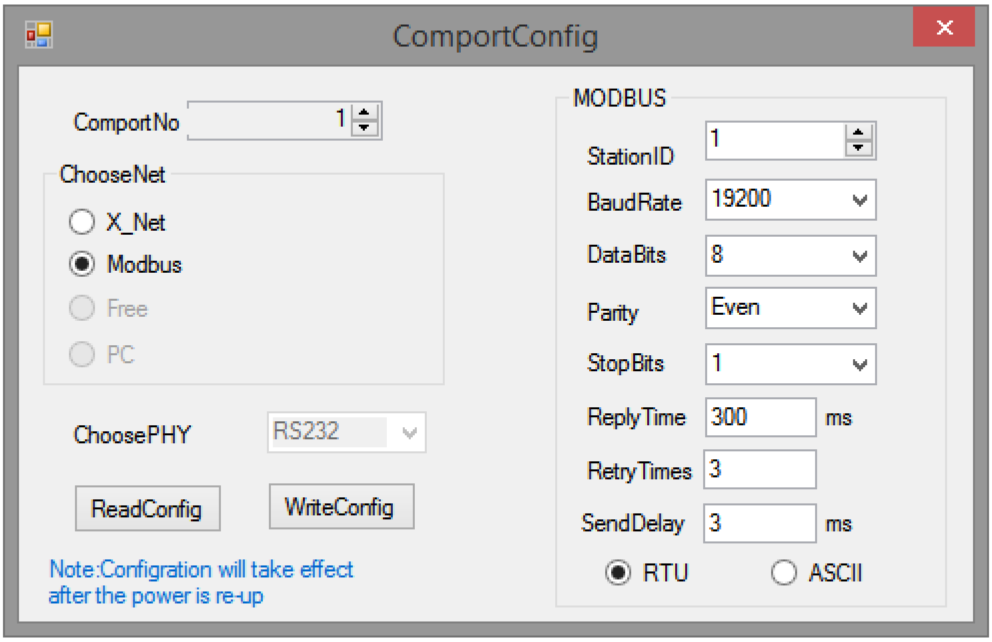
(3) Click config/find device:



(4) Choose the com port connecting PC and PLC, click ok. Click config/single device/comport.

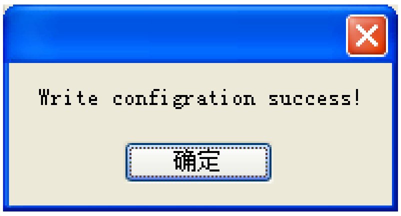


(5) It will show below window.



Serial port: K0 ~ K5. Port0 (RS232), Port1 (RS232), Port2 (RS485) or Port2-RS232 (RS232) or Port2-RS485 (RS485), Port3 (left extension port), Port4 (upper extension port 1), Port5 (upper extension port 2).
Here, we can set the communication mode and parameters of each communication port.

(6) When the com port parameters setting is completed, click writeconfig. It will show “write configuration success” message.

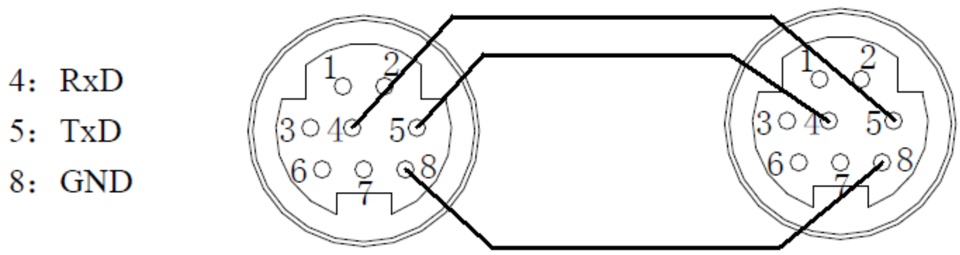


(7) Close XINJEConfig tool, cut the PLC power and power on again to make the settings effective.

6-2-7. Modbus Communication application

Wiring method
There are two wiring methods:
232 wiring methods

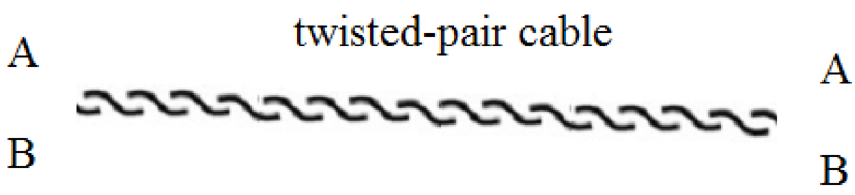
COM2*1 diagram



Mini Din 8 Pins port

Note:
1. COM2 with *1 only show the RS232 pins.
2. XD/XL series PLC，RS232 do not support full-duplex, so it can only communicate in single direction.
3. RS232 communication distance is short (about 13m); RS485 is suitable for longer distance.

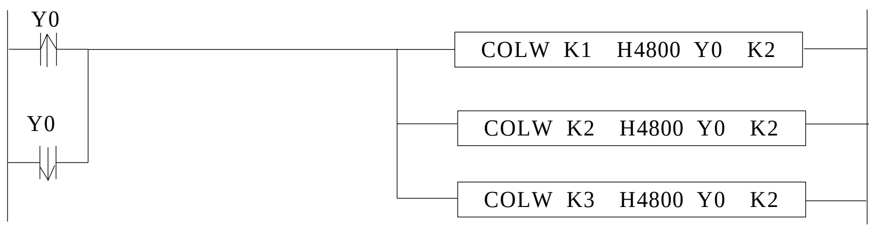
485 wiring methods



Connect all A terminals, connect all B terminals. A is RS485+, B is RS485-.

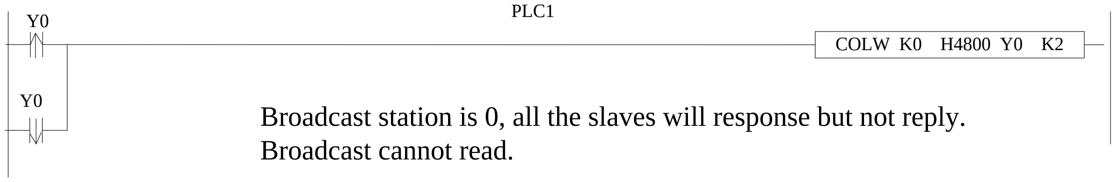
Application：One xinje XD3 series PLC controls 3 XC series PLCs, slave PLCs follow the master’s action. (Master PLC Y0 ON, then slave PLC Y0 ON; Master PLC Y0 OFF, then slave PLC Y0 OFF) Precondition: on-off of Y0 makes communication have enough time to react. Also three slave PLCs can be not that synchronous (not fully synchronous).

Method 1 usual program



The program takes serial port 2 as example, so corresponding communication flag is the serial port 2’s. About other serial port, please refer to appendix 1. Serial port, please refer to appendix 1.

Method 2 use broadcasting function：



When master Y0 status changes, it broadcasts the status to all the slaves. The synchronization of three PLCs is better than method 1.

6-2-8. Application

Example 1:

Following are the programs for reading and writing Modbus communication between 1 master station and 3 slave stations.

Program operation:

(1) Write master PLC Y0~Y11 status to slave PLC 2 Y0~Y11

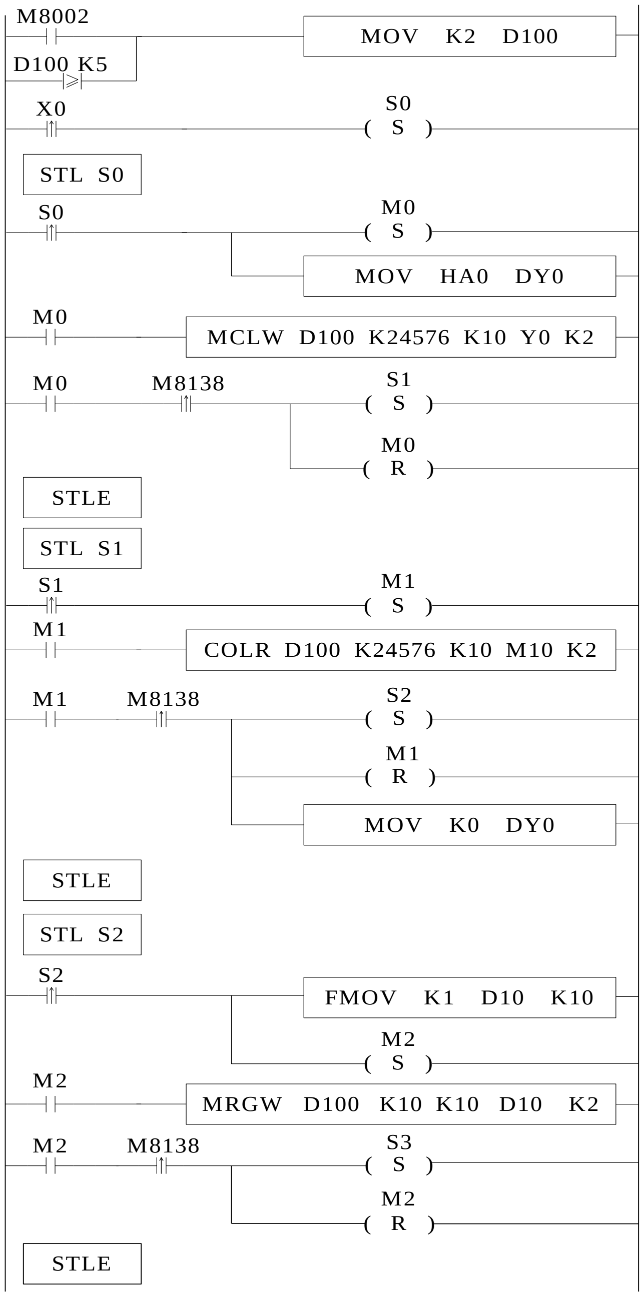
(2) Read slave PLC 2 Y0~Y11 to master PLC M10~M19

(3) Write master PLC D10~D19 to slave PLC 2 D10~D19

(4) Read slave PLC 2 D10~D19 to master PLC D20~D29

(5) So as slave PLC 3 and 4

The following is a comparison of XC and XD series Modbus-RTU communication programs for reference. The communication programs in XC series are as follows:

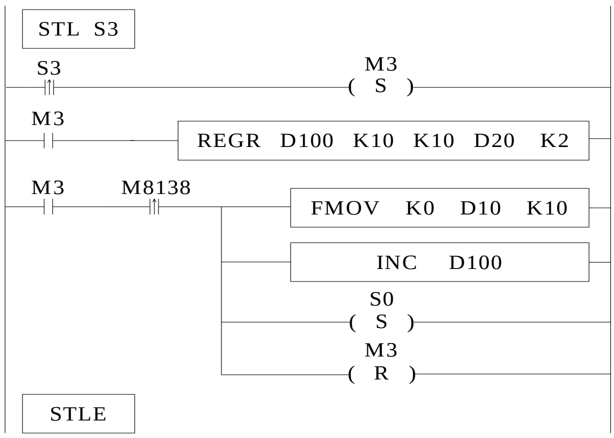


//send station no.2 to D100, execute the process S0

//set ON Y0~Y11 of master station, write the master status to Y0~Y11 of slave PLC 2, 3, 4. Enter process S1 when the communication succeeded.

//read the Y0~Y11 of slave PLC 2, 3, 4 to master PLC M10~M19. Reset master PLC Y0~Y11 and enter process S2 after the communication is successful.

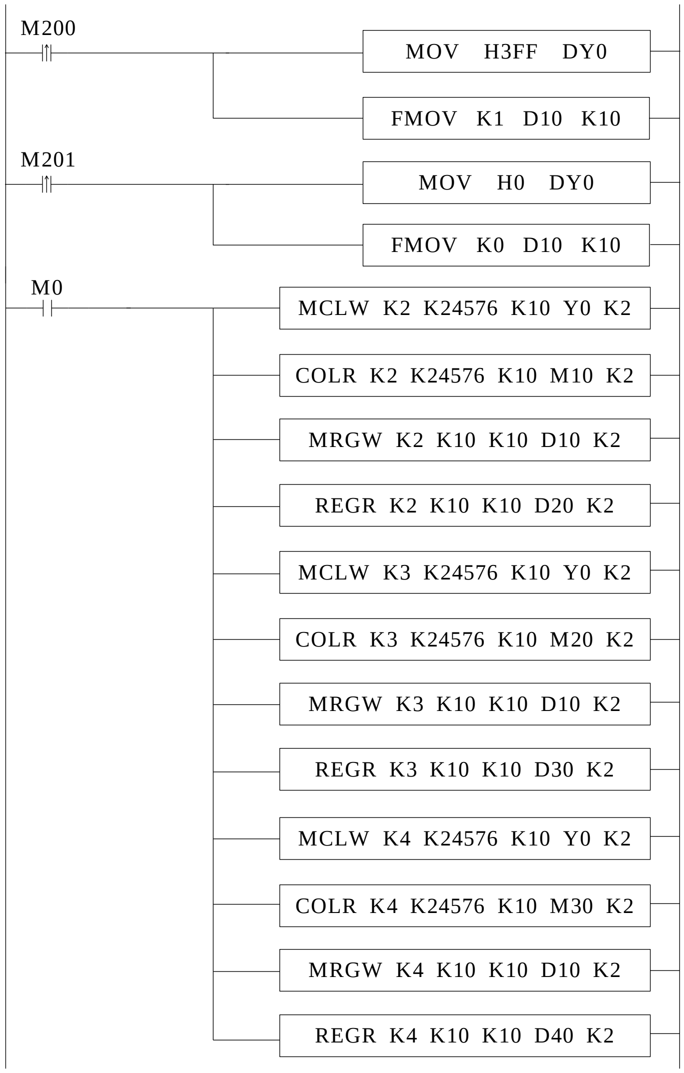
//write 1 to master PLC D10~D19, write the master PLC D10~D19 to D10~D19 of slave PLC 2, 3, 4. Enter process S3 when the communication is successful.



//read the D10~D19 of slave PLC 2, 3, 4 to master PLC D20~D29, reset D10~D19 after the communication is successful, then the station no. is added 1, process S0 is executed, cycle.

Modbus-RTU instruction processing mode has changed. Users can write Modbus-RTU instructions directly in user programs. Protocol stack will queue Modbus-RTU communication requests. Communication is another task. In the main program, users can write multiple Modbus-RTU communication instructions together and trigger them at the same time through the same triggering condition. PLC will trigger these communications. Instructions are queued according to the protocol station by Modbus-RTU, which will not cause communication errors when multiple communication instructions are executed at the same time as the original XC series PLC.

XD series program:



//at the rising edge of M200, set ON the master PLC Y0~Y11, D10~D19 are set to 1, at the rising edge of M201, set OFF Y0~Y11 of master PLC, reset D10~D19.

//write the Y0~Y11 of master PLC to Y0~Y11 of slave PLC 2, read the Y0~Y11 of slave PLC 2 to M10~M19 of master PLC. Write the D10~D19 of master PLC to D10~D19 of slave PLC 2. Read the D20~D29 of slave PLC 2 to D20~D29 of master PLC.

6-3. Free communication

6-3-1. Free communication mode

Free format communication is data transmission in the form of data blocks, limited by the PLC cache, the maximum amount of data sent each time is 256 bytes.

The so-called free communication, i.e. custom protocol communication, now many intelligent devices on the market support RS232 or RS485 communication, but the protocols used by various products are different, such as: Xinje PLC uses standard Modbus-RTU protocol, some temperature controller manufacturers use custom protocols; if using Xinje PLC to communicate with temperature controller, it is necessary to use free communication to send data in full accordance with the protocol of the instrument manufacturer, so as to communicate.

Prerequisites for free communication:

1. Port0(RS232), Port1(RS232), Port2(RS485) or Port2-RS232(RS232) or Port2-RS485(RS485), Port3(left extension port), Port4(upper extension port 1), Port5(upper extension port 2) all support free communication. As the free communication needs to change the communication parameters, port1 is not recommended.
2. Baud rate: 300bps~3Mbps, 4.5Mbps~9Mbps (special model supported)
3. The data format must be the same as the lower device settings. There are several options as follows:
   Data bit: 5 bits (special model supported), 6 bits (special model supported), 7 bits, 8 bits, 9 bits.
   Parity bit: none, odd parity, even parity, empty, mask
   Stop bit: 1 bit, 1.5 bit, 2 bits
4. Starter: 1 byte, terminator: 1 byte
   Users can set a start/termination character. After setting the start/termination character, PLC automatically adds the start/termination character when sending data, and automatically removes the start/termination character when receiving data.
   In fact, the initiator and terminator can be regarded as the data frame head and end in the protocol. Therefore, if the lower device communication has start and termination character, it can be set in the software or written in the protocol.
5. Communication mode: 8 bits, 16 bits
   When 8-bit buffer is selected for communication, the high bytes of registers are invalid. PLC only uses the low bytes of registers to send and receive data.
   When 16-bit buffer is selected for communication, the PLC will send all the data of the register, and send low-byte data first, then high-byte data.
   When it is necessary to transfer low bytes and high bytes of one 16-bit register to another 16-bit register, 16-bit buffers must be selected for communication, and the number of communication bytes is 2. When the value stored in a 16-bit register occupies only low bytes, we can choose 8-bit buffer to communicate. The number of communication bytes is 1. Usually when we communicate, the data will not exceed the

low byte of a register (HFF), so we only need to use the default 8-bit buffer in the software to communicate.

6. Timeout: frame timeout (ms), reply timeout (ms)

   Frame: A data string.

   Frame timeout: refers to the time interval between two frames of data received by the PLC, which ensures that the PLC can distinguish the end time of receiving a frame. It is usually used to judge whether a frame of data in PLC has been received or not. When the interval between two frames of data is longer than the frame time-out, it means the end of one frame of communication data.

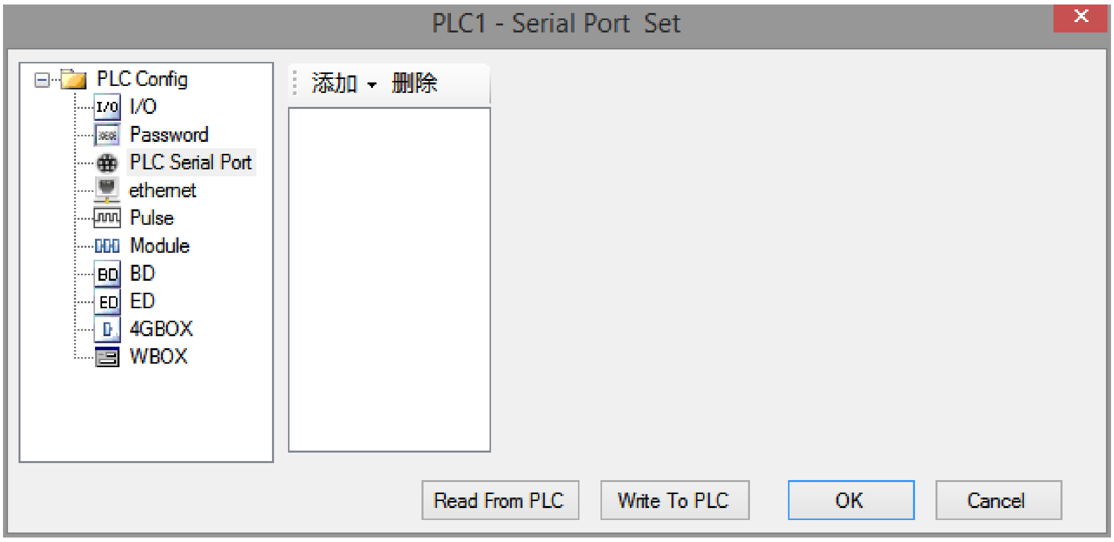
   Reply timeout: refers to the time when the PLC can not receive the response after sending the request, waiting for the resend. If the response time is set to exceed 300 ms, when default communicating, the PLC waits 300ms for the other party to respond. If the response time is not received, the request will be sent again.

   If you want to shorten the communication time, you can adjust the above two parameters according to the size of baud rate.

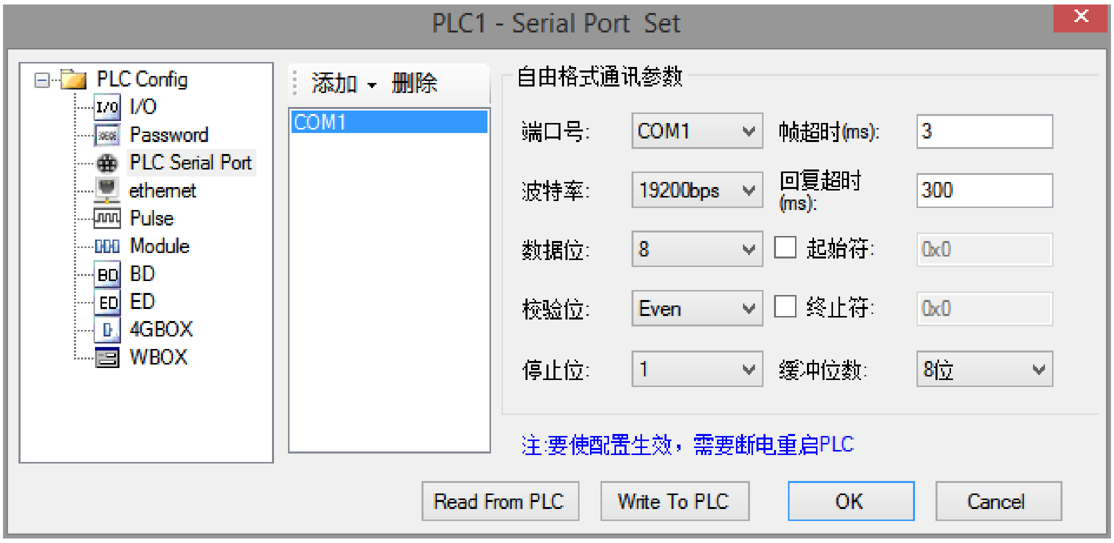
6-3-2. Serial port configuration

(1) Use the USB download cable to connect the PLC with the computer. Here the USB download cable is the HMI download cable, as shown below, the software must switch to XNet communication mode.

(2) Open the programming software, click configure/PLC comm port settings. It will show below figure:



(3) Click add, it will show two modes, modbus mode and free mode, please select free mode, it will show below figure.



Port No.: It refers to Port of PLC, COM0 refers to Port 0 (RS232), COM1 refers to Port 1 (RS232), COM2 refers to Port 2 (RS485) or Port 2-RS232 (RS485) or Port 2-RS485 (RS485), COM3 refers to Port 3 (left extended ED port), COM4 refers to Port 4 (upper extended BD port 1), COM5 refers to Port 5 (upper extended BD port 2).
Frame timeout (ms): It refers to the time interval between two frames of data sent by PLC, which ensures that the receiver distinguishes the end time of receiving a frame.
Response timeout (ms): refers to the time when the PLC can not receive the response after sending the request, waiting for the resend.
Other serial parameters can be set according to the parameters of the lower device.

(4) After setting, click write to PLC, then cut off the PLC power supply and power on again to make the settings effective.

Note: Versions V3.4 and above can be configured in Modbus communication mode (RS232 port); Versions V3.4 and below XD series PLC must use X-NET communication mode when downloading and uloading serial configuration data, that is, downloading and uploading configuration data through USB port.

6-3-3. Suitable occasion

When does free communication need to be used?
As an example, the situation described in the above section is that XINJE PLC communicates with the temperature control instrument, and the instrument uses its own communication protocol, which stipulates that the reading temperature should be sent four characters: "R", "T", "CR". Each character has the following meanings:

Character	Meaning
：	Data start
R	Read
T	temperature
CR	Enter, data end

PLC needs to send the ASCII code of the above characters to the instrument in order to read the current temperature value measured by the instrument. The ASCII code values (hexadecimal) of each character can be obtained by querying the ASCII code table.

Character	ASCII code value
:	3A
R	52
T	54
CR	0D

Obviously, according to the situation described above, using MODBUS instructions can not communicate, at this time you need to use free communication. Detailed usage will be used as an example to program the sample program in later chapters.

6-3-4. Free communication instruction

Send data [SEND]

1. Instruction overview

Write the local data to specified remote station address.

Send data [SEND]			
16-bit instruction	SEND	32-bit instruction	-
Execution condition	Normally ON/OFF, rising edge triggering	Suitable model	XD, XL
Hardware	V3.2.3 and higher version	Software	V3.2.2 and higher version

2. Operand

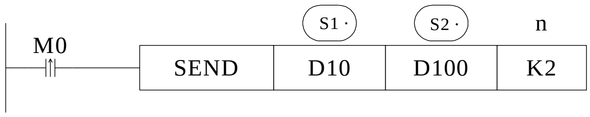
Operand	Function	Type
S1	Local data starting address	16-bit, BIN
S2	Send byte number	16-bit, BIN
n	Communication port no.	16-bit, BIN

3. Suitable soft component

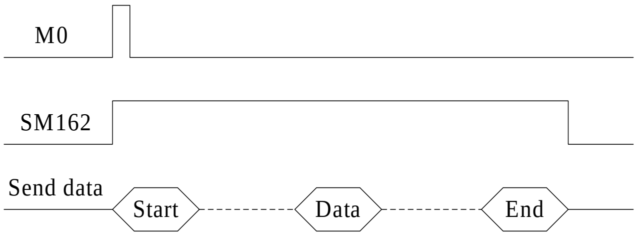
Word

operand	System									constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
S1	●	●		●	●							
S2	●	●		●	●					●		
n	●									K		

Function and action



- Data sending instructions, M0's rising edge sends data once.
- Communication port. Scope: K0 ~ K5. Port0, Port1, Port2 or Port2-RS232 or Port2-RS485, Port3, Port4, Port5.
- In the process of data transmission, the "sending" flag SM162 (communication port 2) is set on.



- When the buffer number is 8 bits, only low-byte data is sent, so D100 = the number of registers sent, for example, to send low-byte data in D10-D17, D100 should be set to 8.
- When the buffer number is 16 bits, high and low byte data will be sent, so D100 = the number of registers sent * 2. For example, when sending high and low byte data in D10-D17, D100 should be set to 16, and when sending, low byte will be before the high byte.

Receive data [RCV]

1. Instruction overview

Write the specified remote station no’s data to local device.

Send data [RCV]			
16-bit instruction	RCV	32-bit instruction	-
Execution condition	Normally ON/OFF, rising edge triggering	Suitable model	XD, XL
Hardware	V3.2.3 and higher version	Software	V3.2.2 and higher version

2. Operand

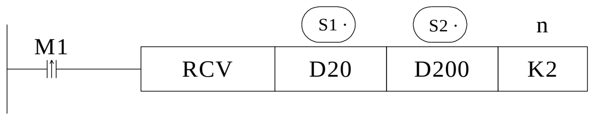
Operand	Function	Type
S1	Local data starting address	16-bit, BIN
S2	Receive byte number or soft component address	16-bit, BIN
n	Communication port no.	16-bit, BIN

3. Suitable soft component

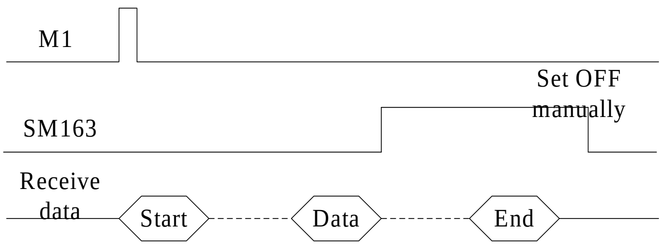
Word

operand	System									constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
S1	●	●		●	●							
S2	●	●		●	●					●		
n										●		

Function and action



- Data receiving instructions, M1's rising edge receives data once.
- Communication port. Scope: K0 ~ K5. Port0, Port1, Port2 or Port2-RS232 or Port2-RS485, Port3, Port4, Port5.
- After receiving the data, the "received" flag SM163 (communication port 2) is set on.



- When the buffer number is 8 bits, the received data is only stored in low bytes, so D200 = the number of bytes to be received * 2, for example, to receive 8 bytes of data, stored in the low bytes of the eight registers D20-D27 in turn, at this time, D200 should be set to 16.
- When the buffer number is 16 bits, the received data is stored in a complete register, so D200 = the number of bytes to be received, for example, to receive 8 bytes of data, stored in the four registers of D20-D23 in turn, at this time, D200 should be set to 8. And when receiving, low bytes are before high bytes.

Release serial port [RCVST]

1. Instruction overview

Release the specified serial port.

Release serial port [RCVST]			
16-bit instruction	RCVST	32-bit instruction	-
Execution condition	Normally ON/OFF, rising edge triggering	Suitable model	XD, XL
Hardware	V3.2.3 and higher version	Software	V3.2.2 and higher version

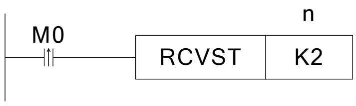
2. Operand

Operand	Function	Type
n	Communication port no.	16-bit, BIN

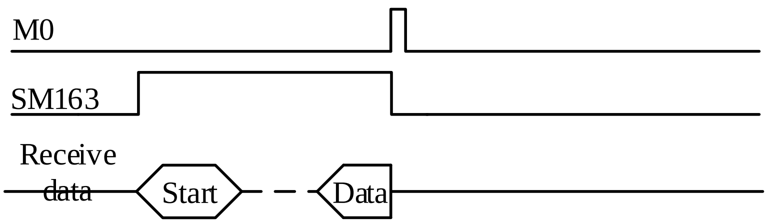
3. Suitable soft component

Word	operand	System									constant	Module	
		D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
	n										K		

Function and action



- Release serial port instructions, M0's rising edge execute once.
- Communication port. Scope: K0 ~ K5. Port0, Port1, Port2 or Port2-RS232 or Port2-RS485, Port3, Port4, Port5.
- When releasing the serial port, the "received" flag SM163 (communication port 2) is set OFF.
- For free communication, if there is no timeout or the timeout time is set too long, the occupied serial port resources can be released immediately through RCVST instructions for other communication operations.



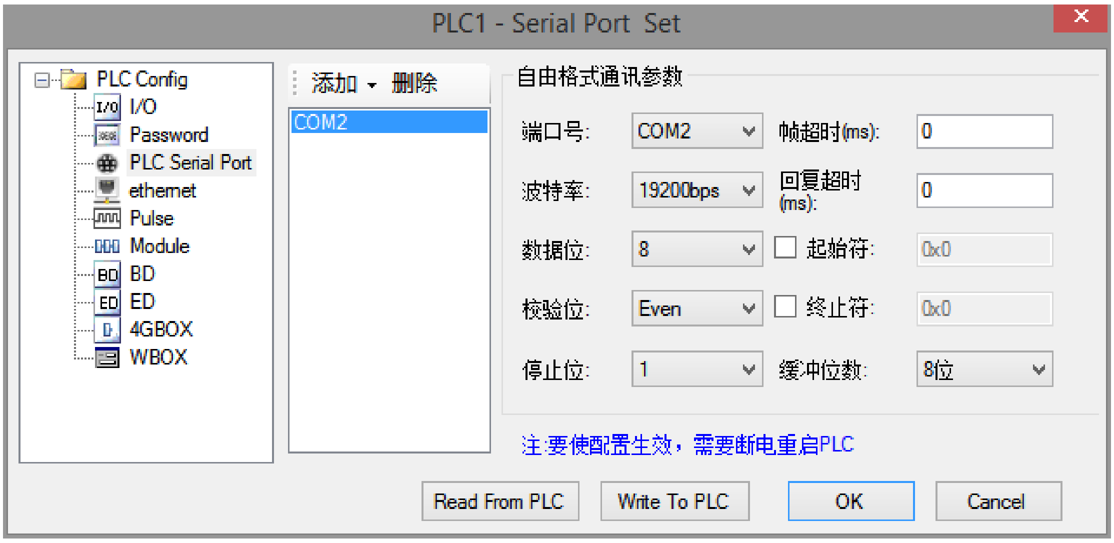
6-3-5. Free communication example

Example 1: In chapter 6-3-3, we give an example of communication between Xinje PLC and temperature control instrument when explaining why to use free communication. Here is an example.

Operation steps:

1. Connect the hardware first. Here we use the serial port 2 of the PLC to communicate, that is, 485 + on the instrument is connected to A of the output port of the PLC, and 485- on the instrument is connected to B of the output port of the PLC.

2. Set the serial port parameters of PLC according to the communication parameters of temperature control instrument. The parameters are set as follows. After setting the parameters, the power can be restarted.



3. make the program according to the descriptions in chapter 6-3-3.

Read temperature: “ : ” “R” “T” “CR”

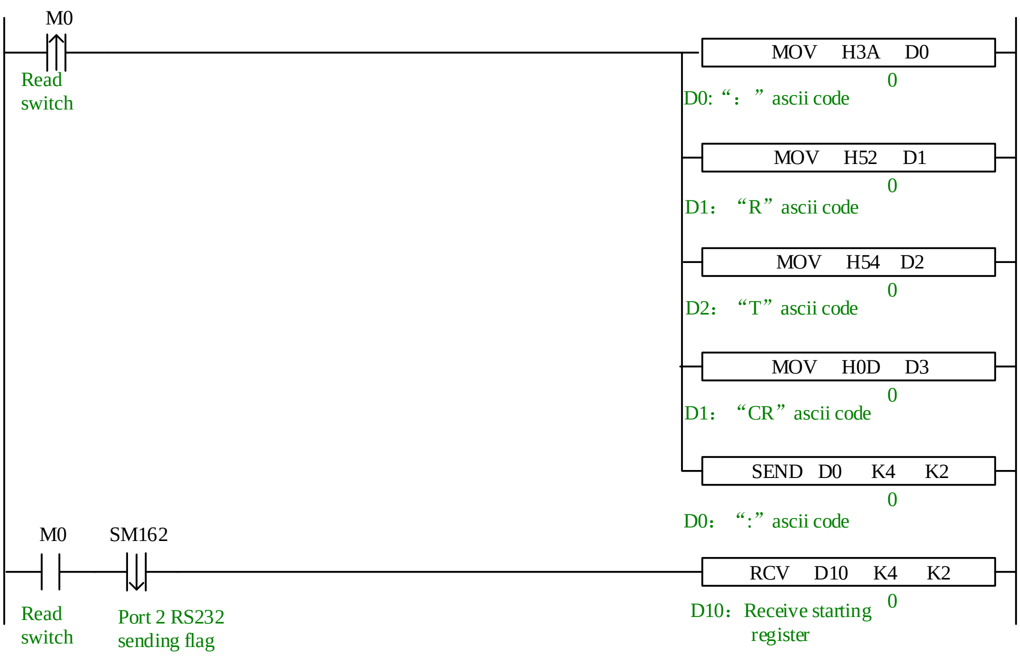
“ : ” -------------- data start

“R” --------------- read

“T” --------------- temperature

“CR” ------------- enter, data end

Program:



When trying to communicate between PLC and other intelligent devices, it is suggested to use serial debugging tool to determine the data format of communication, that is, protocol. The advantages of this method are: the serial debugging tool is easy to modify and flexible to use; after the serial debugging tool determines that communication can be successful, the PLC program is written according to the data format obtained, which is often twice the result with half the effort.

In fact, Modbus-RTU protocol can be regarded as a special kind of free protocol. The relationship between them is similar to ellipse and circle. We can try to use free format to realize the function of Modbus instruction.

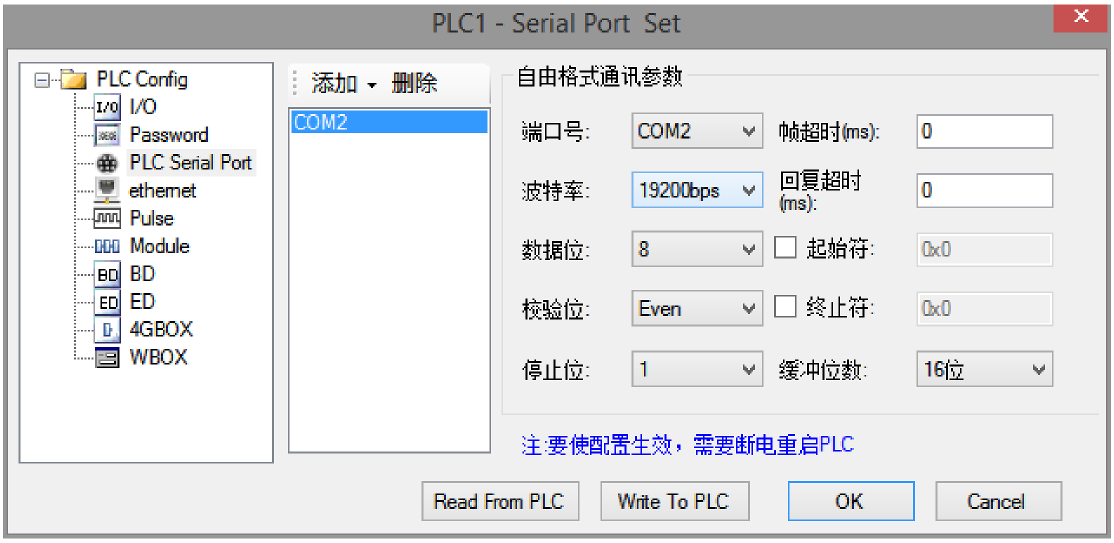
Example 2: The values of the five registers of a XD3 PLC are sent to the D1-D5 of another XDM PLC.

If the user understands the Modbus communication, he can use the Modbus-RTU communication mode to do so, as long as he writes a "write multiple register instructions (MRGW)" in the host. Here we do it in free communication mode.

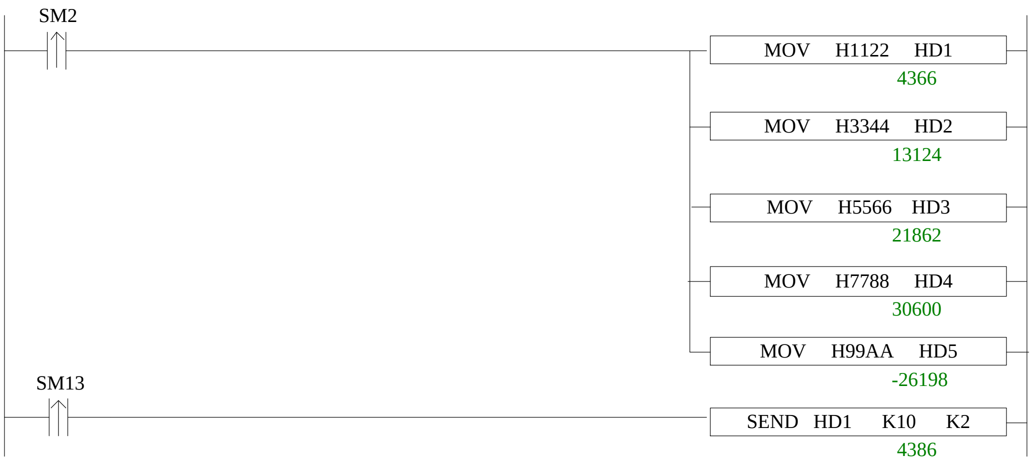
Operation steps:

1. Connect the hardware first. Here we use the serial port 2 of the PLC to communicate, that is, connect A of the two PLC, and connect B of the two PLC.

2. Set the same serial port parameters of the two PLC. The parameters are set as follows. After setting the parameters, the power can be restarted.



3. XD3 program:



XDM program:

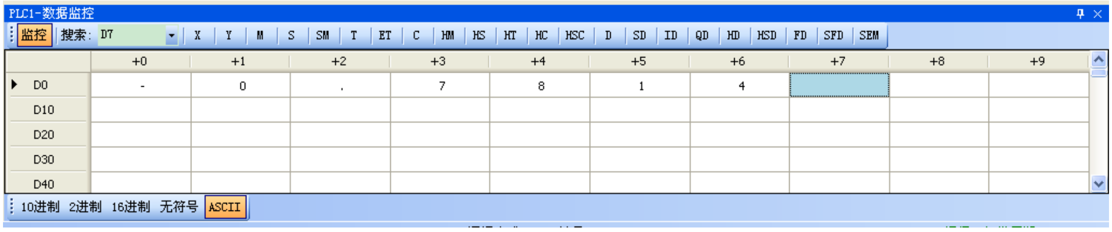
Sometimes the data of user communication is stored in multiple registers in the form of ASCII code. Users need to take this value out, store it in a register and display it on the HMI. Customers often consider using HEX (ASCII to hexadecimal) instructions to achieve it. But HEX instructions are difficult to use and understand. Often, we will not use this instruction to complete it. The relationship between values can be found by ASCII code comparison table.

ASCII code table:

ASCII value	Control character	ASCII value	Control character	ASCII value	Control character	ASCII value	Control character
0	NUT	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	”	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	’	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	，	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	。	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	：	90	Z	122	z
27	ESC	59	；	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	—	127	DEL

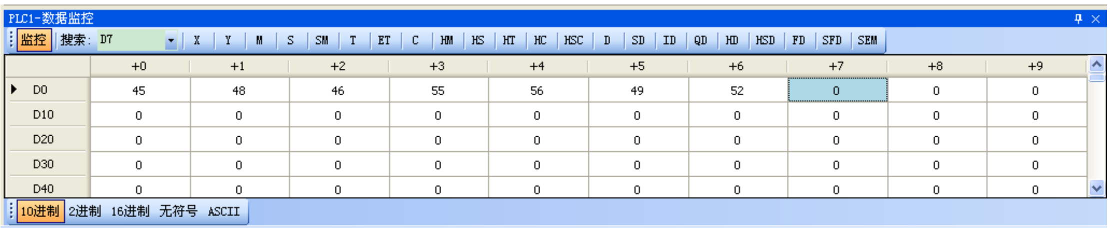
Example 3: A pressure controller communicates with PLC in free communication mode to realize data acquisition. The value displayed on the pressure controller is -0.7814 MPa. The value collected by PLC is stored from D0, and seven registers are stored in turn. However, the value of the seven registers combination needs to be taken out and stored in D46 in the form of decimal.

Through the data monitoring of PLC, ASCII codes in D0~D6 registers can be monitored as follows:



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	-	0	.	7	8	1	4			
D10										
D20										
D30										
D40										

Switch to decimal format and show as below:



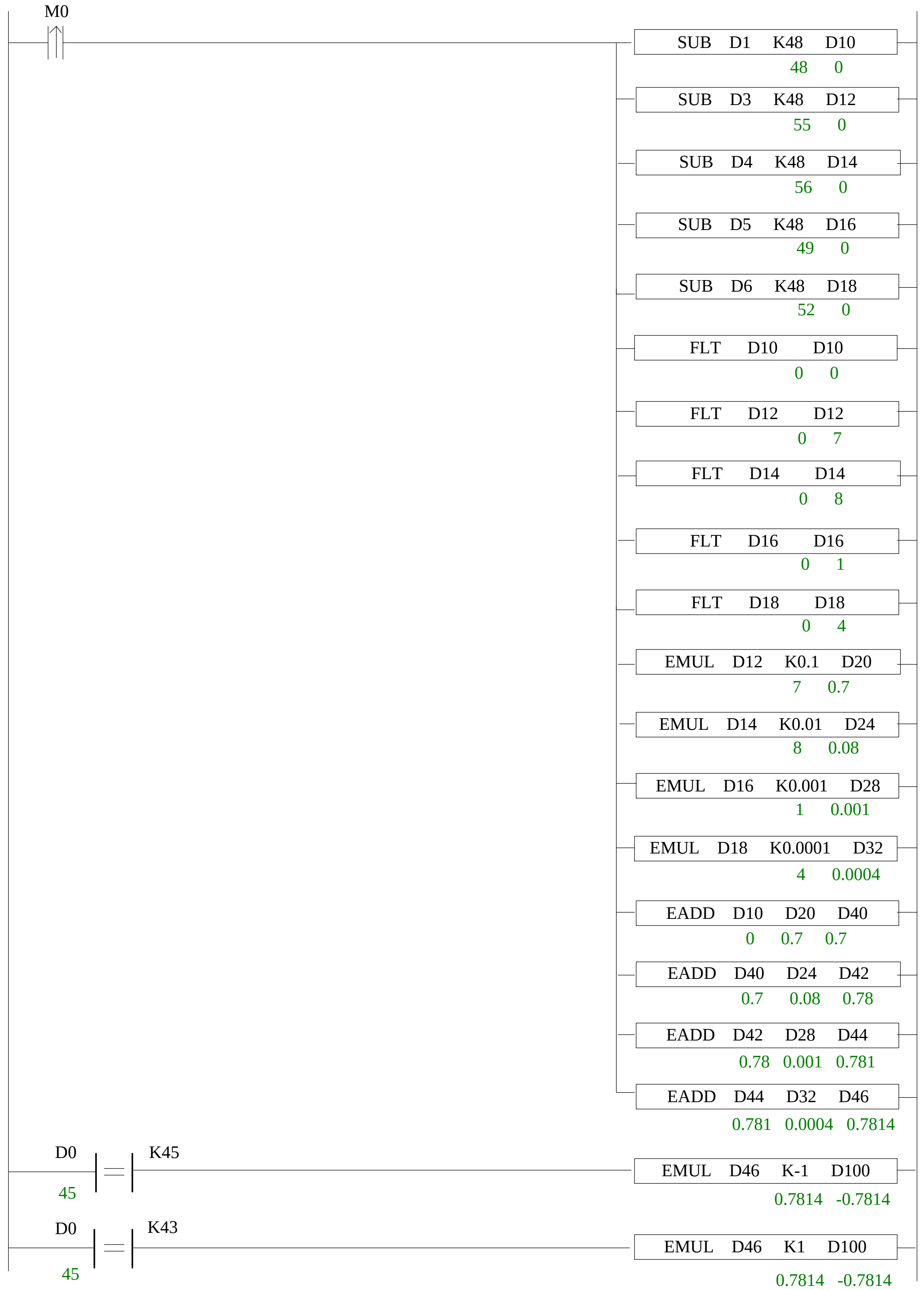
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	45	48	46	55	56	49	52	0	0	0
D10	0	0	0	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0	0	0

By comparing the relationship between ASCII codes and decimal values, we can find the rule that there is 48 difference between ASCII codes in D1, D3, D4, D5, D6 and decimal values. The final decimal values are obtained by subtracting the values in registers by K48 and multiplying by 10. The formula is as follows:

D46=(D1-48)*1+(D3-48)*0.1+(D4-48)*0.01+(D5-48)*0.001+(D6-48)*0.0001

D0 is a symbol bit. Looking up the table, we know that when D0 = K45, it represents a negative value; when D0 = K43, it represents a positive value.

The ladder diagram is as follows:


M0
SUB D1 K48 D10
48 0
SUB D3 K48 D12
55 0
SUB D4 K48 D14
56 0
SUB D5 K48 D16
49 0
SUB D6 K48 D18
52 0
FLT D10 D10
0 0
FLT D12 D12
0 7
FLT D14 D14
0 8
FLT D16 D16
0 1
FLT D18 D18
0 4
EMUL D12 K0.1 D20
7 0.7
EMUL D14 K0.01 D24
8 0.08
EMUL D16 K0.001 D28
1 0.001
EMUL D18 K0.0001 D32
4 0.0004
EADD D10 D20 D40
0 0.7 0.7
EADD D40 D24 D42
0.7 0.08 0.78
EADD D42 D28 D44
0.78 0.001 0.781
EADD D44 D32 D46
0.781 0.0004 0.7814
D0 K45
45
EMUL D46 K-1 D100
0.7814 -0.7814
D0 K43
45
EMUL D46 K1 D100
0.7814 -0.7814

6-4. Communication flag and register

Communication flag

Serial port	Register address	Function	Explanation
Port 0	SM140	Modbus read-write instruction execution flag	When the instruction starts to execute, set ON When execution is completed, set OFF
	SM141		
	SM142	Free communication sending flag	When the instruction starts to execute, set ON When execution is completed, set OFF
	SM143	Free communication received flag	When receiving a frame of data or receiving data timeout, set ON. Require user program to set OFF
	SM144		
	……		
	SM149		
Port 1	SM150	Modbus read-write instruction execution flag	When the instruction starts to execute, set ON When execution is completed, set OFF
	SM151		
	SM152	Free communication sending flag	When the instruction starts to execute, set ON When execution is completed, set OFF
	SM153	Free communication received flag	When receiving a frame of data or receiving data timeout, set ON. Require user program to set OFF
	SM154		
	……		
	SM159		
Port 2	SM160	Modbus read-write instruction execution flag	When the instruction starts to execute, set ON When execution is completed, set OFF
	SM161		
	SM162	Free communication sending flag	When the instruction starts to execute, set ON When execution is completed, set OFF
	SM163	Free communication received flag	When receiving a frame of data or receiving data timeout, set ON. Require user program to set OFF
	SM164		
	……		

	SM169		
Port 3	SM170~SM179		
Port 4	SM180~SM189		
Port 5	SM190~SM199		

Communication registers

	No.	Function	Explanation
Port 0	SD140	Modbus read and write instruction execution result	0: correct 100: receive error 101: receive timeout 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error (erase FLASH)
	SD141	X-Net communication result	0: correct 1: communication timeout 2: memory error 3: receive CRC error
	SD142	Free communication sending result	0: correct 410: free communication buffer overflow
	SD143	Free communication receiving result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive timeout 415: no start symbol 416: no end symbol
	SD144	free communication receiving data number	Count as byte, not include start symbol and end symbol
	……		
	SD149		
Port 1	SD150	Modbus read and write instruction execution result	0: correct 100: receive error 101: receive timeout 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error (erase FLASH)

	SD151	X-Net communication result	0: correct 1: communication timeout 2: memory error 3: receive CRC error
	SD152	Free communication sending result	0: correct 410: free communication buffer overflow
	SD153	Free communication receiving result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive timeout 415: no start symbol 416: no end symbol
	SD154	free communication receiving data number	Count as byte, not include start symbol and end symbol
	……		
	SD159		
Port 2	SD160	Modbus read and write instruction execution result	0: correct 100: receive error 101: receive timeout 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error (erase FLASH)
	SD161	X-Net communication result	0: correct 1: communication timeout 2: memory error 3: receive CRC error
	SD162	Free communication sending result	0: correct 410: free communication buffer overflow
	SD163	Free communication receiving result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive timeout 415: no start symbol 416: no end symbol
	SD164	free communication receiving data number	Count as byte, not include start symbol and end symbol
	……		
	SD169		
Port 3	SD170~SD179		
Port 4	SD180~SD189		
Port 5	SD190~SD199		

6-5. Read write serial port parameters

In addition to modifying communication parameters through serial configuration panel, it can also be realized by reading instruction [CFGCR] of serial parameters and writing instruction [CFGCW] of serial parameters.

6-5-1. Read serial port parameters [CFGCR]

1. Instruction overview

Read the serial port parameters to local specified registers.

Read serial port parameters [CFGCR]			
16-bit instruction	CFGCR	32-bit instruction	-
Execution condition	Normally ON/OFF, rising edge triggering	Suitable model	XD, XL
Hardware	-	Software	V3.4 and higher version

2. Operand

Operand	Function	Type
D	Local register starting address	16-bit, BIN
S1	Read serial port parameters number	16-bit, BIN
S2	Serial port no.	16-bit, BIN

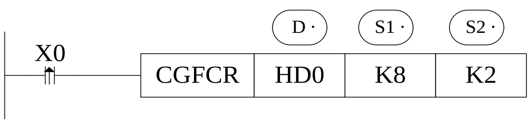
3. Suitable soft component

Word

operand	System									constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
D	●											
S1	●	●								●		
S2	●									●		

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM;**
DS stands for DS DHS.

Function and action



- Operator S1: The number of registers used to read serial parameters is generally 8 (XD5E/XDME series is 9).
- Operator S2: Serial port range: K0 ~ K5. K0: Port0, K1: Port1, K2: Port2 or Port2-RS232 or Port2-RS485, K3: Port3, K4: Port4, K5: Port5.
- Read 8 parameters of serial port 2 to HD0~HD7. See sections 6-5-3 for the names and

definitions of specific parameters.

6-5-2. Write serial port parameters [CFGCW]

1. Instruction overview

Write the local specified register value to specific serial port.

Write serial port parameters [CFGCW]			
16-bit instruction	CFGCW	32-bit instruction	-
Execution condition	Normally ON/OFF, rising edge triggering	Suitable model	XD, XL
Hardware	-	Software	V3.4 and higher version

2. Operand

Operand	Function	Type
S1	Local register starting address	16-bit, BIN
S2	Write serial port parameters number	16-bit, BIN
S3	Serial port no.	16-bit, BIN

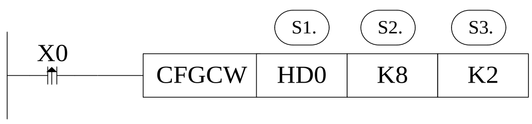
3. Suitable soft component

Word

operand	System									constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
S1	●											
S2	●	●								●		
S3	●									●		

*** Note: D denotes D HD; TD denotes TD HTD; CD denotes CD HCD HSCD HSD; DM denotes DM DHM;**
DS stands for DS DHS.

Function and action



- Operator S2: The number of registers used to write serial parameters is generally 8 (XD5E/XDME series is 9).
- Operator S3: Serial port range: K0 ~ K5. K0: Port0, K1: Port1, K2: Port2 or Port2-RS232 or Port2-RS485, K3: Port3, K4: Port4, K5: Port5.
- Write HD0~HD7 parameters to serial port 2. See sections 6-5-3 for the names and definitions of specific parameters.

6-5-3. Serial port parameter name and setting

Assuming that HD0-HD14 corresponds to serial port parameters, the parameter names and settings represented by registers are shown in the table below.

Parameter address	Parameter name and settings				
	MODBUS communication（HD0=1）	Free communication（HD0=2）	X-NET communication		Ethernet communication（HD0=3）
			OMMS（HD0=3）	TBN（HD0=3）	
HD0	Network type 1: MODBUS; 2: free ; 3: X-NET; 4: MODBU-TCP				
HD1	MODBUS station no. 1~254	Baud rate refer to table 1	Net ID 0~32767	Net ID 0~32767	Net ID IP address high 2-byte
HD2	Transmission mode 0: RTU 128: ASCII	Frame format refer to table 2	Station no. 0~100	Station no. 0~100	Station no. IP address low 2-byte
HD3	Baud rate refer to table 1	Free properties bit7: 1: with start character 0: no start character bit6： 1: with end character 0: no end character	Physical layer type 1: PHY_RS485 2: PHY_SOF (Unidirectional Fiber Ring Network) 3: PHY_OFPP (Optical Fiber Point Network) 4: PHY_RS232 5: PHY_RS422 6: PHY_TTL (TTLvoltage network)		
HD4	Frame format refer to table 2	Start character	Link Layer Type 0: TBN 1: HDN 2: CCN 3: PPFD 4: PPU 5: Ethernet		
HD5	retry count 0~5	End character	OMMS properties 128: Supports periodic communication, otherwise does not support	Baud rate refer to table 1	Subnet mask high 2-byte
HD6	Reply timeout 0~65535	Frame timeout 0~255	OMMS baud rate refer to table 1	Token Cycle Time 1~60000 (ms)	Subnet mask low 2-byte
HD7	Delay before sending 0~255	Reply timeout 0~65535 (0 is infinite wait)	OMMS slave station list Each bit of each byte in the array indicates whether the slave station is accessible (the	Max station number 1~100	Gateway address high 2-byte

			master station is valid, i.e. the station number is 1).		
HD8	-	-	-	-	Gateway address low 2-byte

Note: The table does not contain "buffer digits" in free communication mode, so "buffer digits" can not be read and written through CFGCR and CFGCW instructions, but can be read and written using MOV instructions. The address of "buffer digits" is shown in Appendix 3.

Table 1: baud rate

Value	Baud rate	Value	Baud rate	Value	Baud rate	Value	Baud rate
1	300 bps	7	19200 bps	13	256000 bps	19	1000000 bps
2	600 bps	8	28800 bps	14	288000 bps	20	1200000 bps
3	1200 bps	9	38400 bps	15	384000 bps	21	1500000 bps
4	2400 bps	10	57600 bps	16	512000 bps	22	2400000 bps
5	4800 bps	11	115200 bps	17	576000 bps	23	3000000 bps
6	9600 bps	12	192000 bps	18	768000 bps		

Table 2: frame format

Stop bit		Parity bit			Data bit length		
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00: 1 01: 1.5 10: 2		000: no 001: odd 010: even 011: empty 100: Mask			000: 5 001: 6 010: 7 011: 8 100: 9		

7 PID Control Function

In this chapter, we mainly introduce the applications of PID instructions for XD, XL series, including: call the instructions, set the parameters, items to notice, sample programs etc.

7-1． PID Introduction

PID instruction and auto tune function are added into XD/XL series PLC basic units. Via auto tune method, users can get the best sampling time and PID parameters and improve the control precision.
PID instruction has brought many facilities to the users.
Output can be data form D, HD, and on-off quantity Y, user can choose them freely when programming.
Via auto tune, users can get the best sampling time and PID parameters and improve the control precision.
User can choose positive or negative action via software setting. Positive action is used for heating control; negative action is used for cooling control.
PID control separates the basic units with the expansions, which improves the flexibility of this function.
XD/XL series PLC have two methods for auto tune, step response method and critical oscillation method.
For temperature control object:
Step response method: the PID auto tune will start when current temperature of object controlled is equal to ambient temperature.
Critical oscillation method: the PID auto tune can start at any temperature.

7-2． Instruction Form

Brief Introduction of the Instructions
Execute PID control instructions with the data in specified registers.

PID control [PID]			
16 bits instruction	PID	32 bits instruction	-
Executing condition	Normally ON/normally closed coil trigger	Suitable models	XD/XL
Hardware requirement	-	Software requirement	V3.2

Operands

Operands	Function	Type
S1	set the address of the target value (SV)	16bits, BIN
S2	set the address of the tested value (PV)	16 bits, BIN
S3	set the start address of the control parameters	16 bits, BIN
D	the address of the operation result (MV) or output port	16 bits, BIN; bit

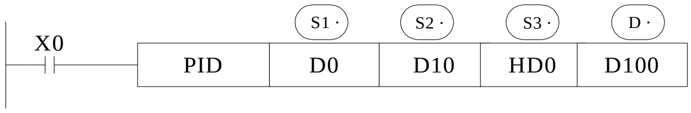
Suitable soft components

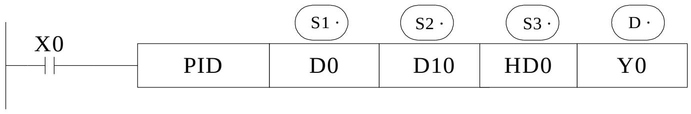
	Operands	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
Word	S1	●	●							●		
	S2	●	●									
	S3	●	●									
	D	●	●									

	Operands	System						
		X	Y	M*	S*	T*	C*	Dn.m
Bit	D		●	●	●	●	●	

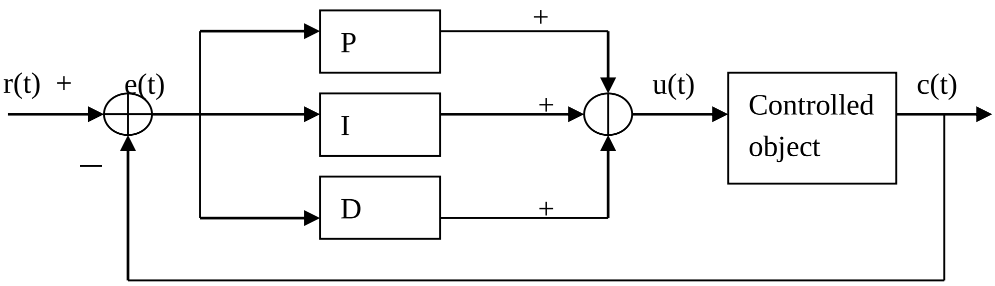
*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD;
DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

Functions and Action





S3~ S3+ 69 will be occupied by this instruction, so please don’t use them as the common data registers.
This instruction executes when each sampling time interval comes.
For the operation result, data registers are used to store PID output values; the output points are used to output the occupy duty ratio in the form of ON/OFF.
PID control rules are shown as below:
P: proportion, I: integral, D: differential



Analog PID control system

$e(t) = r(t) - c(t)$ (1-1)

$u(t) = Kp [e(t) + 1/Ti \int e(t)dt + TD\, de(t)/dt]$ (1-2)

Here, e(t) is offset value, r (t) is the setting value, c (t) is actual output value and the u(t) is the control value;

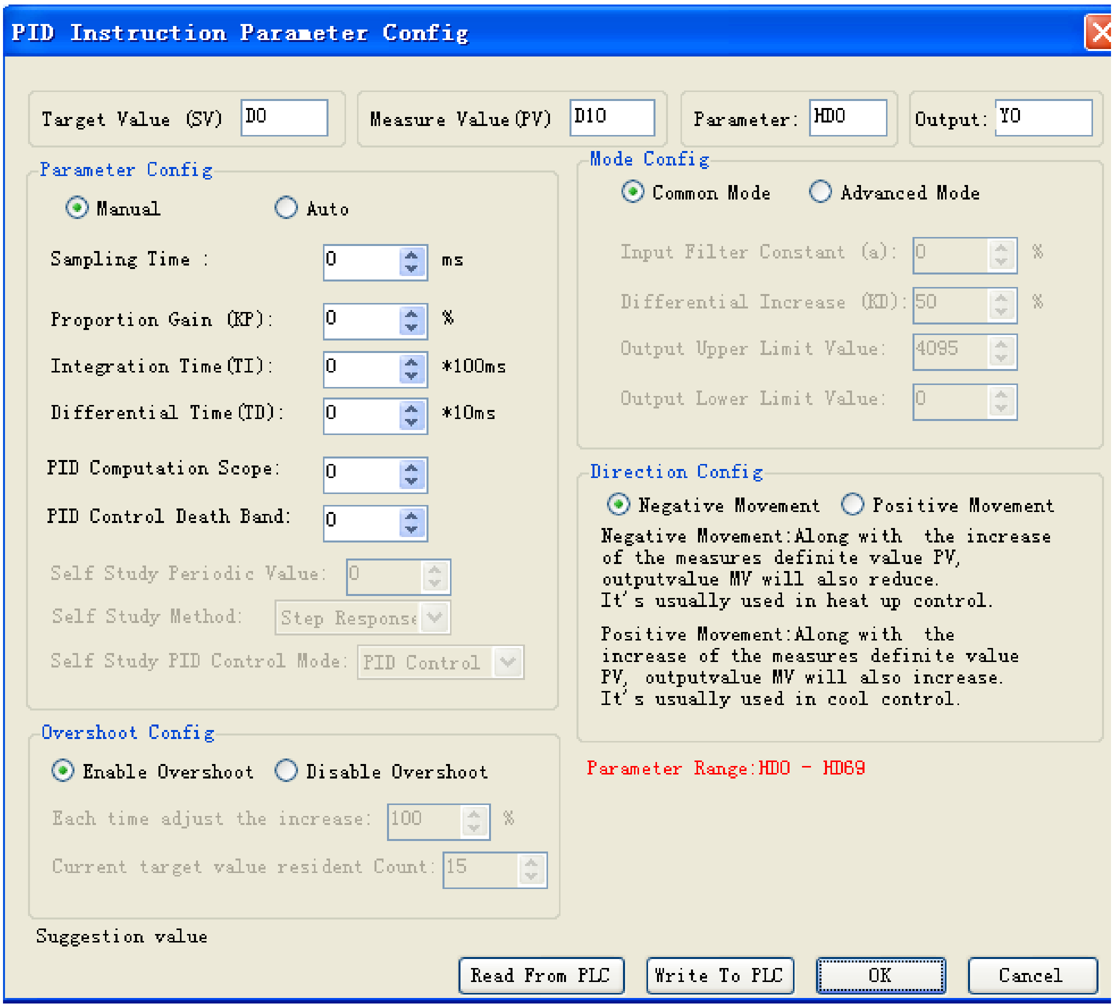
In function (1-2), Kp is the proportion coefficient, Ti is the integration time coefficient, and TD is the differential time coefficient.

The result of the operation:

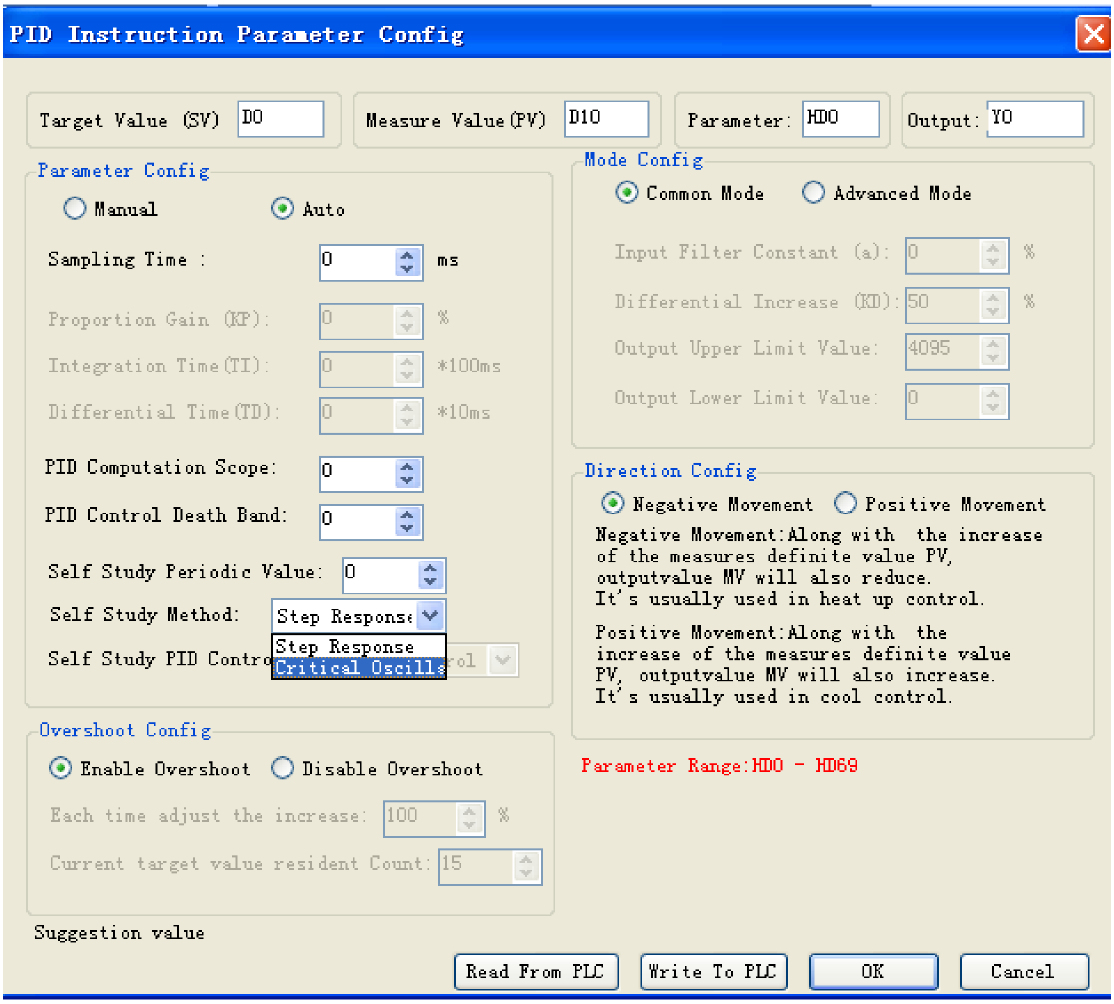
1. Analog output: digital form of MV = u(t), the default range is 0~4095.
2. Digital output: Y = T * [MV / PID output upper limit]. Y is the outputs activate time within the control cycle. T is the control cycle, equals to the sampling time. PID output upper limit default value is 4095.

7-3. Parameters setting

Users can call PID in XDP Pro software directly and set the parameters in the window (see graph below), for the details please refer to XDP Pro user manual. Users can also write the parameters into the specified registers by MOV instructions before PID operation.



Auto tune mode:



V3.2 and higher version software can choose auto tune mode: step response or critical oscillation.

7-3-1. Register and their functions

PID control instruction's relative parameters ID, please refer to the below table:

ID	Function	Description	Memo
S3	Sampling time	Whatever it is manual or auto mode, all needs to set	32 bits without sign, Unit ms
S3+2	Mode setting	bit0: 0: negative action; 1: positive action bit1～bit6 not usable bit7: 0: manual PID; 1: auto tune PID bit8: 1: auto tune successful flag bit9～bit10: auto tune method 00: step response 01: critical oscillation	

		bit11～bit12：not useful bit13～bit14 auto tune PID mode (valid in critical oscillation mode) 00：PID control 01：PI control 10：P control bit15： 0：regular mode； 1：advanced mode;	
S3+3	Proportion Gain (Kp)	Range：1～32767[%]	
S3+4	Integration time (TI)	0～32767[unit: 100ms]	0 is taken as no integral.
S3+5	Differential time (TD)	0～32767[unit: 10ms]	0 is taken as no differential.
S3+6	PID operation zone	0～32767	PID adjustment band width value
S3+7	Control death zone	0～32767	PID output value will not change in death zone
S3+8	Sampling temperature filter coefficient	0～100[%]	Filter the input sampling temperature in advanced mode, 0 is no input filter
S3+9	Differential gain(KD)	0～100[%]	Only for advanced mode (normal mode default value is 50%), 0 is no differential gain
S3+10	Upper limit value of output	0～32767	
S3+11	Lower limit value of output	0～32767	
S3+12	Change of Unit Temperature Corresponds to Change of AD Value	full scale AD value *（0.3~1%） default value is 10	16-bit no sign, only for step PID
S3+13	PID auto tune overshoot	0：enable overshoot 1： not overshoot（try to reduce the overshoot）	only for step PID
S3+14	Current target value adjusting percentage every time in auto tune end transition stage	Cannot adjust	16-bit no sign, only for step PID
S3+15	Number of times exceeding the target value in auto tune end transition stage when limiting the overshoot		only for step PID, default value is 15
S3+16	PID type and status	Bit0~bit1： 00: manual mode 01: step mode	Internal use parameters of the system for

		10: Critical oscillation mode Bit8： 0: manual control status 1: auto tune end, enter manual control status	monitoring purposes only
S3+17	PID max output	0～32767	Internal use parameters of the system for monitoring purposes only
S3+18	PID min output	0～32767	Internal use parameters of the system for monitoring purposes only
S3+19	Last time sampling time	0~sampling time (unit: ms)	16-bit no sign, Internal use parameters of the system for monitoring purposes only
S3+20	Actual sampling time space	The value is around the sampling time	32-bit no sign, Internal use parameters of the system for monitoring purposes only
S3+22	Last time user set target temperature	The value before changing the target temperature	Internal use parameters of the system for monitoring purposes only
S3+23	-	-	Parameter is reserved

The following is the joint address (divided into step setting, critical oscillation setting and manual control)			
Step part (read only parameters, only for monitoring)			
S3+24	Actual sampling space	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+26	Operating segment of auto-tuning PID	0: Preparation stage 1～2: auto tune parameter collection 3: calculate PID parameters	Internal usage parameters of the system
S3+28	Duration of auto-tuning PID operating parameters	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+30	Real-time accumulation of two inflection points	Clear and recalculate the time when reaching the inflection point 0～4294967296 (unit: ms)	Internal usage parameters of the system

S3+32	Sampling variation of inflection point	Sampling difference between two inflection points -2147483648~2147483647	Internal usage parameters of the system
S3+34	Sampling interval time of inflection point EK	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+36	Time from auto-tuning PID to inflection point	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+38	Last sampling temperature	-32767~32767	Internal usage parameters of the system
S3+39	The time from auto-tuning PID operation to inflection point	-32767~32767 (unit: ms)	Internal usage parameters of the system
S3+40	Starting sampling value of auto-tuning PID operation	-32767~32767	Internal usage parameters of the system
S3+41	Number of times at inflection point during auto-tuning	0~65535	Internal usage parameters of the system
S3+42	Useless time	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+44	Stop temperature	Temperature at the end of auto-tuning Range: -32767~32767	Internal usage parameters of the system
Critical oscillation part (read only parameters, only for monitoring)			
S3+24	PID control mode	0: PID control 1: PI control 2: P control	16-bit no sign, internal usage parameters of the system
S3+25	Current auto-tuning segment	0: Preparation stage 1: start to auto tune 2～3: auto-tuning parameter collection 4: calculation of PID parameters	16-bit no sign, internal usage parameters of the system
S3+26	The auto-tuning temperature is located at the number of peaks	0: first peak 1: second peak	16-bit no sign, internal usage parameters of the system
S3+27	The lowest sampling temperature	-32767～32767	Internal usage parameters of the system
S3+28	The highest sampling temperature	-32767～32767	Internal usage parameters of the system
S3+30	sampling time of the lowest sampling temperature	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+32	sampling time of the highest sampling temperature	0～4294967296 (unit: ms)	Internal usage parameters of the system

S3+34	auto-tuning time cumulative	0～4294967296 (unit: ms)	Internal usage parameters of the system
Manual control part (read only parameters, only for monitoring)			
S3+24	current target temperature	-32767~32767	Internal usage parameters of the system
S3+25	Need to update target temperature	0: no need 1: need	16-bit no sign, internal usage parameters of the system
S3+26	Number of times to reach target temperature	0~65535	Internal usage parameters of the system
S3+27	PID upper limit of operational range	-32767~32767	Internal usage parameters of the system
S3+28	PID lower limit of operational range	-32767~32767	Internal usage parameters of the system
S3+30	High voltage time when PID uses Y to output	0～4294967296 (unit: ms)	Internal usage parameters of the system
S3+32	Sampling temperature after last filtering	The filtered temperature acquired in the last sampling time (the input filter constant in the advanced mode needs to be set first)	Floating point, internal usage parameters of the system
S3+34	Last temperature deviation		Floating point, internal usage parameters of the system
S3+36	Value of last integral term	digital value corresponding to Ui of the last sampling time	Floating point, internal usage parameters of the system
S3+38	Value of last differential term	digital value corresponding to Ud of the last sampling time	Floating point, internal usage parameters of the system
S3+40	Last PID output		Floating point, internal usage parameters of the system

Note: When the auto-tuning mode is changed to manual control, the value in the original address of S3+24~S3+40 will be overwritten by the value in manual control mode.

7-3-2. Parameters Description

Movement direction:

Positive movement: the output value MV will increase with the increasing of the measured value PV, usually used for cooling control.

Negative movement: the output value MV will decrease with the increasing of the measured value PV, usually used for heating control.

Mode setting

Common Mode:

Parameters register range: S3～S3+69, and S3～S3+7 need to be set by users;
S3+8～S3+69 are occupied by system, users can't use them.

Advanced Mode

Parameters register range: S3～S3+69, among them S3～S3+7 and S3+8～S3+11 need to be set by users; S3+16～S3+69 are occupied by system, users can't use them.

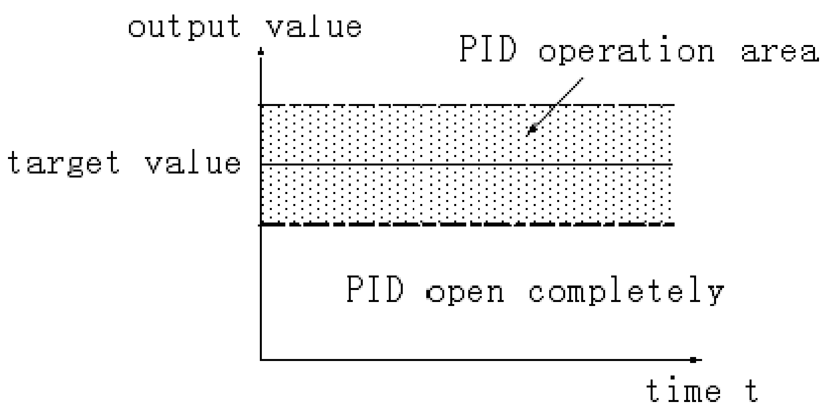
Sample time[S3]

The system samples the current values according to some certain interval and compares them with the output value. This time interval is the sample time **T**. There is no requirement for **T** during **DA** output; **T** should be larger than one PLC scan period during port output. **T** value should be chosen among 100~1000 times of PLC scan periods.

PID Operation Zone[S3+6]

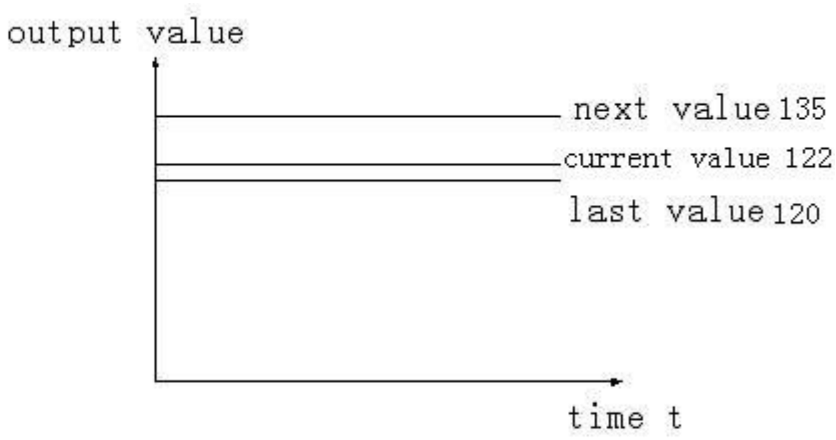
PID control is entirely opened at the beginning and close to the target value with the highest speed (default value is 4095), when it entered into the PID computation range, parameters Kp, TI, TD will be effective.

See graph below:



If the target value is 100, PID operation zone is 10, and then the real PID's operation zone is from 90~110.

Death Region [S3+7]
If the measured value changed slightly for a long time, and PID control is still in working mode, then it belongs to meaningless control. Via setting the control death region, we can overcome this situation. See graph below:



Suppose: we see the death region value to be 10. Then in the above graph, the difference is only 2 comparing the current value with the last value. It will not do PID control; the difference is 13 (more than death region 10) comparing the current value with the next value, this difference value is larger than control death region value. it will do the PID control with 135.

7-4. Auto Tune Mode

If users do not know how to set the PID parameters, they can choose auto tune mode which can find the best control parameters (sampling time, proportion gain **Kp**, integral time **Ti**, differential time **TD**) automatically.

Auto tune mode is suitable for these controlled objects: temperature, pressure; not suitable for liquid level and flow.

Auto-tuning is the process of extracting PID parameters. Sometimes auto-tuning can not find the best parameters at one time. It needs auto-tuning for many times. It is normal that there is a vibration in the process. After the optimum parameters are found at the end of auto-tuning, please switch to the manual PID mode. If the control object is unstable in the process of manual PID, it can not be controlled at a constant target value, which may be caused by the unsatisfactory adjustment of parameters. It is necessary to re-adjust the parameters of PID to achieve stable control.

For step response method: Users can set the sampling cycle to be 0 at the beginning of the auto tune process then modify the value manually in terms of practical needs after the auto tune process is completed.

For step response method: Before doing auto tune, the system should be under the non-control steady state. Take the temperature for example: the measured temperature should be the same to the environment temperature.

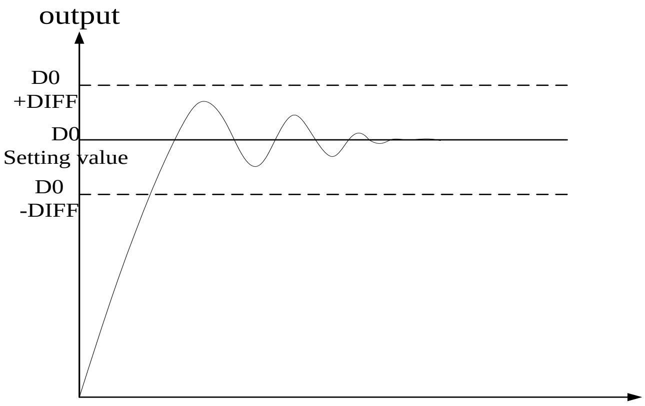
For critical oscillation method: user needs to set the sampling time at the beginning of the auto tune process. For slow response system, 1000ms. For fast response system, 10-100ms.

For critical oscillation method: the system can start the auto tune at any state. For object temperature, the current temperature doesn’t need to be same to ambient temperature.

Two different methods and PID control diagram:

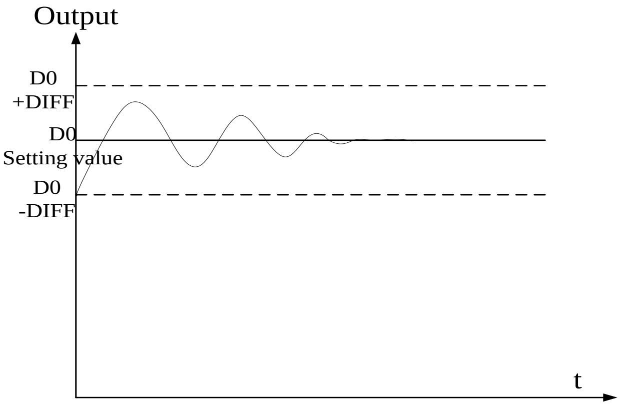
(1) Step response method

Make sure current temperature is equal to ambient temperature



(2) Critical oscillation method

The auto tune start temperature can be any value.



To enter the auto tune mode, please set bit7 of (**S3+ 2)** to be 1 and turn on PID working condition. If bit8 of (**S3+ 2)** turn to 1, it means the auto tune is successful.

PID auto tune period value [S3+12]

Set this value in S3+12 during auto tune. This value decides the auto tune performance, in a general way, set this value to be AD result corresponding to one standard tested unit. The default value is 10. The suggested setting range: fall-scale AD result×0.3~1%.

User doesn't need to change this value. However, if the system is interfered greatly by outside, this value should be increased modestly to avoid wrong judgment of positive and negative movement. If this value is too large, the PID control period (sampling time) got from the auto tune process will be too long. As the result do not set this value too large.

※1：If users have no experience, please use the default value 10, set PID sampling time (control period) to be 0msthen start the auto tune.

PID auto tune overshooting permission setting [S3+13]

If set 0, overshooting is permitted, and the system can study the optimal PID parameters all the time. But in auto tune process, detected value may be lower or higher than the target value, safety factor should be considered here.

If set 1, overshooting is not permitted. For these objectives which have strict safety demand such as pressure vessel. Set **[S3+13]** to be 1 to prevent from tested value over the target value seriously.

In the process, if **[S3+2]** bit8 changes from 0 to 1, it means the auto tune is successful and the optimal parameters are got; if **[S3+2]** bit8 keeps 0, when **[S3+2]** bit7 changes from 1 to 0, it means auto tune is finished, but the parameters are not the best and they need to be modified by hand.

Every adjustment percent of current target value in auto tune end transition stage [S3+14]

This parameter is effective only when **[S3+13]** is 1.

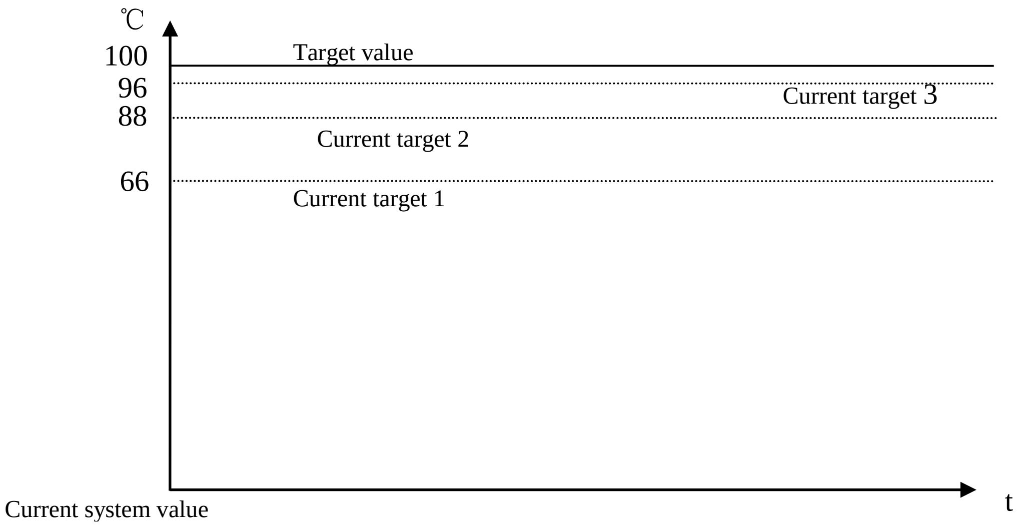
If doing PID control after auto tune, small range of overshooting may be occurred. It is better to decrease this parameter to control the overshooting. But response delay may occur if this value is too small. The defaulted value is 100% which means the parameter is not effective. The recommended range is 50~80%.

Cutline Explanation:

Current target value adjustment percent is 2/3（**S3 + 14** = 67%), the original temperature of the system is 0 ℃, target temperature is 100 ℃, and the current target temperature adjustment situation is shown as below:

Next current target value = current target value + (final target value – current target value) × 2/3;

So the changing sequence of current target is 66 ℃, 88 ℃, 96 ℃, 98 ℃, 99 ℃, 100 ℃.



Over target value times in auto-tuning end transition stage when limiting the overshoot [S3+15]

This parameter is valid only when [S3+13] is 1;

If entering into PID control directly after auto tune, small range of overshoot may occur. It is good to prevent the overshoot if increasing this parameter properly. But it will cause response lag if this value is too large. The default value is 15 times. The recommended range is from 5 to 20.

7-5. Advanced Mode

Users can set some parameters in advanced mode in order to get better PID control effect. Enter into the advanced mode, please set **[S3+2]** bit 15 to be 1, or set it in the XDP Pro software.

Input Filter constant [S3+8]

It will smooth the sampling value. The default value is 0%, which means no filter.

Differential Gain[S3+9]

The low pass filtering process will relax the sharp change of the output value. The default value is 50%; the relaxing effect will be more obviously if increasing this value. Users do not need to change it.

Upper-limit and lower-limit value [S3+10], [S3+11]

Users can choose the analog output range via setting this value.

Default value: lower-limit output =0

Upper-limit =4095

7-6. Application outlines

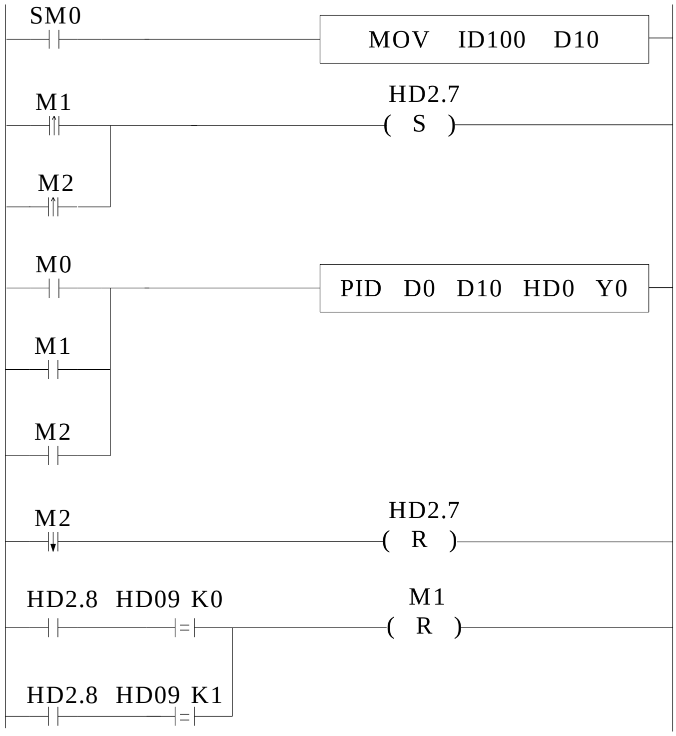
Under the circumstances of continuous output, the system whose effect ability will die down with the change of the feedback value can do auto tune, such as temperature or pressure. It is not suitable for flux or liquid level.
Under the condition of overshooting permission, the system will get the optimal PID parameters from auto tuning.
Under the condition that overshoot not allowed, the PID parameters got from auto tune is up to the target value, it means that different target value will produce different PID parameters which are not the optimal parameters of the system and for reference only.
If the auto tune is not available, users can set the PID parameters according to practical experience. Users need to modify the parameters when debugging. Below are some experience values of the control system for your reference:

- Temperature system: P (%) 2000 ~ 6000, I (minutes) 3 ~ 10, D (minutes) 0.5 ~ 3
- Flux system: P (%) 4000 ~ 10000, I (minutes) 0.1 ~ 1
- Pressure system: P (%) 3000 ~ 7000, I (minutes) 0.4 ~ 3
- Liquid level system: P (%) 2000 ~ 8000, I (minute) 1 ~ 5

7-7. Application

Example 1:

PID control program is shown below:



// Move ID100 content into D10

// auto tune mode, or set to autotune mode after auto tune end

// start PID, D0 is target value, D10 is the measured value, from HD0 is PID parameters area; output PID result byY0

// PID control finish, close auto tune PID mode

// if auto tune is successful, and overshoot is permitted, close auto tune control bit, auto tune will finish;

If auto tune turns to be manual mode, and overshoot is not permitted, close auto tune control bit.

Soft element function comments:

HD2.7: Auto tune bit

HD2.8: Successful flag of auto tune

M0: Normal PID control

M1: Auto tune control

M2: Enter PID control after auto tune

Operation steps:

1. Send the actual temperature to PID collection register
2. Set probably value for P, I, D, sampling period
3. Set ON auto tune control bit M1 to startup PID auto tune
4. M1 will be reset after the auto tune is finished
5. Set ON M0, use the PID parameters getting from auto tune
6. If the PID effect is not good by using the auto tune PID parameters, user can adjust the PID parameters to get good effect.

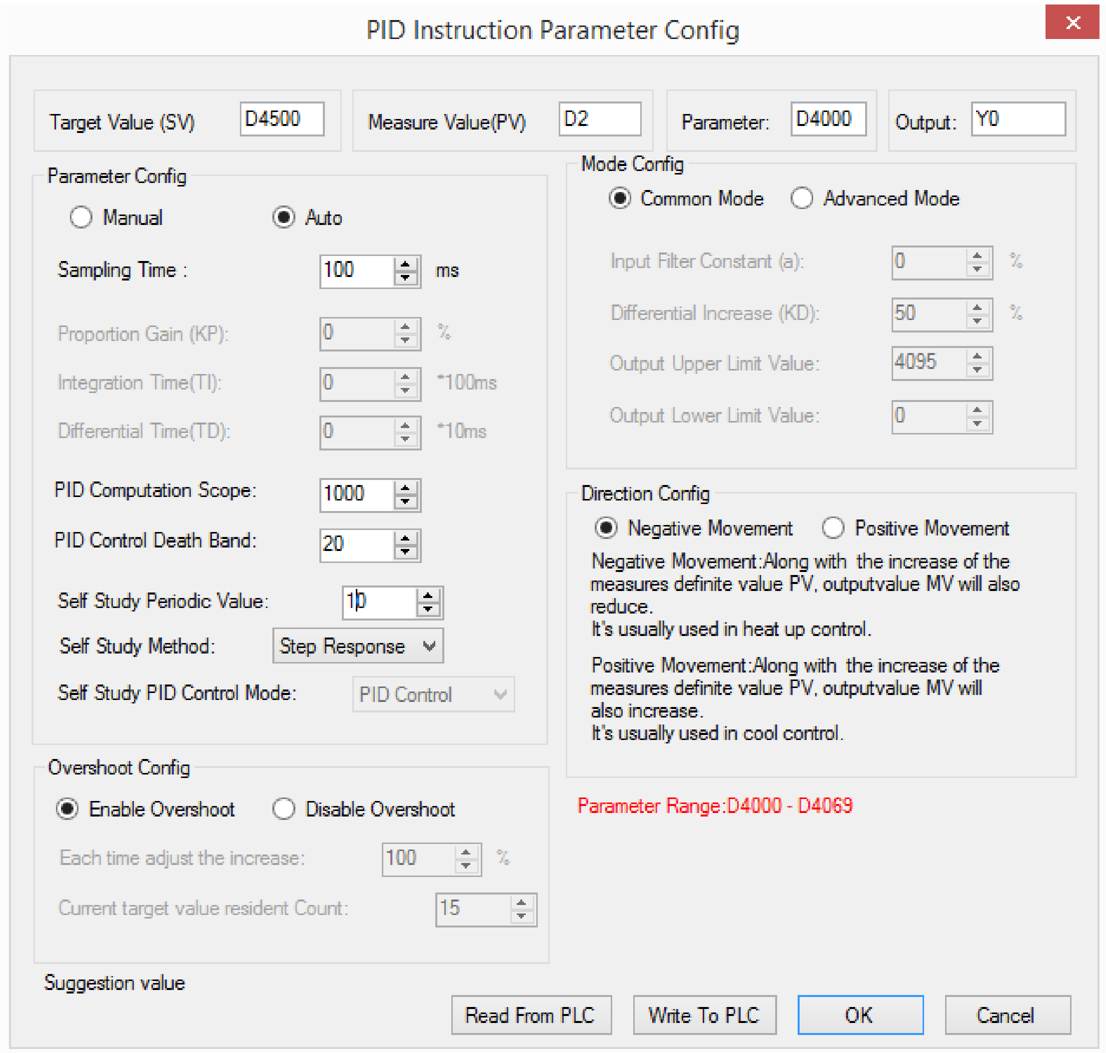
Note: This PLC temperature PID control program is applicable to almost all temperature control projects.

Example 2:

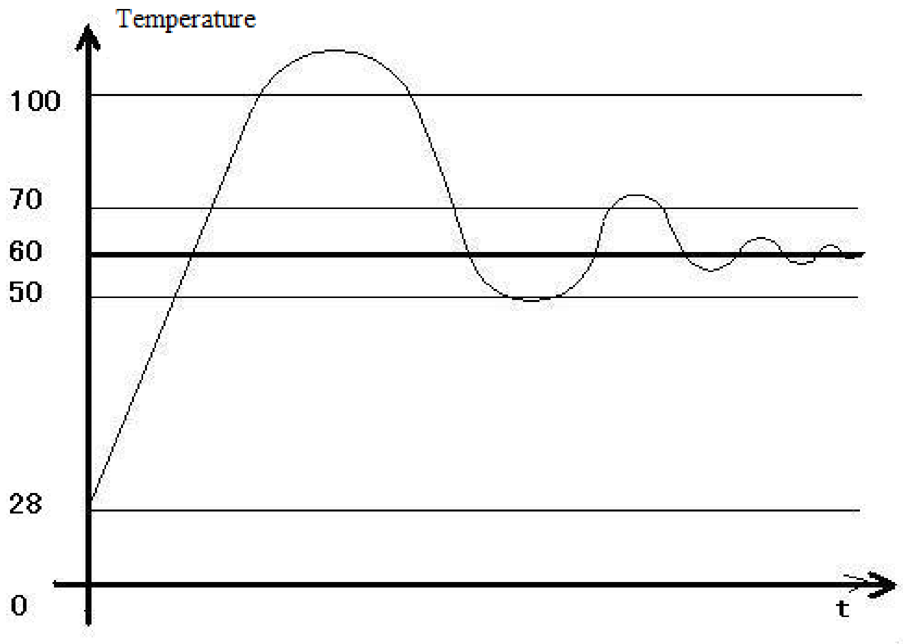
To control the target temperature 60℃ in step response mode.

Overshoot is permitted:

1. The target temperature 60℃ (600)
2. Parameters setting



3. The result curve



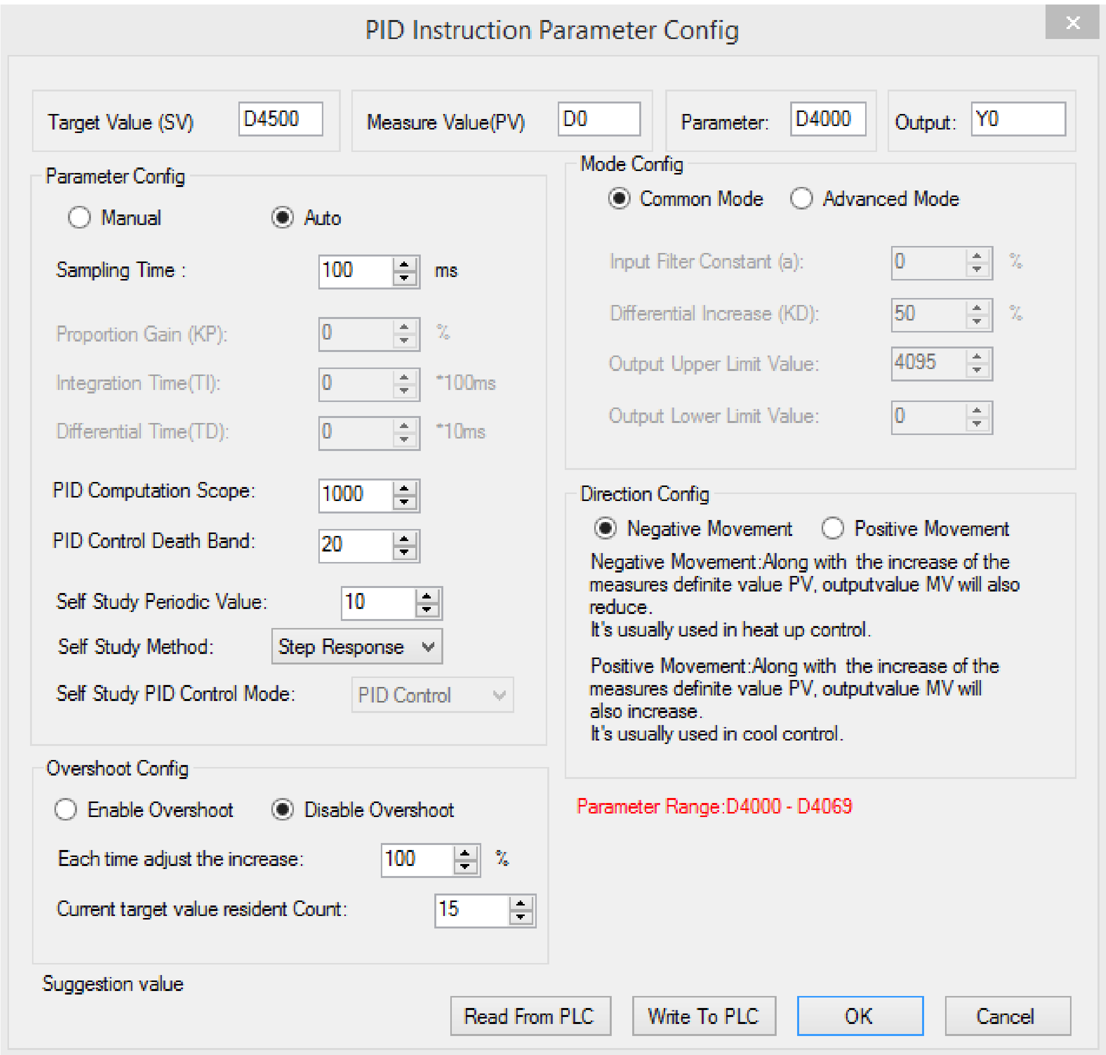
Explanation:
The target temperature is 60 degree, PID calculation range is 10 degree, PID control dead area is 0.2 degree, auto tune period changing value is 10. When the PID control works in normal atmospheric temperature, the PID output terminal will heat the temperature from 28 to 100 degree, then the output stops, the temperature keeps increasing to 110 degree (max temperature) as the remaining warmth. Then the temperature keeps decreasing to 60 degree, the output starts to heat again to 70 degree and stops. The temperature increases a little then decreases again. This process will repeat. Finally, the temperature will fluctuate close the target temperature.

Note:
1. When the temperature reaches 100 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset.
2. When the temperature reaches 100 degree and stops heating, the PID auto tune success bit D4002.8 will be ON at once.
3. When it starts PID calculation, the PLC will auto set a sampling time (about 2500). This parameter will be replaced by the PID best sampling time after stoping heating at 100 degree.
4. When it starts PID calculation, the PLC will auto set the PID parameters (P=4454, I=926, D=2317). These parameters will be replaced by the best PID value after stoping heating at 100 degree.
5. When the temperature reaches 100 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset. At this time, the sampling temperature is higher than target temperature. If user sets ON the PID auto tune again, PLC will get all the PID parameters as 0. Please set ON the PID after the temperature decreases under the normal atmospheric temperature.
6. If PID auto tune start bit and auto tune success bit are power-off retentive, please set or reset them propably to avoid calculation error when starting the PLC next time.

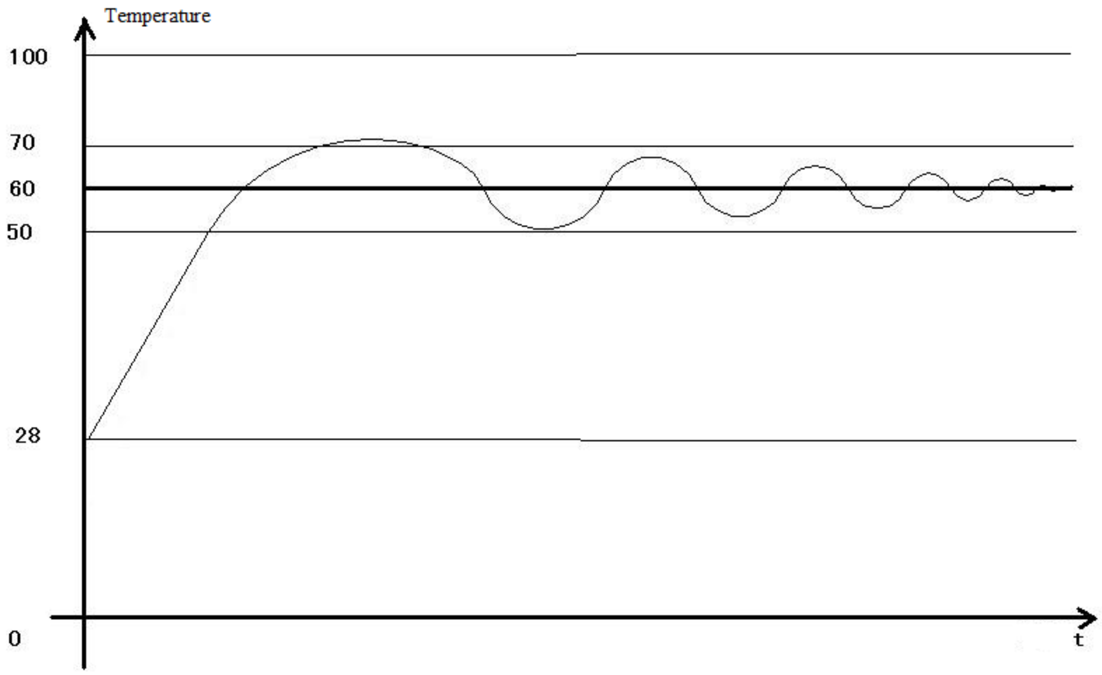
7. The final heating temperature will up to 110 degree when the overshoot is permitted. It is over the target temperature by 50 degree, the overshoot amount is too large.
8. When the PID starts to work, the output will heat the object from 28 degree to 60 degree, then the output is forced to stop heating to avoid overshoot, but this will interrupt the PID auto tune process.
9. To enlarge the PID calculation range can suppress the heating overshoot.

Overshoot is not permitted:

1. The target temperature is 60 degree (600)
2. The related parameter settings:



3. The result curve



Explanation:

The target temperature is 60 degree, PID calculation range is 10 degree, PID control dead area is 0.2 degree, auto tune period changing value is 10. When the PID control works in normal atmospheric temperature, the PID output terminal will heat the temperature from 28 to 48 degree, then the output stops, the temperature keeps increasing to 70 degree (max temperature) as the remaining warmth. Then the temperature keeps decreasing to 60 degree, the output starts to heat again to 62 degree and stops. The temperature increases a little (about 64 degree) then decreases again. This process will repeat. Finally, the temperature will fluctuate close the target temperature. The precision is ± 0.25 degree.

Note:

1. When the temperature reaches 48 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset.
2. When the temperature reaches 48 degree and stops heating, the PID auto tune success bit D4002.8 will not be ON at once. It hasn't set ON even when the auto tune succeeded.
3. When it starts PID calculation, the PLC will auto set a sampling time (about 2500). This parameter will be replaced by the PID best sampling time after stoping heating at 48 degree.
4. When it starts PID calculation, the PLC will auto set the PID parameters (P=4454, I=926, D=2317). These parameters will be replaced by the best PID value after stoping heating at 48 degree.
5. When the temperature reaches 48 degree and stops heating, the PID start bit D4002.7 will not reset at once, it has delay before reset. At this time, the sampling temperature is higher than target temperature. If user sets ON the PID auto tune again, PLC will get all the PID parameters as 0. Please set ON the PID after the temperature decreases under the normal atmospheric temperature.
6. If PID auto tune start bit and auto tune success bit are power-off retentive, please set or reset them propably to avoid calculation error when starting the PLC next time.

7. The final heating temperature will up to 70 degree when the overshoot is permitted. It is over the target temperature by 10 degree, the overshoot amount is small.
8. To enlarge the PID calculation range can suppress the heating overshoot.

8 C Language Function Block

In this chapter, we focus on C language function block's specifications, edition, instruction calling, application points etc. We also attach the common function list.

8-1. Summary

XD, XL supports almost all C language function in XDPPro software (also supports global variable). Users can call the function at many places and call different functions, which greatly increase program security and programmer's efficiency.

8-2. Instruction Format

1. Instruction Summary

Call the C language Function Block at the specified place.

Call the C language function block [NAME_C]			
16 bits instruction	NAME_C	32 bits Instruction	-
Execution condition	Normally ON/OFF, Rising/Falling Edge activation	Suitable Models	XD, XL
Hardware		Software	

2. Operands

Operands	Function	Type
S1	Name of C Function Block, defined by the user	String
S2	Corresponding start ID of word W in C language function	16 bits, BIN
S3	Corresponding start ID of word B in C language function	bit, BIN

3． Suitable Soft Components

Word

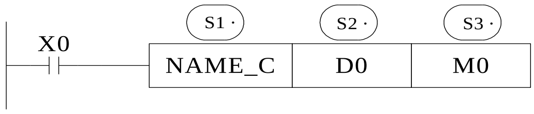
Operands	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S2	●										

Bit

Operands	System						
	X	Y	M	S*	T*	C*	Dn.m
S3			●				

*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS. M includes M, HM, SM; S includes S and HS; T includes T and HT; C includes C and HC.

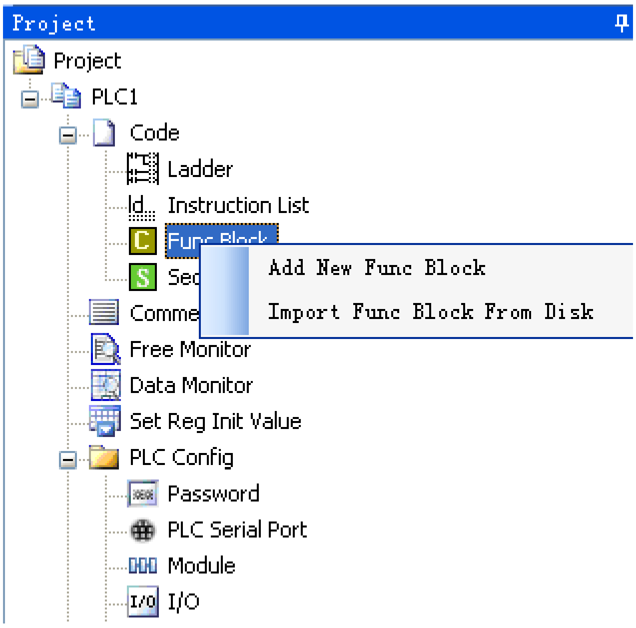
Function and Action



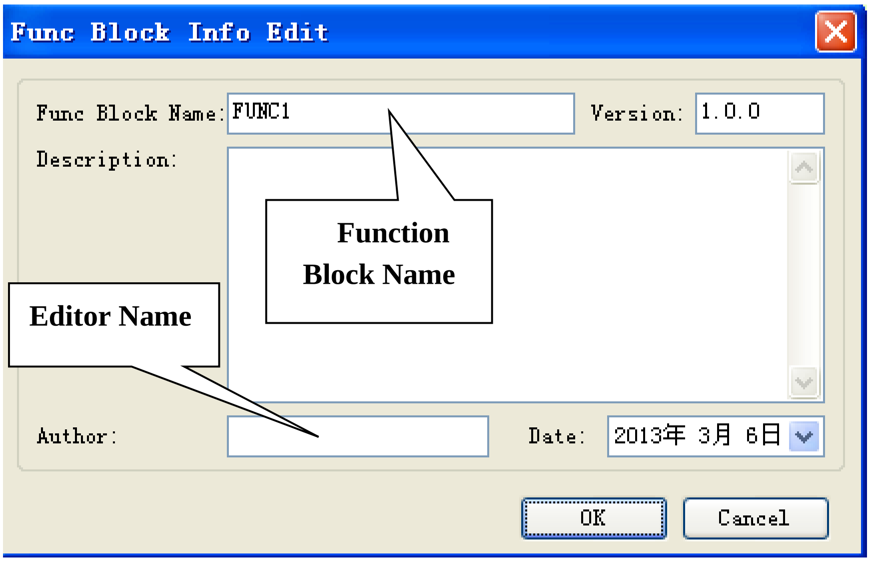
S1 is the function name. It consists of numbers, letters and underlines. The first character can't be number, and the name length should be < =9 ASC.
The name can be the same with PLC's self instructions like LD, ADD, SUB, PLSR etc.
The name can't be the same with the function blocks existing in current PLC;

8-3． Operation Steps

1． Open PLC edit tool, in the left "Project" toolbar, choose "Func Block", right click it and choose "Add New Func Block".

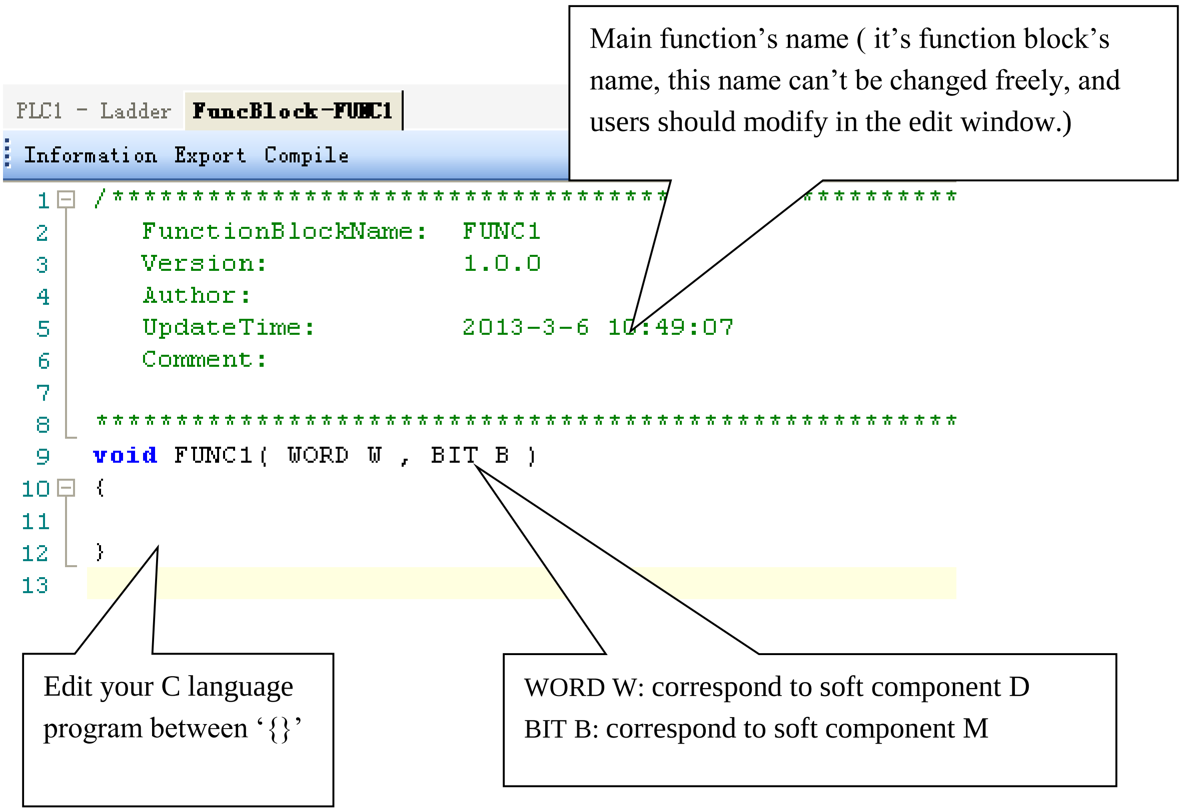


2. See graph below, fill in the information of your function;



Function Block name is the name we use to call the BLOCK. For example: the diagram of FUNC1 should be written as below:

3. After creating the new Function Block, you can see the edit interface as shown below:



- Parameters' transfer way: if call the **Function Block** in ladder, the transferred D and M is the start ID of W and B. Take the above graph as the example, start with D0 and M0, then W[0] is D0, W[10] is D10, B［0］is M0, B［10］is M10; if the used parameters in the ladder are D100, M100, then W[0] is D100, B［0］is M100; if the parameters in the ladder are HD0, HM0, then W[0]=HD0,B[0]=HM0; if the parameters in the ladder are D100, HM100, then W[0]=D100, B[0]=HM100. So, word and bit components start address are defined in PLC program by the user.

Note: The coil and data type in one C language should be the same. All the coils in C language are power loss retentive, or not power loss retentive; so is the same with data register.

- Parameter **W**: represent **Word** soft component, use it in the form of data group. E.g W[0]=1; W[1]=W[2]+W[3]; in the program, use soft components according to standard C language rules.
- Parameter **B**: represent **Bit** soft component, use it in the form of data group. Support **SET** and **RESET**. E.g: B[0]=1; B[1]=0; And assignment, for example, B[0]=B[1].
- Double word operation: add **D** in front of **W**. E.g. DW[10]=100000, it means assignment to double-word W[10]W[11]. Double-word operation: Support the definition of floating variable in the function, and execute floating operation; （E.g: float register D0(double word) means FW[0], FW[0]=123.456）
- Other soft elements definition in C language:

In C language of PLC, if you want to use input(X) and output(Y), then macro definition '#define SysReg Addr_X_Y' is needed; E.g: send the state of input X0 to given coil M0, then B[0]=X[0]; send the state of Y0 to given coil M10, then: B[10]=Y[0];（Note: corresponding X Y in C language is decimal, not Octonary number）.

Note: Marco definition #define SysRegAddr_X_Y should be behind the variable definition, otherwise, it will be error.

Eg.
```
int a,b,c;
#define SysRegAddr_Y
b=3000;
c=W[1030];
a=b+c;
if(B[a]==1)
  Y[3]=0;
```

In a similar way, if the not-power-loss-retentive flow S, Counter C, timer T, counter register TD is in the C language, macro definition '#define SysRegAddr_S_C_T_CD_TD' is also needed; if the power-loss-retentive flow HS, counter HC, timer HT, counter register HCD, timing register HTD etc, macro definition '#define SysRegAddr_HS_HC_HT_HCD_HTD' is needed.

E.g: W[0]=CD[0];W[1]=TD[0];B[1]=C[0];B[2]=T[0];

- Function Library: In **Function Block**, users can use the Functions and Constants in function library directly. For the Functions and Constants in function library, see 9-8.
- The other data type supported:

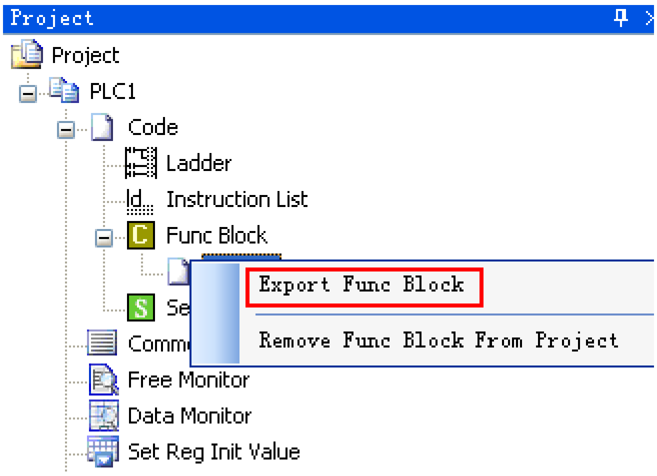
```
BOOL;                //BOOL Quantity
```

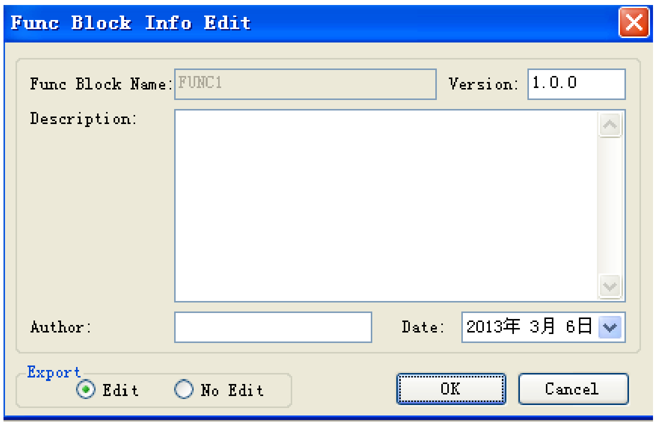
INT8U; //8 bits unsigned integer
INT8S; //8 bits signed integer
INT16U //16 bits unsigned integer
INT16S //16 bits signed integer
INT32U //32 bits unsigned integer
INT32S //32 bits signed integer
FP32; // single precision floating
FP64; //double precision floating
Predefined Marco: #define true 1
#define false 0
#define TRUE 1
#define FALSE 0

8-4. Import and Export the Functions

1. Export

(1) Function: Export the function as the file, then other PLC program can import to use;





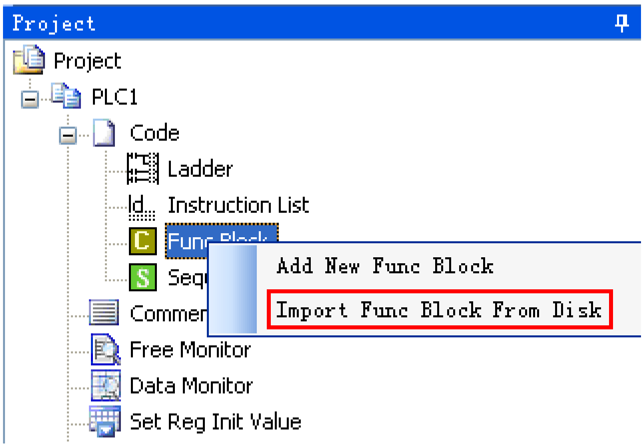
(2) Export Format

a) Editable: Export the source codes out and save as a file. If import again, the file is editable;

b) Not editable: Don’t export the source code, if import the file, it’s not editable;

2. Import

Function: Import the existing **Func Block** file, to use in the PLC program.



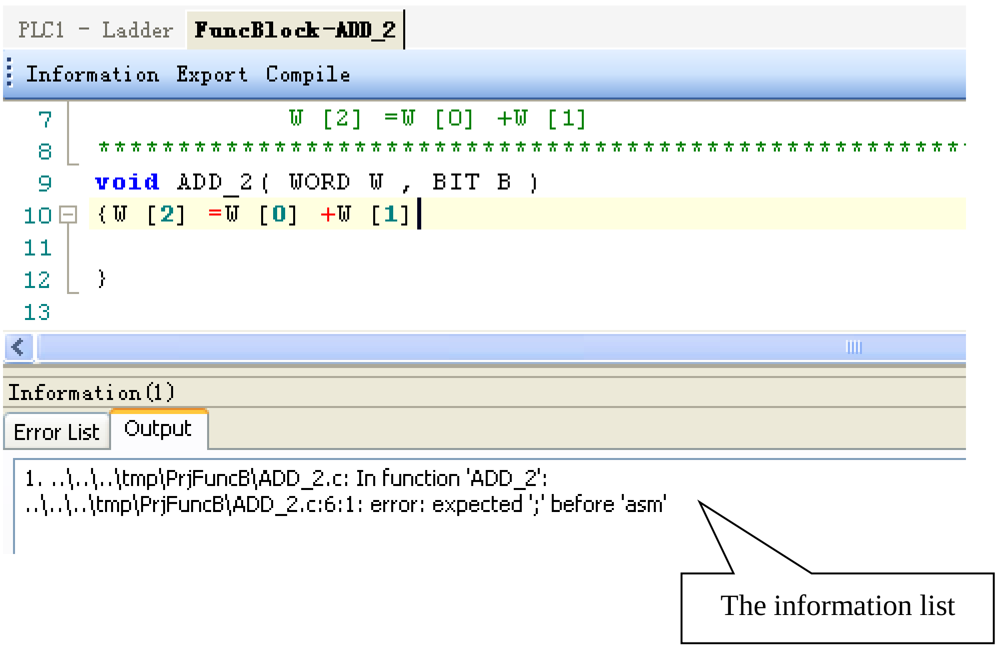
Choose the **Func Block**, right click ‘Import Func Block from Disk’, choose the correct file, and then click OK.

8-5. Edit the Func Blocks

Example: Add D0 and D1 in PLC’s registers, and then assign the value to D2;

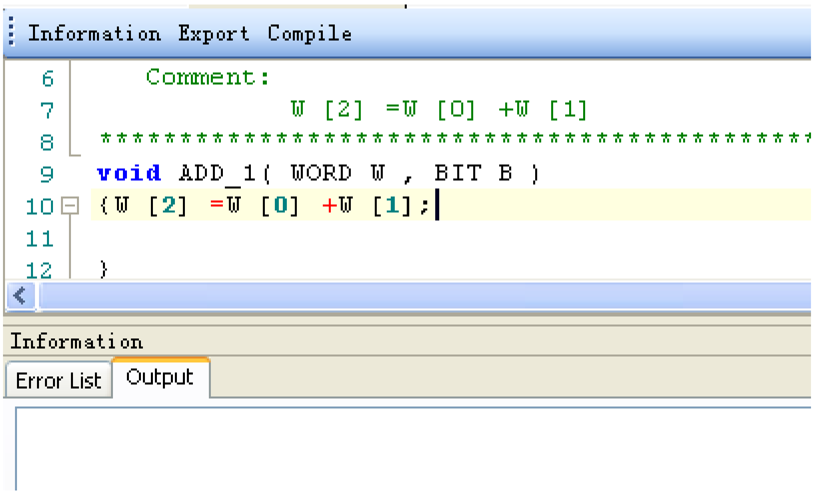
(1) In ‘Project’ toolbar, new create a **Func Block**, here we name the **Func Block** as **ADD_2**, then edit C language program;

(2) Click ‘compile' after edition.

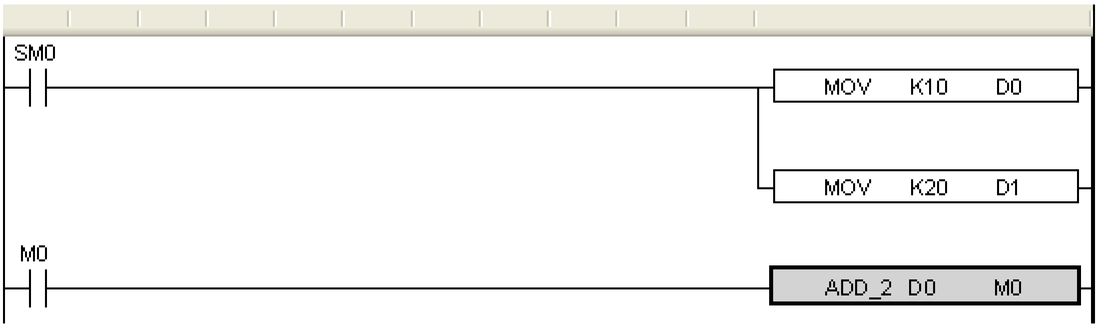


According to the information shown in the output blank, we can search and modify the grammar error in C language program. Here we can see that in the program there is no ‘;’ sign behind W [2] =W [0] + W [1].

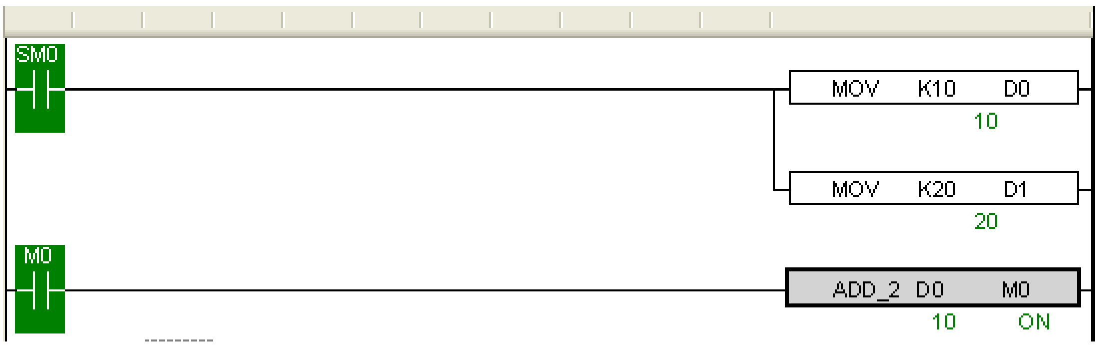
Compile the program again after modifying the program. In the information list, we can confirm that there is no grammar error in the program.



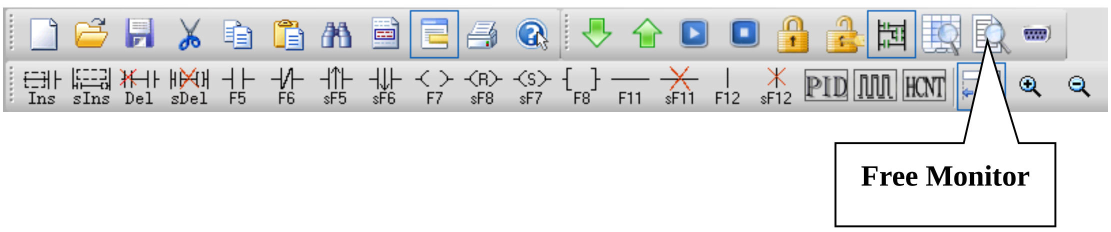
(3) Write PLC program, assign value 10 and 20 into registers D0, D1 separately, then call Func Block ADD_2, see graph below:



(4) Download program into PLC, run PLC and set M0.



(5) From Free Monitor in the toolbar, we can see that D2 changes to be 30, it means assignment is successful;



8-6. Program Example

If PLC needs to do complicated calculation (including plus and minus calculation), the calculation will be used for many times, C language function is easy to use.

Example 1:

Calculation a= b/c + b*c+(c-3)*d

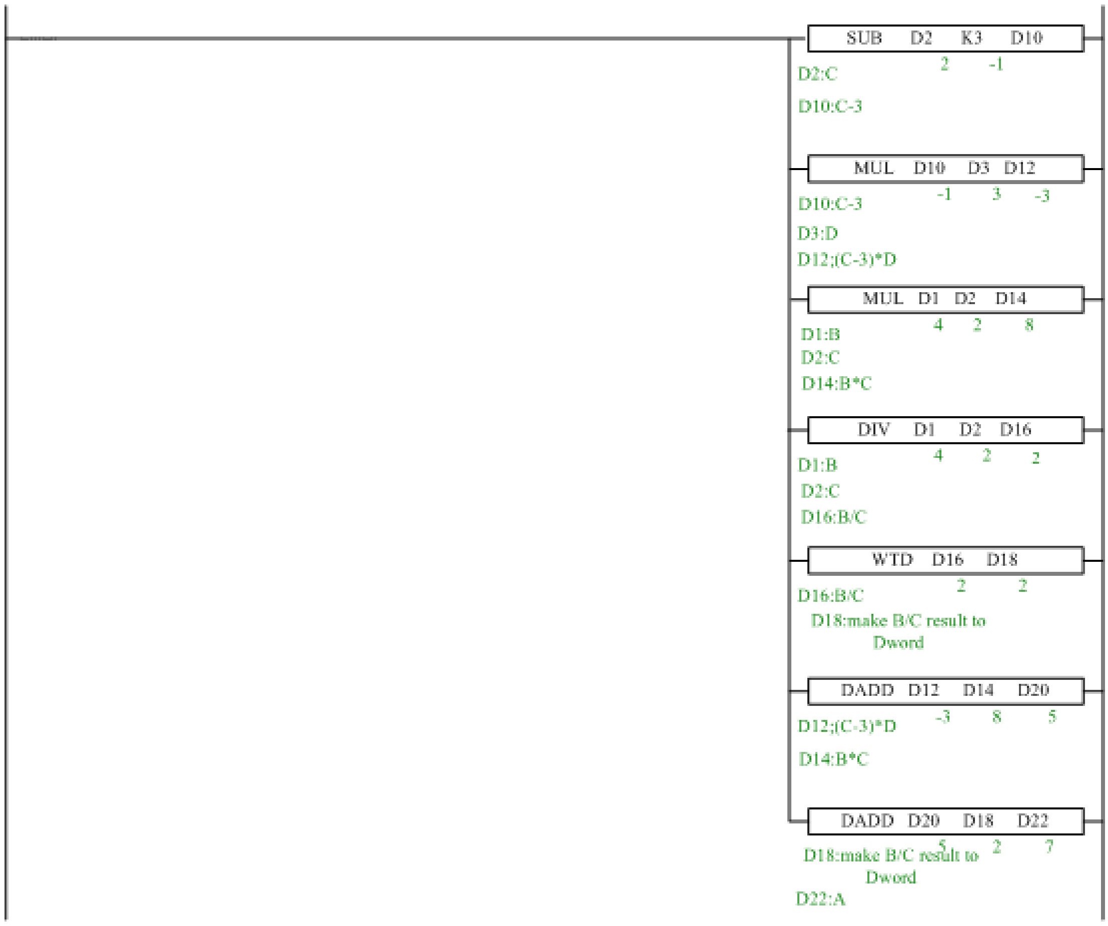
Method 1: use ladder chart:

Get the result of c-3

Get the result of three multiplication equations

Get the sum

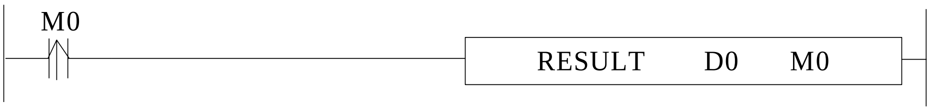
Ladder chart only support two original operands, it needs many steps to get the result.



Note:

1. The result of MUL is Dword, the result is stored in D14~D15.
2. The result of DIV has quotient D16 and remainder D17. If D17 has value, the calculation precision will decrease. Please use float format to ensure the precision.
3. D16 quotient is word value, in plus calculation all the data should be changed to Dword. The final result is stored in D22~D23.

Method 2: use C language:



RESULT	Function name
D0	In the function, W [0] =D0, W [1] =D1… If D0=D32, then W [0] =D32, W [1] =D33… If S2=HD32, then W [0] =HD32, W [1] =HD33…
M0	In the function, B [0] = M0, B [1] =M1… If S2=M32, then B [0] = M32, B [1] =M33… If S2=HM32, then B [0] = HM32, B [1] =HM33…

C program

```
void RESULT( WORD W , BIT B )
{
long int a,b,c,d;;
b=W[1];
c=W[2];
d=W[3];
a=b/c+b*c+(c-3)*d;
DW[4]=a;
}
```

Method 2 can simplify the program.

The above C language function is similar to ladder chart of method 1, whose precision is not high. If it needs to get the high precision, please use float calculation.

Example 2: Calculate CRC parity value via Func Block

CRC calculation rules:

（1）Set 16-bit register (CRC register) = FFFF H

（2）XOR (Exclusive OR) the first 8-bit byte message and the low 16-bit CRC register.

（3）Right shift 1 bit of CRC register, fill 0 into the highest bit.

（4）Check the right shifted value, if it is 0, save the new value from step3 into CRC register; if it is not 0, XOR the CRC register value with A001 H and then save the result into the CRC register.

（5）Repeat step3&4 until all the 8-bit have been calculated.

（6） Repeat step（2）~（5）, then calculate the next 8-bit message. Until all the messages have been calculated, the result will be the CRC parity code in CRC register.

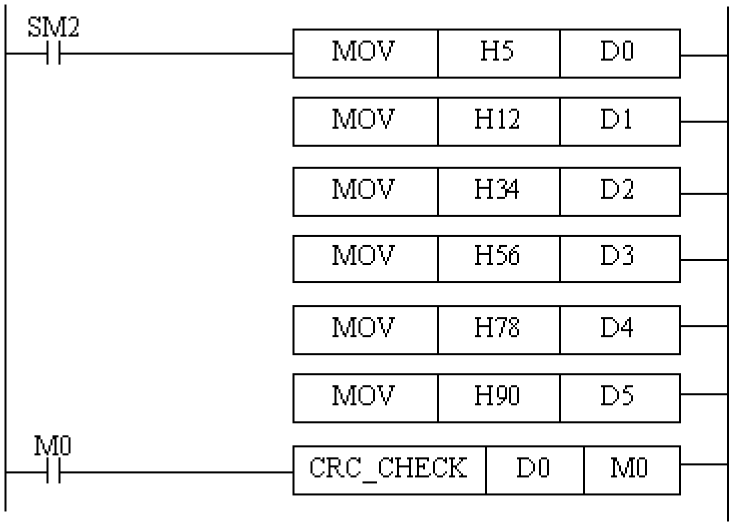
Edit C language Function Block program, see graph below:

```
void CRC_CHECK( WORD W , BIT B )
{
      int i,j,m,n;
      unsigned int reg_crc=0xffff,k;

      for( i = 0 ; i < W[0] ; i++ )
            {
            reg_crc^=W[i+1];
            for(j=0;j<8;j++)
            {
            if(reg_crc&0x01)
                  reg_crc=(reg_crc>>1)^0xa001;
            else
                  reg_crc=reg_crc>>1;
            }
            }

            m=W[0]+1;
            n=W[0]+2;
            k=reg_crc&0xff00;
            W[n] = k>>8;
            W[m]=reg_crc&0xff;
          }
```

Edit PLC ladder program,

D0: Check byte number of data,

D1～D5: Check data content. See graph below:



Download to PLC, then RUN PLC, set M0, via Free Monitor, we can find that values in D6 and D7 are the highest and lowest bit of CRC parity value;

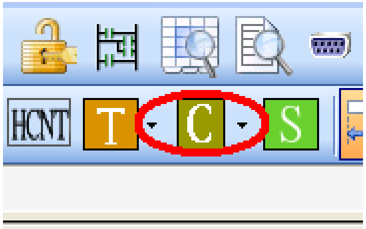
8-7. Application notes

In one Func Block file, you can write many functions, and they can be called by each other. Each Func Block file is independent, they can't call block in each other;

Func Block files can call C language library function in form of floating, arithmetic like sin, cos, tan.

XC series PLC only support local variable, while XD/XL series PLC support both local and global variable. This makes C language Block more flexible and convenient.

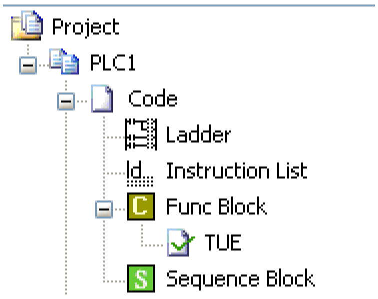
XDPPro software v3.3 and later version keep C function library:



In this function block, user can call the C function directly:

TCA	Calculation area of a circle
TCC	Circumference calculation
TCRC	CRC Check
TDSL	Input data (short) from big to small order
TDSS	Input data (short) from small to large order
TECA	Calculation area of a circle
TECC	Circumference calculation
TEEX	Exponentiation calculation
TEL10	Natural logarithm
TELO	Natural logarithm
TEPTH	Known two right-angle sides and the hypotenuse demanded
TEPTR	Known one right-angle side and hypotenuse need to demand the other right-angle side
TEQE	Quadratic equation (float)
TESUM	Sum of memory 32-bit floating data
TETP	The product of memory data (float)
TEUE	Quadratic equation (float)
TEX	Exponentiation calculation
TFA	Factorial solving
TITF	Inverse trigonometric functions
TQE	Quadratic equation (short)
TSUM	Sum of memory 32-bit integer data
TTP	The product of memory data (short)
TUE	Quadratic equation (short)

For example: click TEL10, the function name will show on the project bar:



User can call it in the ladder chart editing window at any time.

8-8. Function Table

The default function library

Constant	Data	Description
_LOG2	(double)0.693147180559945309417232121458	Logarithm of 2
_LOG10	(double)2.302585092994045901093613792909 3	Logarithm of 10
_SQRT2	(double)1.41421356237309504880168872421	Radical of 2
_PI	(double)3.14159265358979323846264338327 95	PI
_PIP2	(double)1.570796326794896619231321691639 75	PI/2
_PIP2x3	(double)4.712388980384689857693965074919 25	PI*3/2

String Function	Description
void * memchr(const void *s, int c, size_t n);	Return the first **c** position among **n** words before **s** position
int memcmp(const void *s1, const void *s2, size_t n);	Compare the first **n** words of position **s1** and **s2**
void * memcpy(void *s1, const void *s2, size_t n);	Copy **n** words from position **s2** to **s1** and return **s1**
void * memset(void *s, int c, size_t n);	Replace the **n** words start from **s** position with word **c**, and return to position **s**
char * strcat(char *s1, const char *s2);	Connect string **ct** behind string **s**
char * strchr(const char *s, int c);	Return the first word **c** position in string **s**
int strcmp(const char *s1, const char *s2);	Compare string s1 and s2
char * strcpy(char *s1, const char *s2);	Copy string s1 to string s2

Double-precision math function	Single-precision math function	Description
double acos(double x);	float acosf(float x);	Inverse cosine function
double asin(double x);	float asinf(float x);	Inverse sine function
double atan(double x);	float atanf(float x);	Inverse tangent function
double atan2(double y, double x);	float atan2f(float y, float x);	Inverse tangent value of parameter (y/x)
double ceil(double x);	float ceilf(float x);	Return the smallest double integer which is greater or equal with parameter **x**

double cos(double x);	float cosf(float x);	Cosine function
double cosh(double x);	float coshf(float x);	Hyperbolic cosine function, cosh(x)=(e^x+e^(-x))/2
double exp(double x);	float expf(float x);	Exponent (e^x) of a nature data
double fabs(double x);	float fabsf(float x);	Absolute value of parameter x
double floor(double x);	float floorf(float x);	Return the largest double integer which is smaller or equals with **x**
double fmod(double x, double y);	float fmodf(float x, float y);	If **y** is not zero, return the reminder of floating **x/y**
double frexp(double val, int _far *exp);	float frexpf(float val, int _far *exp);	Break floating data **x** to be mantissa and exponent **x** = m*2^exp, return the mantissa of m, save the logarithm into **exp**.
double ldexp(double x, int exp);	float ldexpf(float x, int exp);	X multiply the (two to the power of n) is x*2^n.
double log(double x);	float logf(float x);	Nature logarithm logic
double log10(double x);	float log10f(float x);	logarithm (log10x)
double modf(double val, double *pd);	float modff(float val, float *pd);	Break floating data X to be integral part and decimal part, return the decimal part, save the integral part into parameter ip.
double pow(double x, double y);	float powf(float x, float y);	Power value of parameter **y** (x^y)
double sin(double x);	float sinf(float x);	sine function
double sinh(double x);	float sinhf(float x);	Hyperbolic sine function, sinh(x)=(e^x-e^(-x))/2
double sqrt(double x);	float sqrtf(float x);	Square root of parameter X
double tan(double x);	float tanf(float x);	Tangent function.
double tanh(double x);	float tanhf(float x);	hyperbolic tangent function tanh(x)=(e^x-e^(-x))/(e^2+e^(-x))

The using method of the functions in the table:

float asinf（float x）;

float asinf: float means the return value is float format;

float x: float means the function formal parameter is float format. In actual using, it do not need to write the float. See line 14 in the following example:

```
9   void ZHENGXIAN( WORD W , BIT B )
10  {
11  int a;
12  float x,y,z;
13  x=FW[0];
14  y=asinf(x);
15  z=180*y/3.14159;
16  a=(int)z;
17  W[2]=a;
18  }
```

9 Sequence BLOCK

This chapter mainly introduces sequence block instruction and the application.

Sequence Block instruction:

Mnemonic	Function	Ladder chart	Chapter
Sequence Block			
SBSTOP	Pause BLOCK	SBSTOP S1 S2	9-6-1
SBGOON	Go to execute BLOCK	SBGOON S1 S2	9-6-1

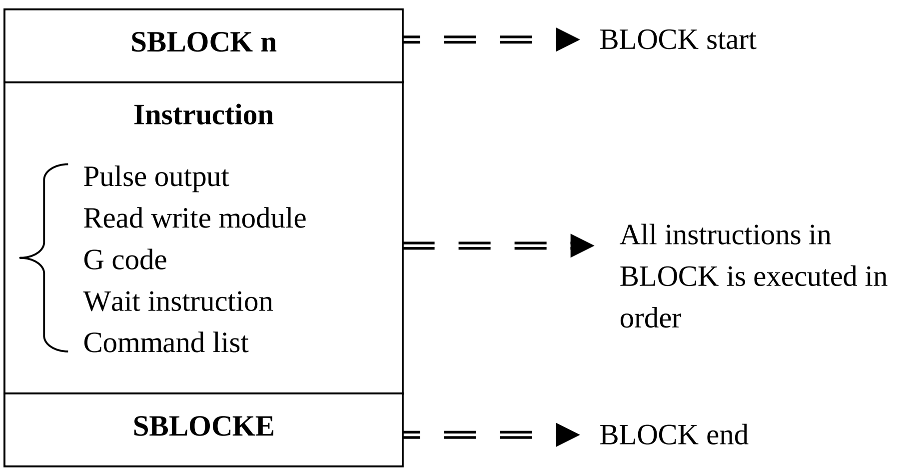
9-1. Concept of the BLOCK

Sequence block whose brief name is BLOCK is a program block to realize some functions. As a special flow, all instructions in the block are executed in order, which is the biggest difference with general processes.

BLOCK starts from SBLOCK and ends with SBLOCKE, and programmers can write instructions in the BLOCK. If one BLOCK contains multiple pulse output instructions（or other instructions）, then pulse output instructions will execute in accordance with conditions meet order; And meanwhile the next pulse output instruction will not execute until the current instruction is over.

The XD3, XDM series PLC supports multiple BLOCKs[※1].

A complete BLOCK structure is shown as below:



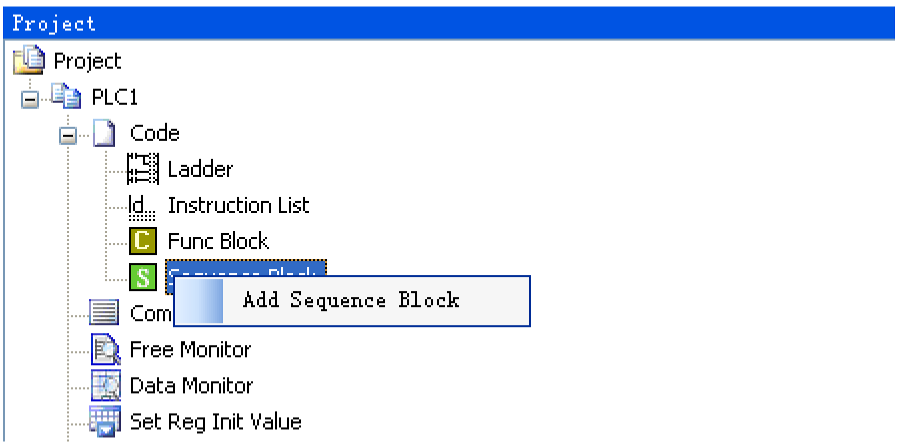
※1: Firmware version below V3.4.5: the XD series PLC allows up to eight BLOCKs. Firmware version V3.4.5 and above: XD/XL series PLC can write up to 100 BLOCKs, but at the same time can only run 8.
※2: When the trigger condition of the BLOCK is triggered by the closure of the normally open coil, it will be executed from the top of the BLOCK to the bottom in turn. When the last instruction is executed, the execution of the BLOCK will be restarted immediately from the top to the bottom. When the trigger condition is disconnected, the BLOCK will not stop immediately, but will complete the last scan and stop after the execution of the unexecuted program.
※3: When the triggering condition of BLOCK is triggered by the rising edge of the coil, the sequential function BLOCK will be executed one time from top to bottom and will not be executed circularly.

9-2． Call the BLOCK

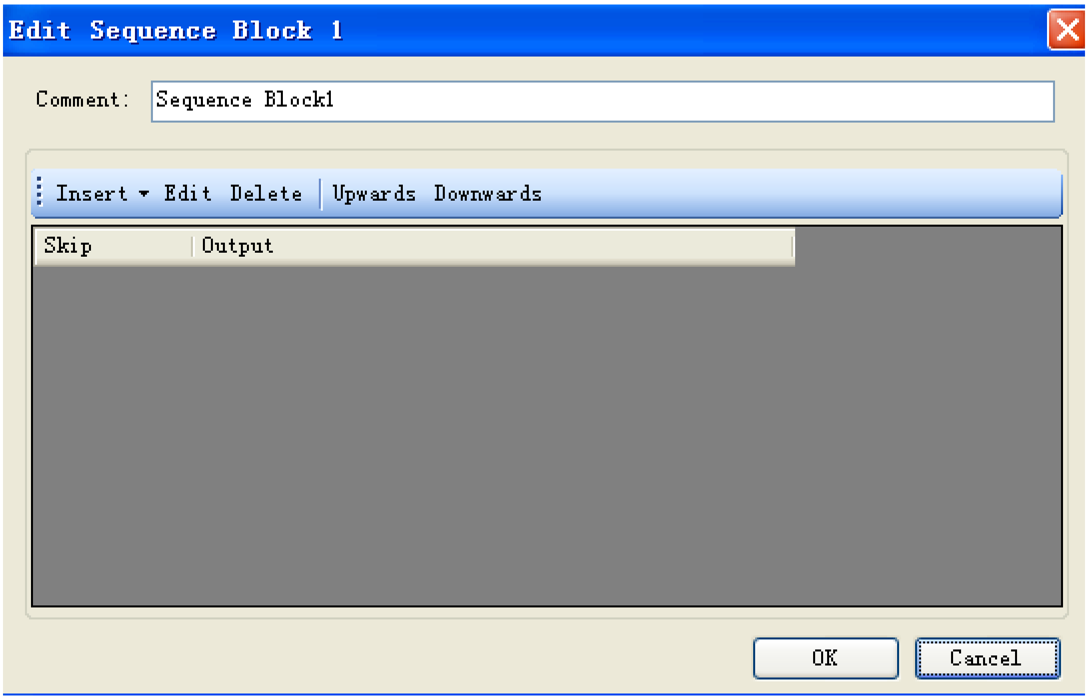
In one program file, it can call many BLOCK; the following is the method to add BLOCK in the program.

9-2-1． Add the BLOCK

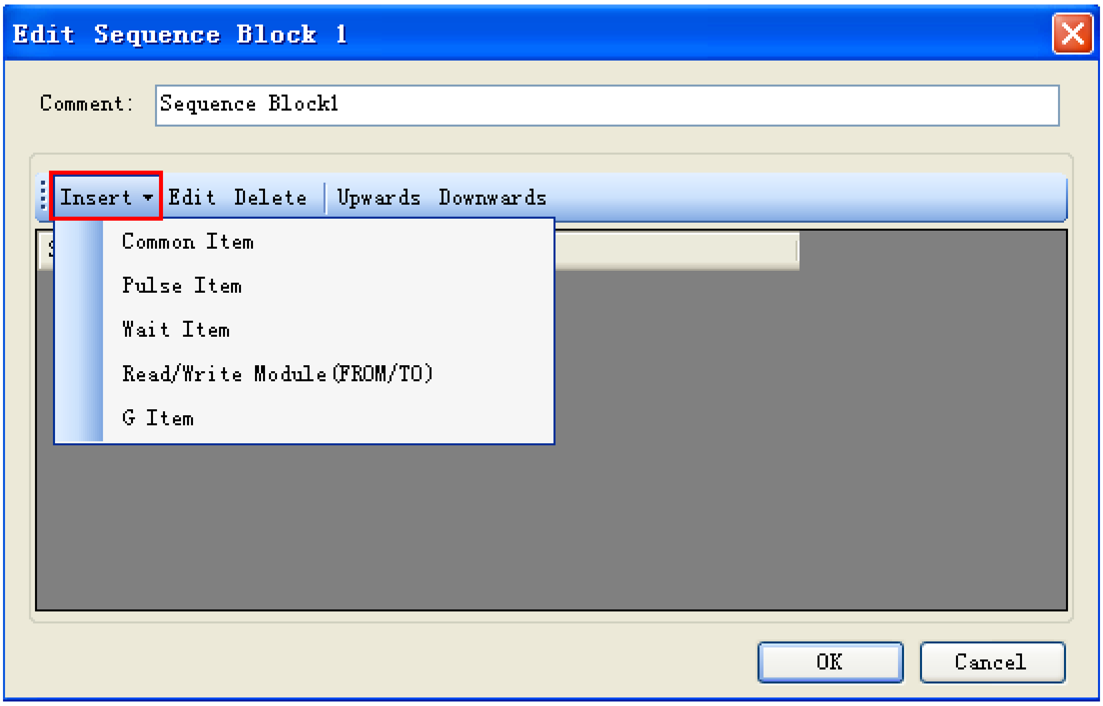
Open XDPPro software, right click the sequence block in the project bar:



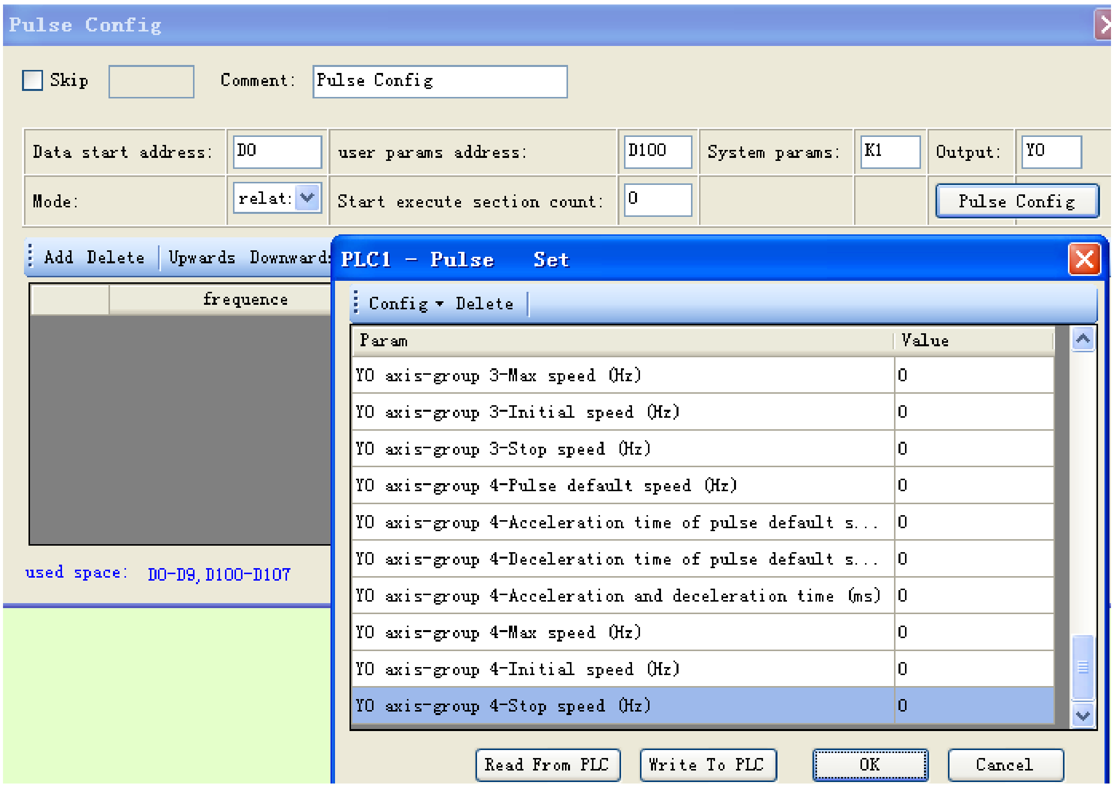
Click the command ‘add sequence block’, the following window will jump out:



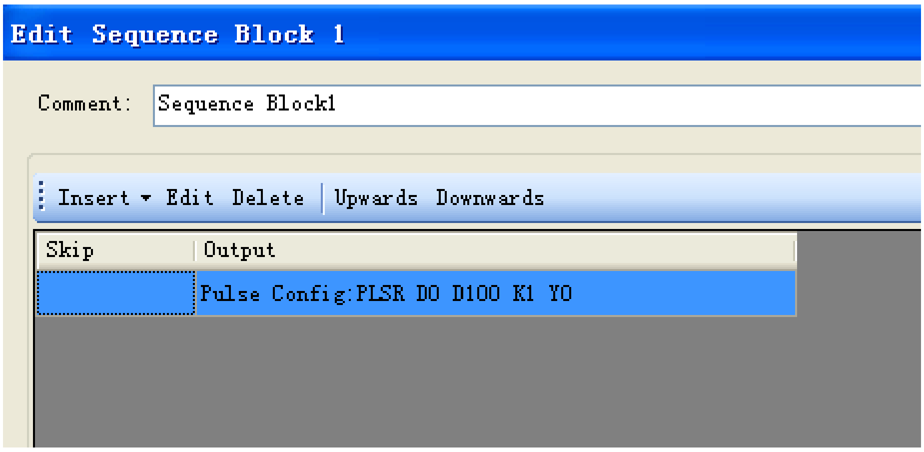
You can edit the BLOCK in the window, Upwards/Downwards are used to change the position of instructions in the block.
Click 'insert' button, some instructions list under the menu:



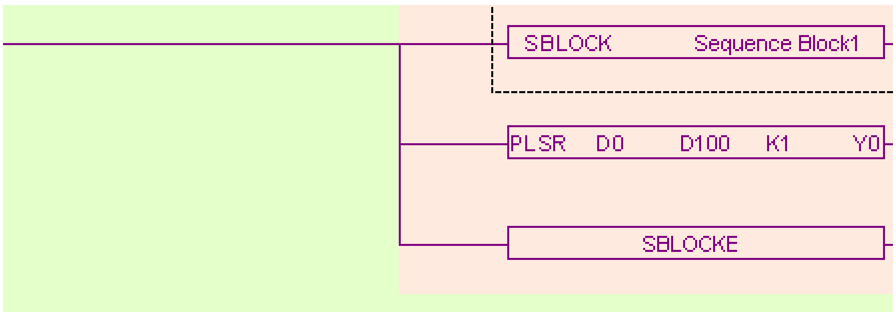
Take 'Pulse Item' for example:



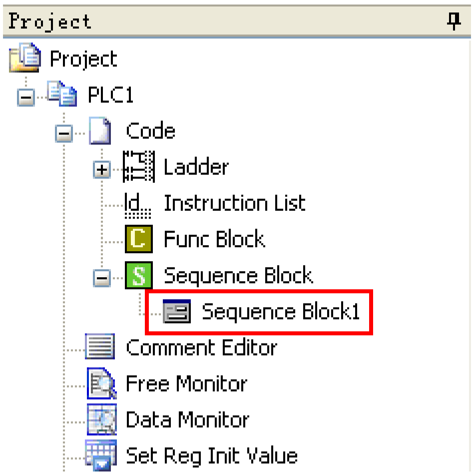
After click 'OK', you will find information in the configuration:



Click 'OK', the following instructions are added in the ladder:

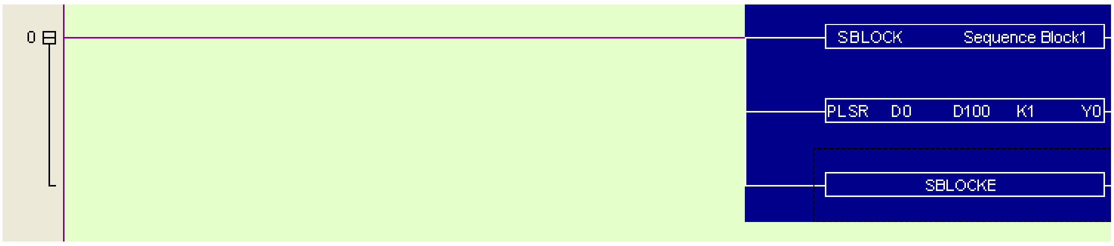


Meantime, a new sequence block is added in the right of the project bar:

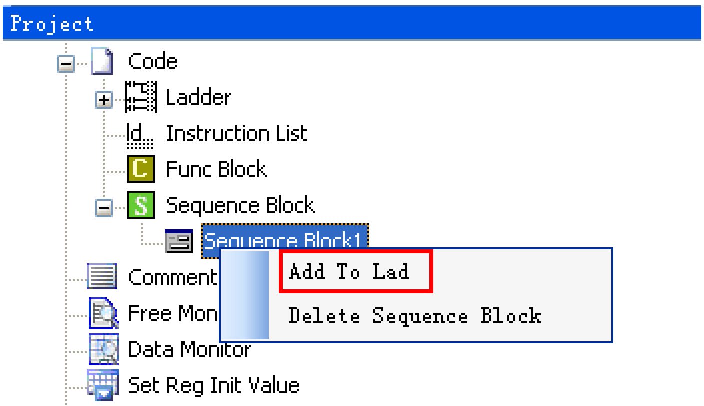


9-2-2. Move the BLOCK

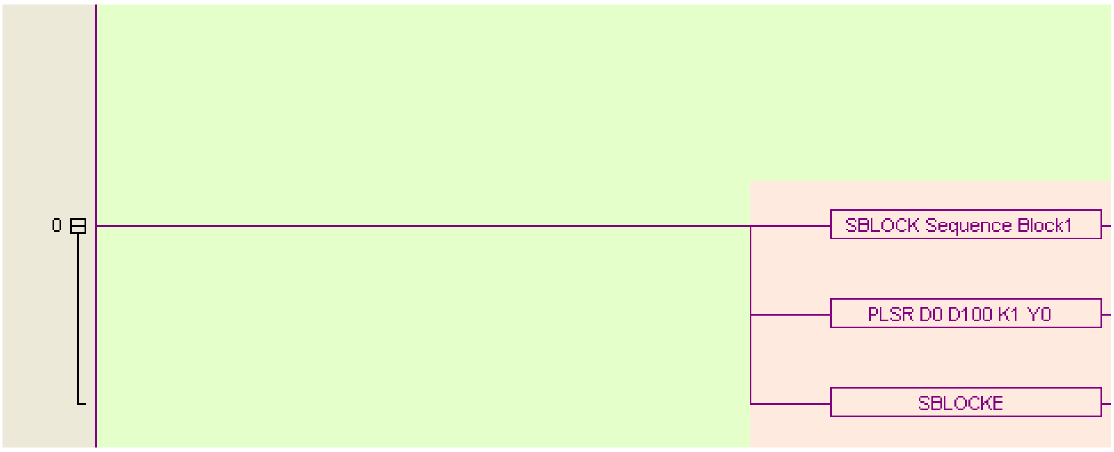
If you want to move the BLOCK to other place, you have to select the original BLOCK and delete it (select all, then delete):



Move the cursor to the new place, and then right click the BLOCK and select 'add to lad':

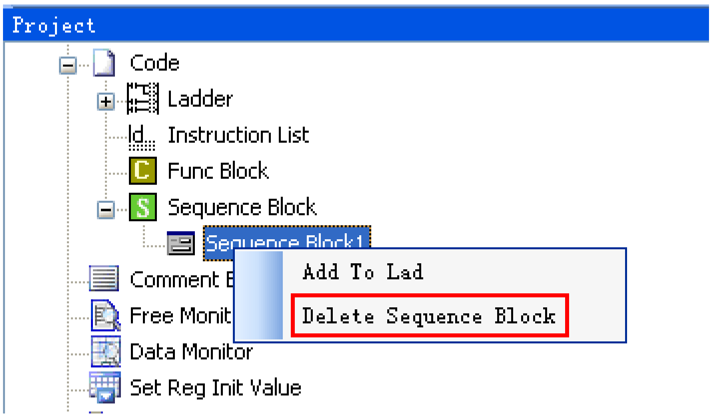


Now the BLOCK is moved to the new place:



9-2-3. Delete the BLOCK

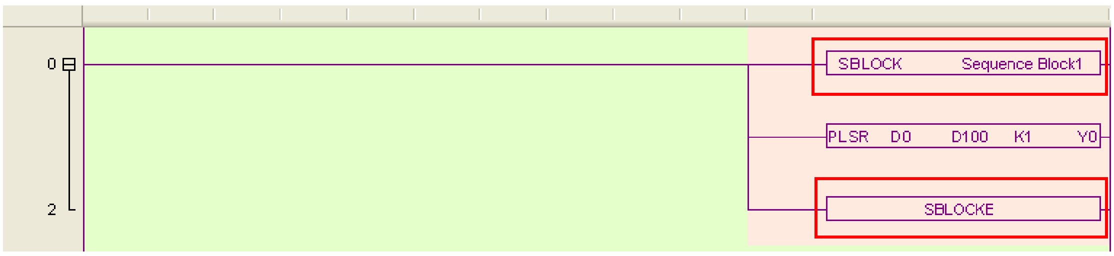
You can select the called BLOCK and delete it. If you want to completely delete the BLOCK, right click the function block and select ‘delete sequence block’. After this operation, you can’t call this BLOCK any more:

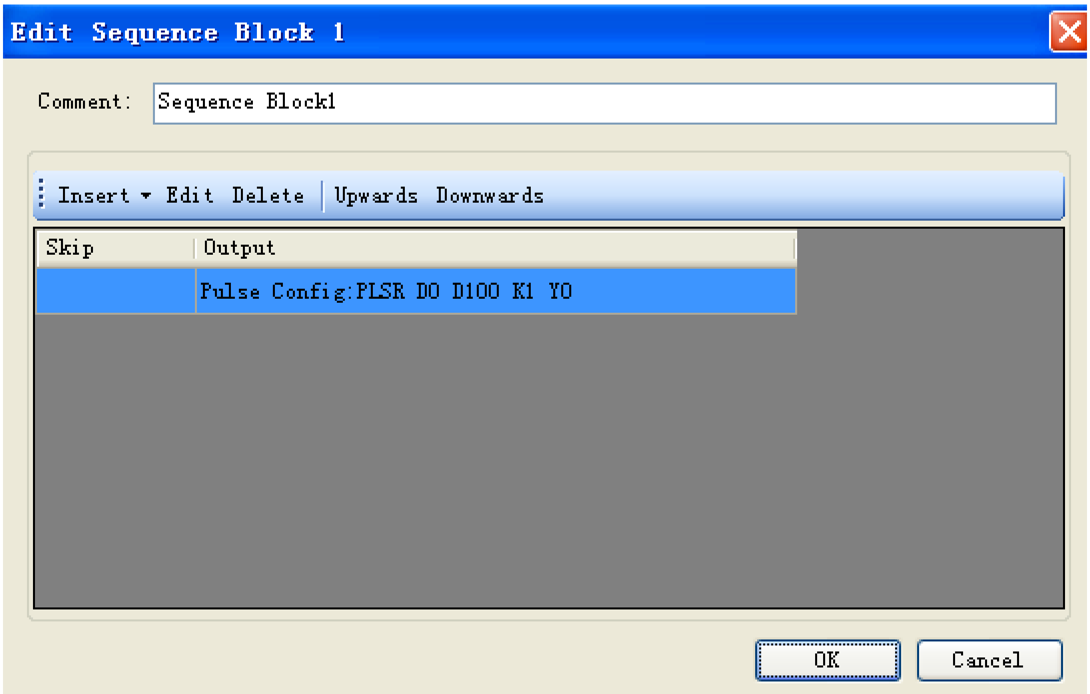


9-2-4. Modify the BLOCK

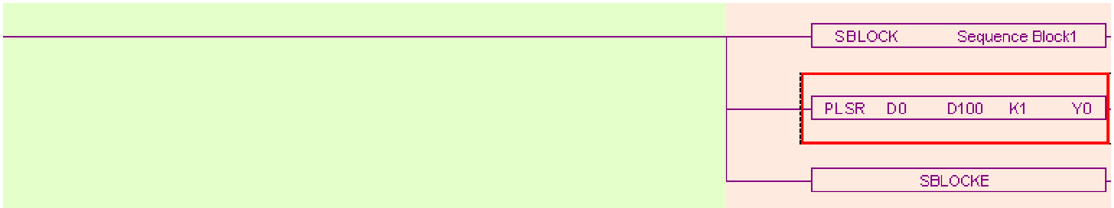
There are two methods to modify the BLOCK.

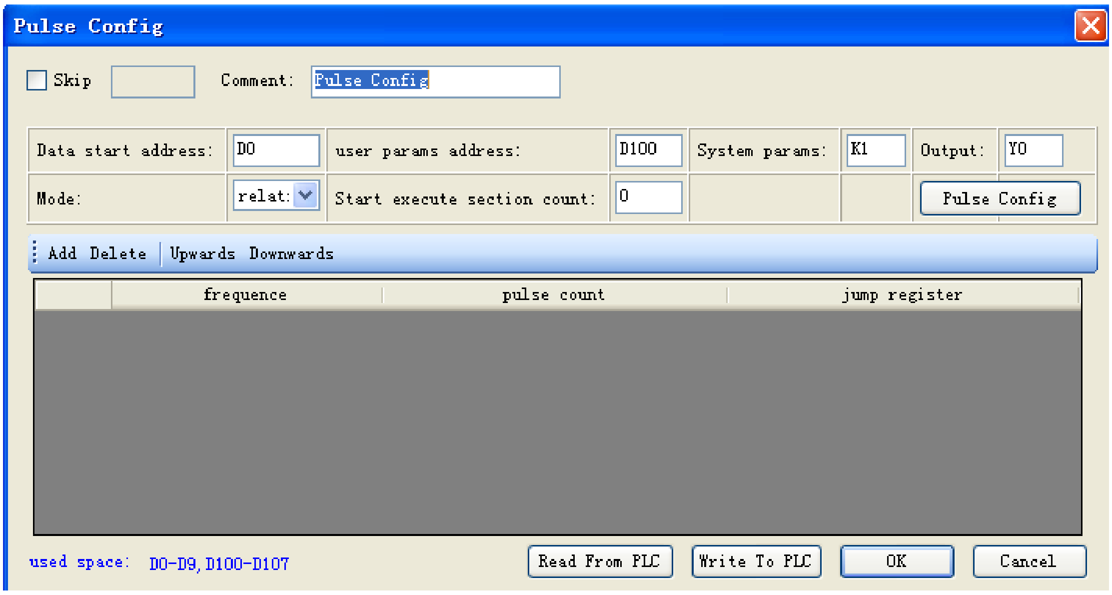
(A) Double click the start/end segment to modify the BLOCK in general:





(B) Double click the middle part to modify :

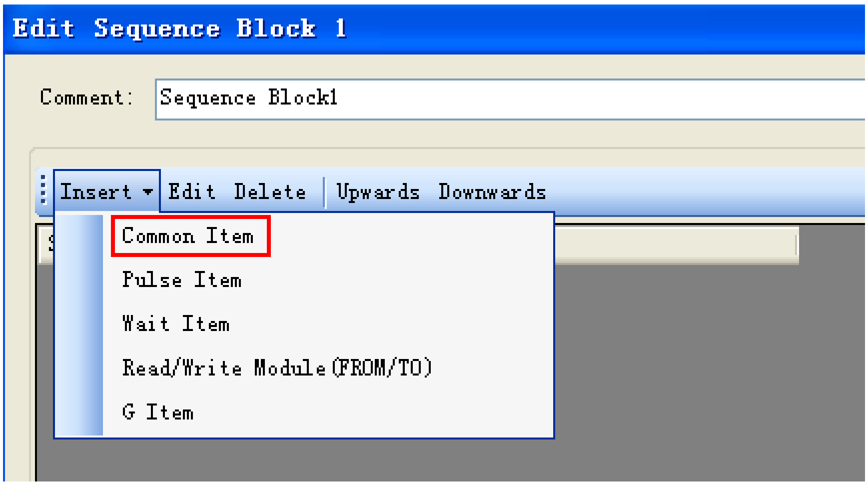




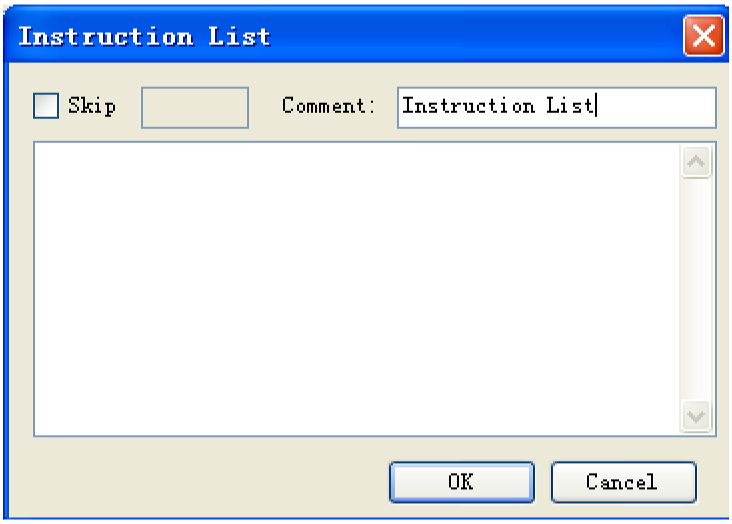
9-3. Edit the instruction of the BLOCK

9-3-1. Command item

Use ‘command item’ to edit the program:



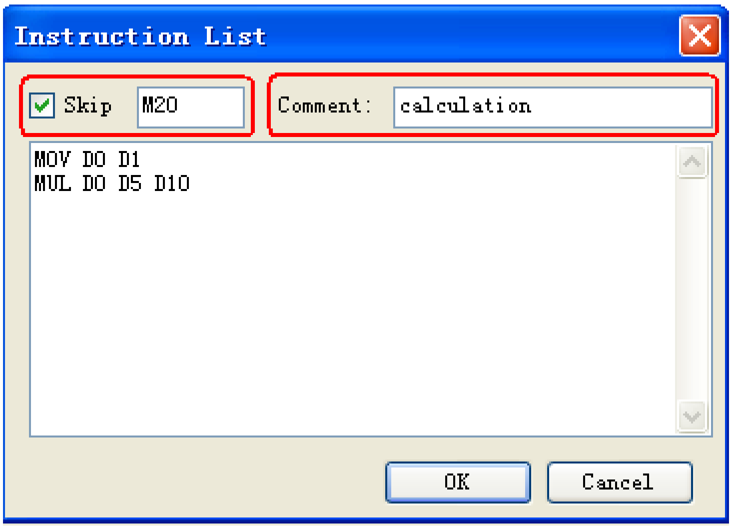
An ‘instruction list’ will jump out after click the ‘command item’:



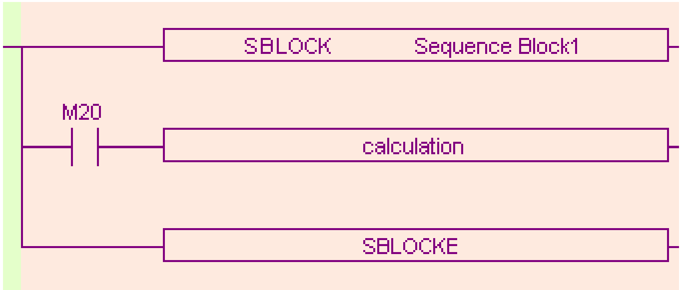
Users can add instructions in the frame.

Skip: to control the stop and run of the instructions. If you select skip and input control coil in the frame, then when the control coil is ON, the command will not be executed. If not select, the default action is execution.

Comment: to modify the note for the instruction.



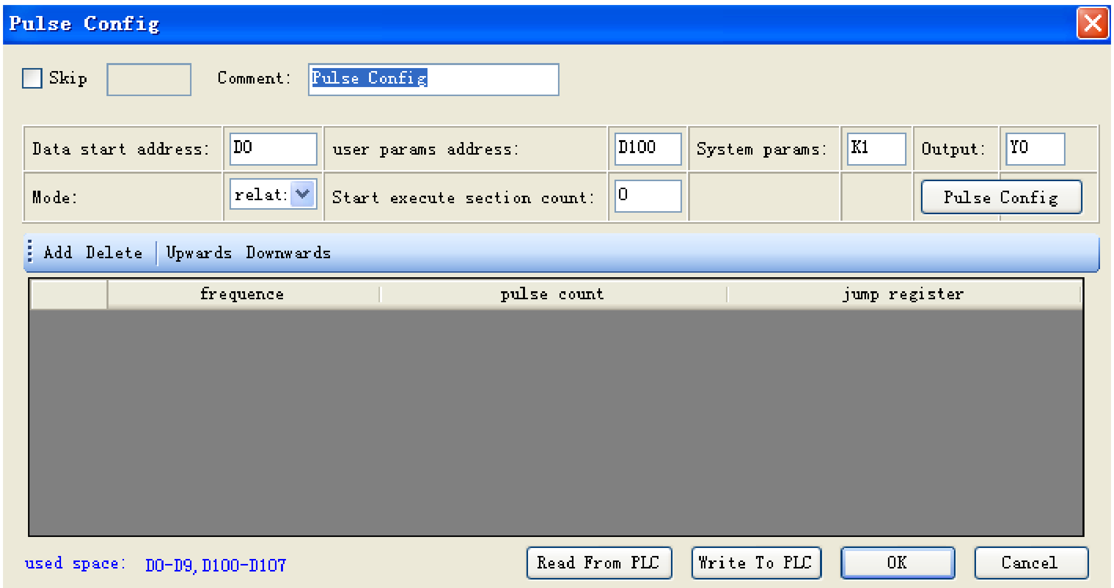
Click 'OK', the ladder program will change as the following:



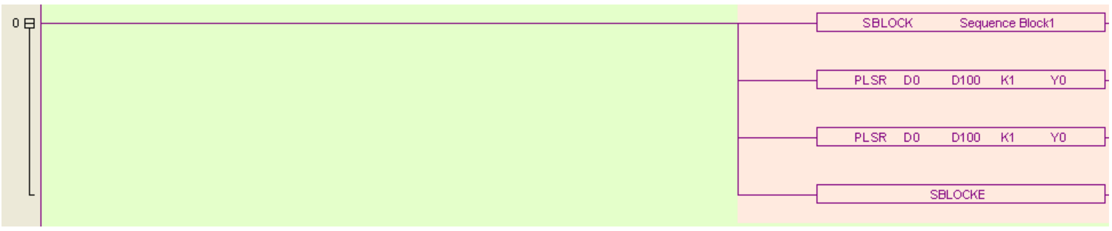
Note: We can add multiply instructions in one BLOCK and use 'Skip' as every instruction's execution condition.

9-3-2. Pulse Item

Open the 'pulse item' in the same way:



In the following BLOCK, we add two impulse instructions:

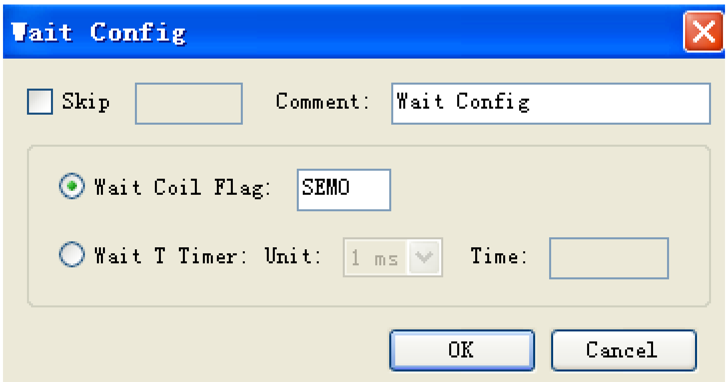


9-3-3. Wait Item

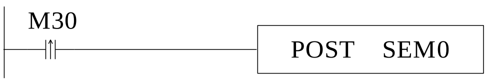
'Wait Item': to wait coil flag or timer bit.

Open 'Wait Item' in the same way. There are two waiting modes: flag bit and timer wait.

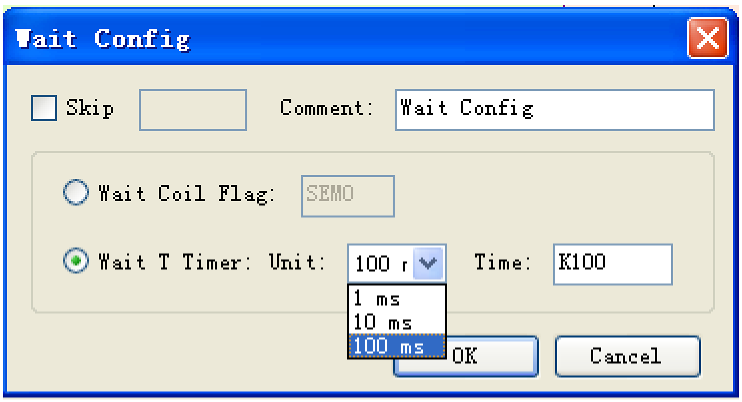
(A) Flag bit



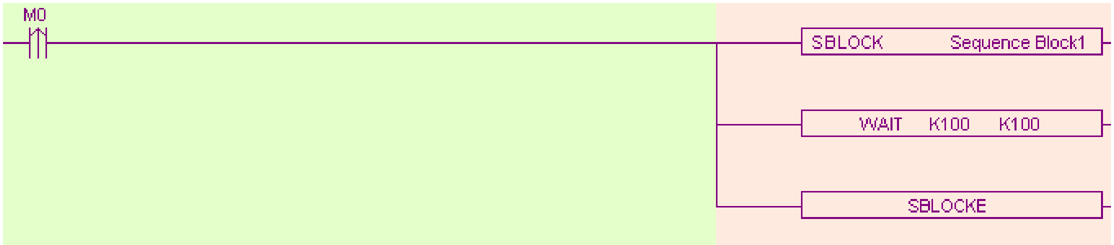
SEM corresponding ladder diagram is as below:



(B) Timer wait



(C) Corresponding ladder diagram:

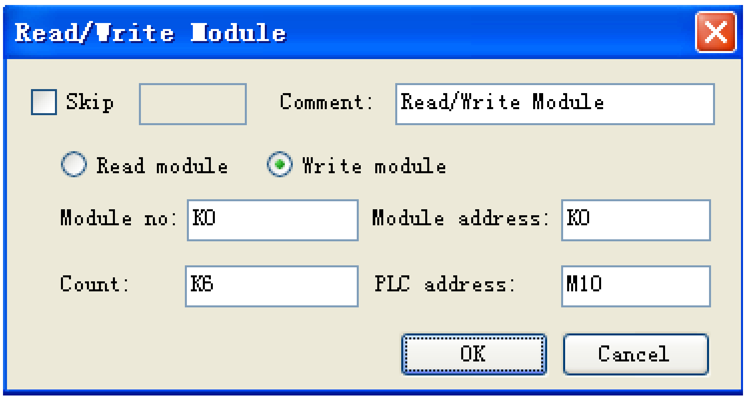


Note: Do not add normal coil after WAIT instruction in XD/XL series PLC sequence BLOCK, and add XD, XL series PLC special signal SEM bit(SEM0~SEM31); SEM cannot be controlled by set or reset. It can only be set by POST instruction and reset by WAIT SEM instruction. Or output via OUT instruction. The difference between them is that the POST command needs to be triggered by the pulse edge to keep the state of SEM; the OUT command needs to be triggered by the normally open coil, and the SEM is reset when the triggering condition is disconnected.

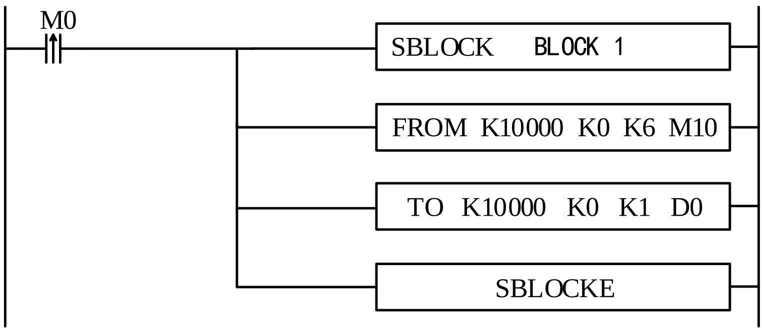
9-3-4. Module Read and Write （FROM/TO）instruction

This item is used to read and write data between PLC and modules, and the operate panel is as below:

1#read



FROM\TO instruction can be selected from pull-down list:

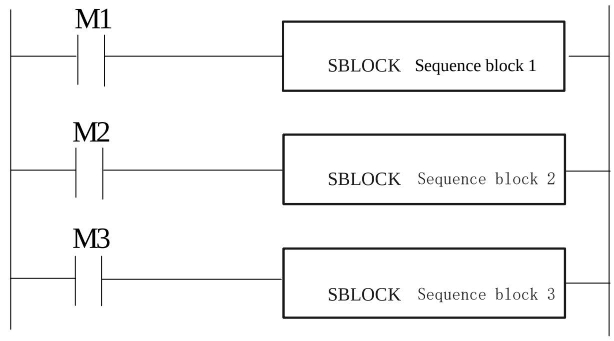


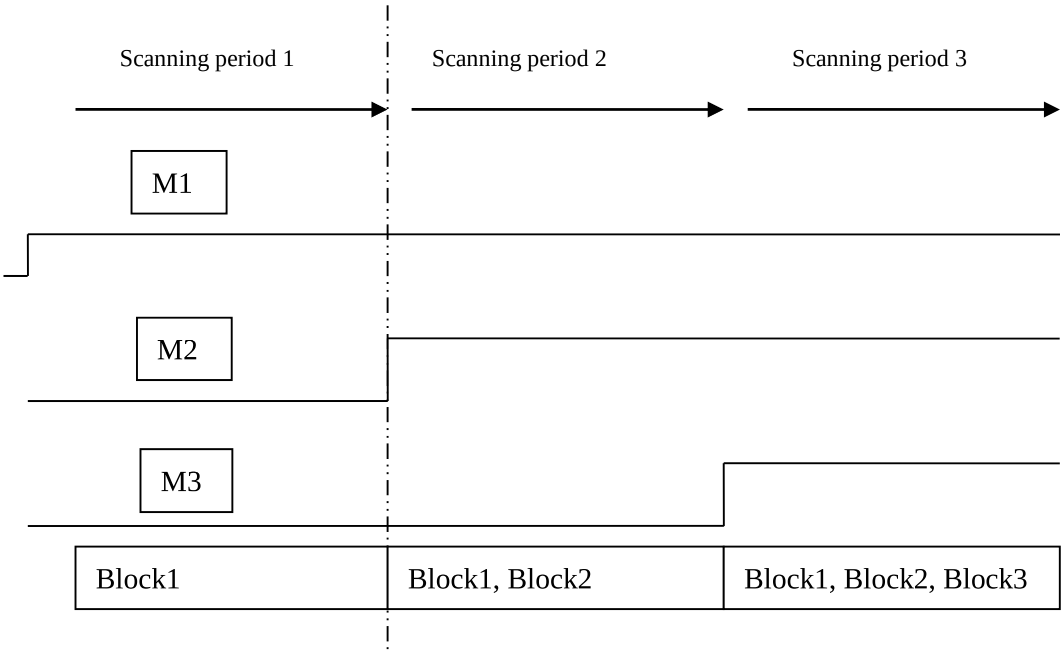
Note: As shown in the figure above, in V3.4 and above version software, when the module number is set to K0~K15, the corresponding ladder diagram will be displayed as K10000~K10015.

9-4. Running form of the BLOCK

1. If there are many blocks, they run as the normal program. The block is running when the condition is ON.

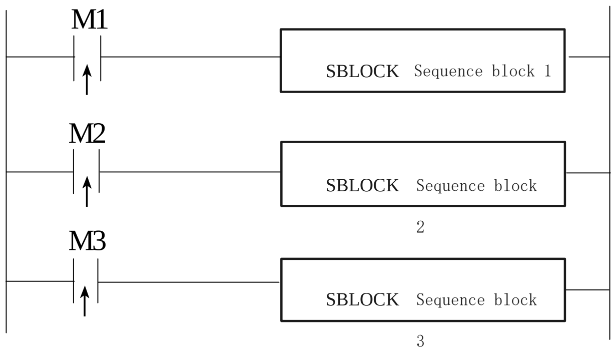
(A) The condition is normal ON, normal OFF coil





Note: When the program in the BLOCK is not executed and the triggering condition M is disconnected, the BLOCK will not stop immediately, but will complete the last scan, and will stop after the rest of the program has been executed.

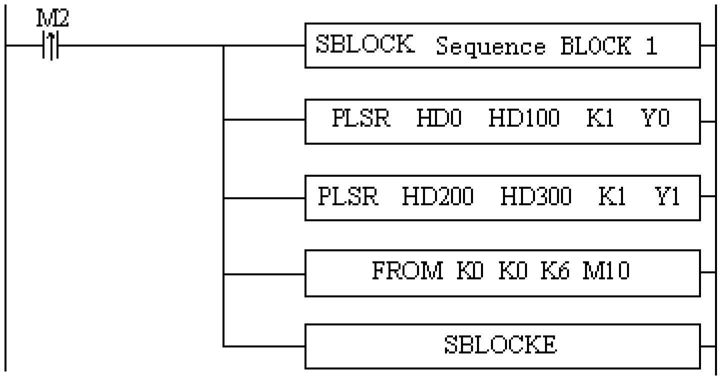
(B) The condition is rising or falling edge of pulse



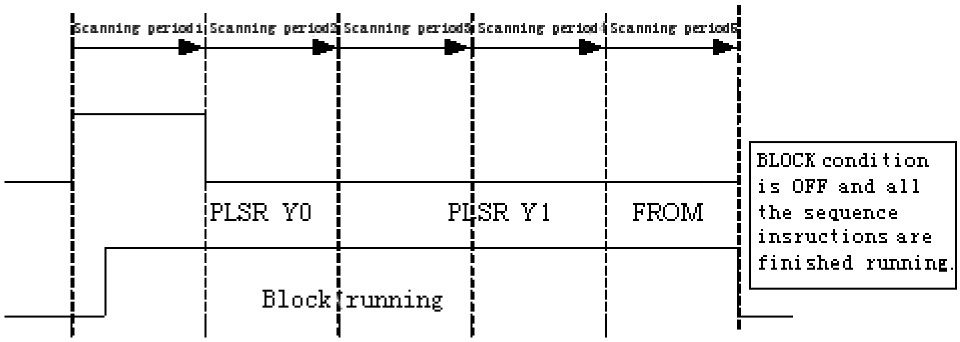
When M1, M2, M3 is from OFF to ON, all these blocks will run once.

2. The instructions in the block run in sequence according to the scanning time. They run one after another when the condition is ON.

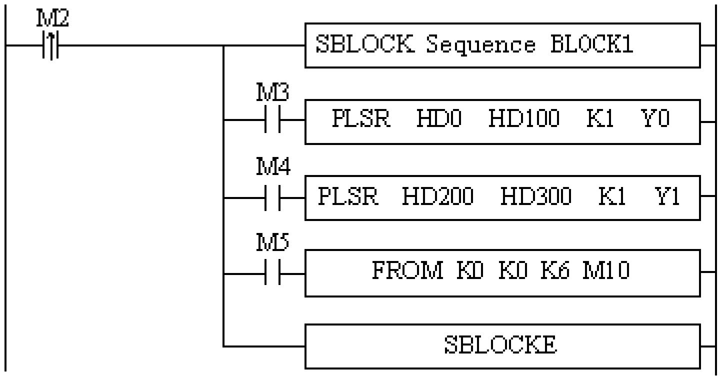
(A) Without SKIP condition



The instructions running sequence in block 1 is shown as below:



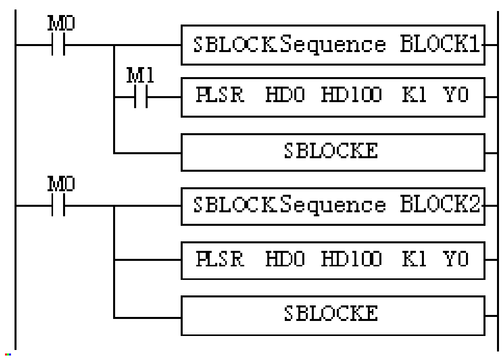
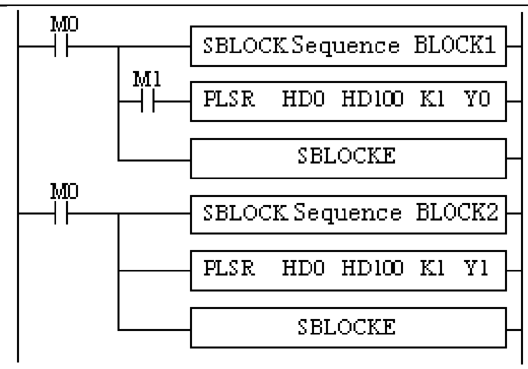
(B) With SKIP condition



Explanation:
A) When M2 is ON, block 1 is running.
B) All the instructions run in sequence in the block.
C) M3, M4, M5 are the sign of SKIP, when they are ON, this instruction will not run.
D) When M3 is OFF, if no other instructions use this Y0 pulse, PLSR HD0 HD100 K1 Y0 will run; if not, the PLSR HD0 HD100 K1 Y0 will run after it is released by other instructions.
E) After Y0 pulse sending completed, check M4. If M4 is OFF, check Y1 block, if M4 is ON, check M5. If M5 is OFF, module commmunication will run.

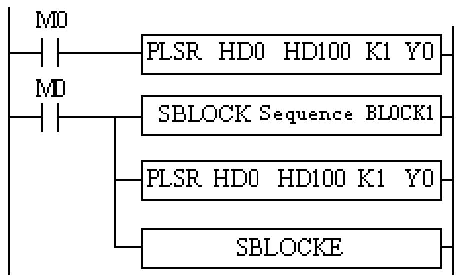
9-5. BLOCK instruction editing rules

In the BLOCK, the instruction editing should accord with some standards.
Do not use the same pulse output terminal in different BLOCK.

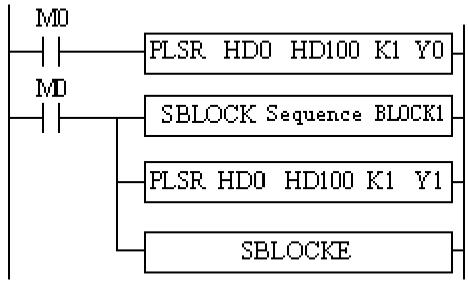
NO (×)	**YES** (√)
	

Do not use the same pulse output terminal in BLOCK and main program.

NO （×）

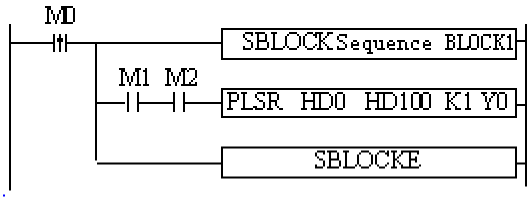


YES （√）

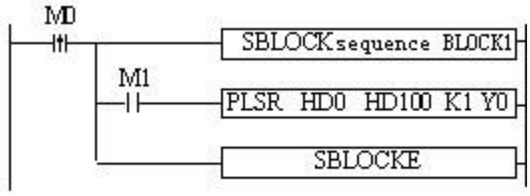


There only can be one SKIP condition for one BLOCK instruction.

NO （×）

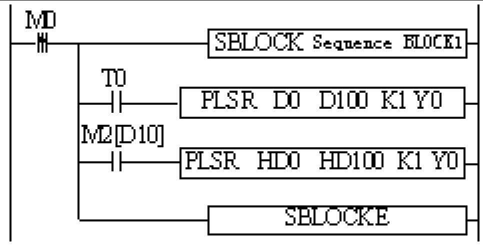


YES （√）

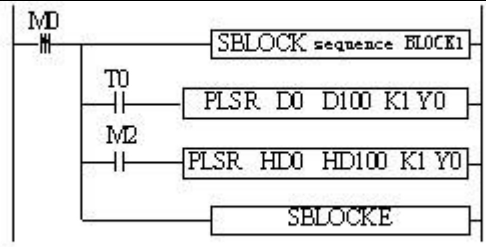


The SKIP condition only can use M, X, can not use other coil or register.

NO （×）

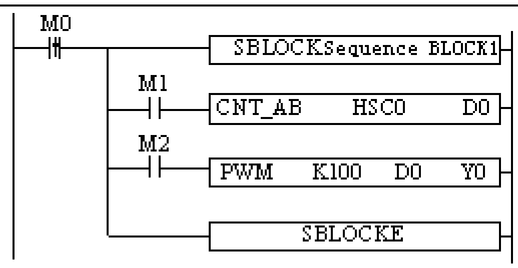


YES （√）

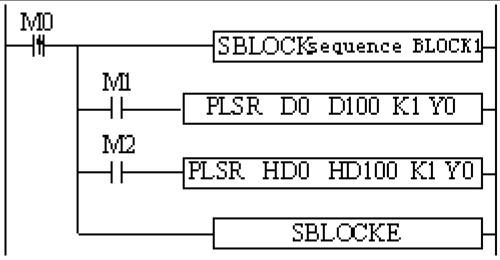


The output instructions cannot be CNT_AB(CNT), PWM.

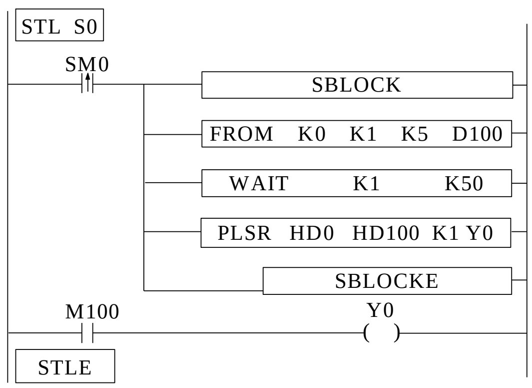
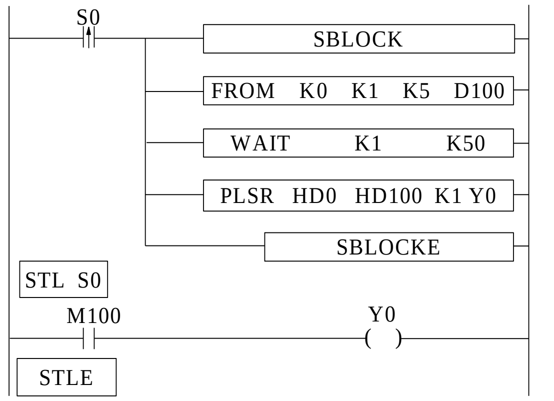
NO （×）



YES （√）



BLOCK is not recommended to put in the STL, because if one STL ends, while the BLOCK doesn't end, then big problem will happen.

NO（×）	YES（√）
	

Label Kind type cannot be used in the block

Sign P, I cannot be used in block. Even they can be added in block, but they do not work in fact.

9-6．BLOCK related instructions

9-6-1．Instruction explanation

stop running the BLOCK [SBSTOP]

Summarization

Stop the instructions running in the block

[SBSTOP]			
16 bits	SBSTOP	32 bits	-
Condition	NO,NC coil and pulse edge	Suitable types	XD, XL
Hardware		Software	V3.2

Operand

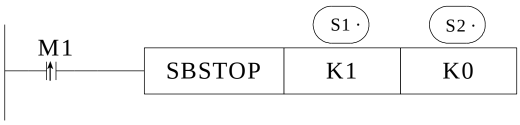
Operand	Function	Type
S1	The number of the BLOCK	16bits, BIN
S2	The mode to stop the BLOCK	16bits, BIN

Suitable component

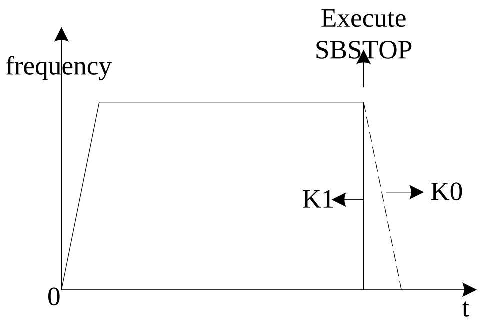
	Operand	Register								Constant	Module	
Word		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S1	●								●		
	S2									●		

***Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.**

Function



S2 is the mode for BLOCK stop, operand: K0, K1, K2
K0: stop the BLOCK slowly, if the pulse is outputting, the BLOCK will stop after the pulse outputting is finished.
K1: stop the BLOCK immediately; stop all the instructions running in the BLOCK.



K2: Destructive slow stop BLOCK, that is, when the pulse is being sent, the SBSTOP condition holds, then the pulse will slow down along the slope, without to use with the SBGOON instruction, so the remaining instructions will not be executed. After executing this instruction, the BLOCK can be restarted. (Note: K2 mode is only supported by V3.4.2 and above PLC)

Continue running the BLOCK[SBGOON]

Summarization
This instruction is opposite to SBSTOP. To continue running the BLOCK.

[SBGOON]			
16 bits	SBGOON	32 bits	-
Condition	Pulse edge	Suitable types	XD, XL
Hardware	-	Software	V3.2

Operand

Operand	Function	Type
S1	The number of the BLOCK	16 bits, BIN
S2	The mode to continue running the BLOCK	16 bits, BIN

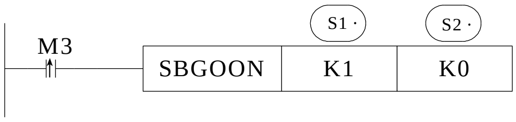
Suitable component

Word

Operand	Register								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●								●		
S2									●		

***Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM;   DS includes DS, DHS.**

Function

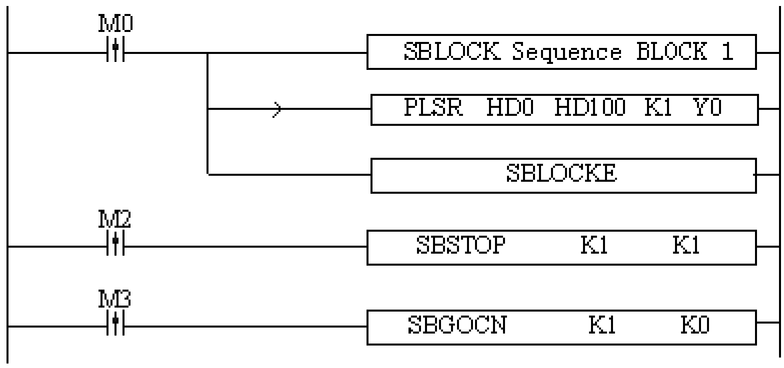


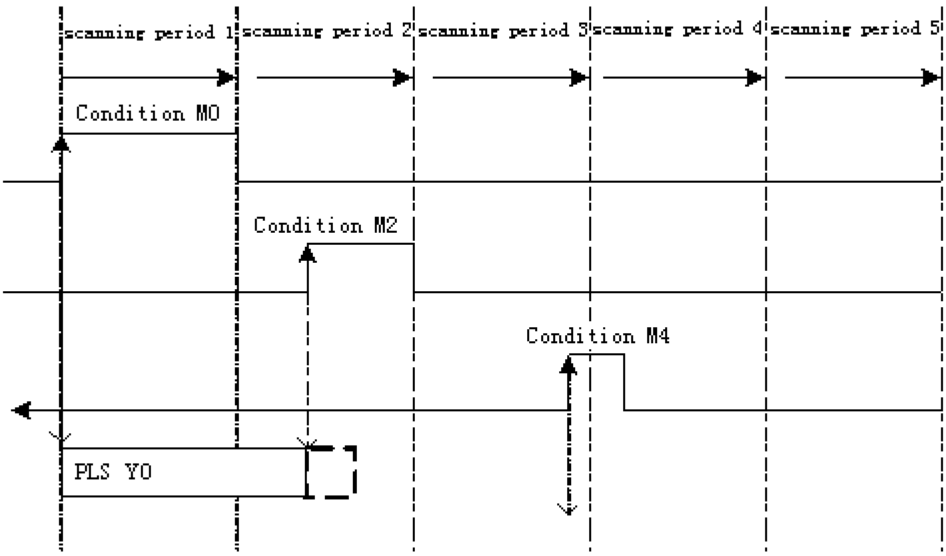
S2 is the mode to continue running the BLOCK. Operand: K0, K1.
K0: continue running the instructions in the BLOCK.
For example, if pulse outputting stopped last time, SBGOON will continue outputting the rest pulse;
K1: continue running the BLOCK, but abandon the instructions have not finished last time. Such as the pulse output instruction, if the pulse has not finished last time, SBGOON will not continue outputting this pulse but go to the next instruction in the BLOCK.

This instruction only applies to PLSR instructions in BLOCK, and can only send the remaining pulses for interpolation instructions, which can not be skipped.

9-6-2. The timing sequence of the instructions

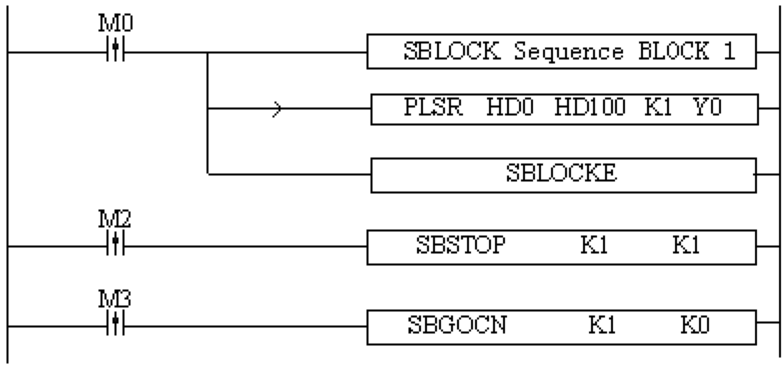
SBSTOP (K1 K1) + SBGOON (K1 K1)

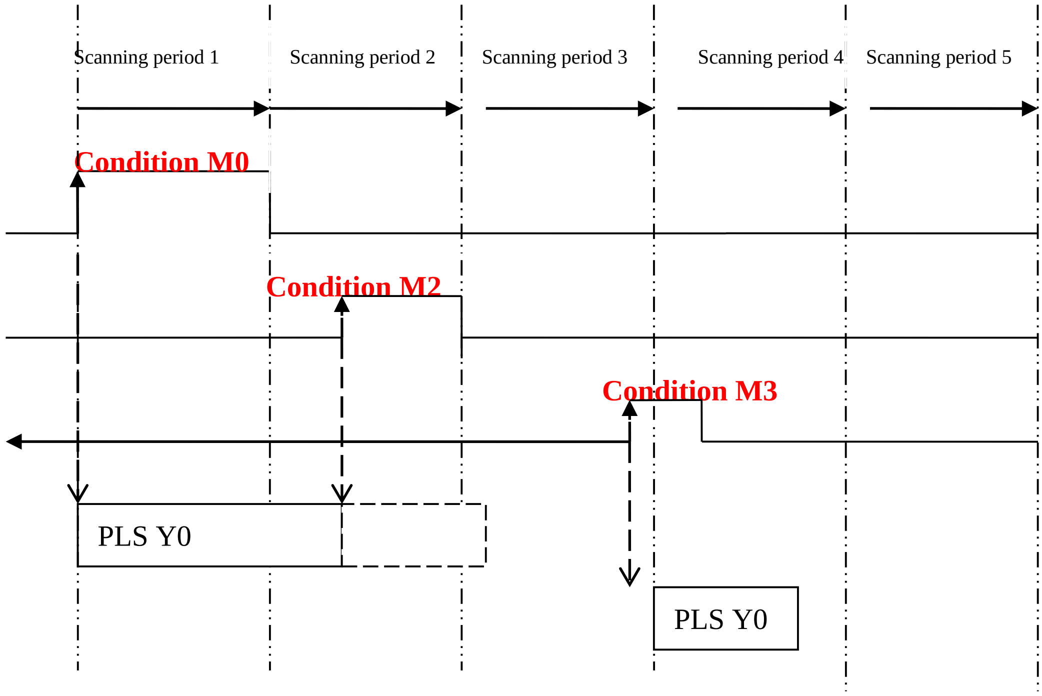




When M0 is from OFF→ON, run “PLSR HD0 HD100 K1 Y0” in the BLOCK to output the pulse;
When M2 is from OFF→ON, the BLOCK stops running at once;
When M4 is from OFF→ON, abandon the rest pulse.

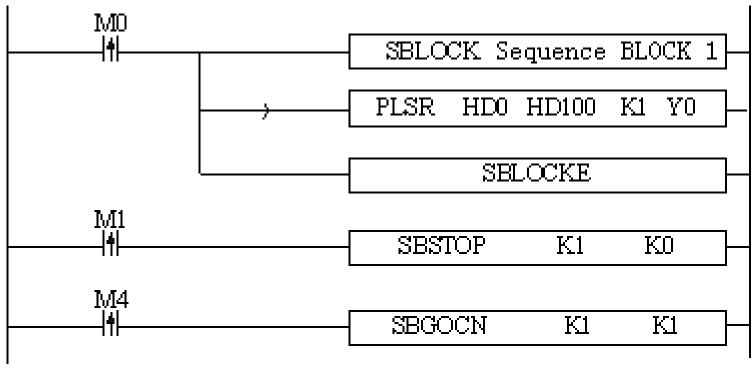
SBSTOP（K1 K1）+SBGOON（K1 K0）

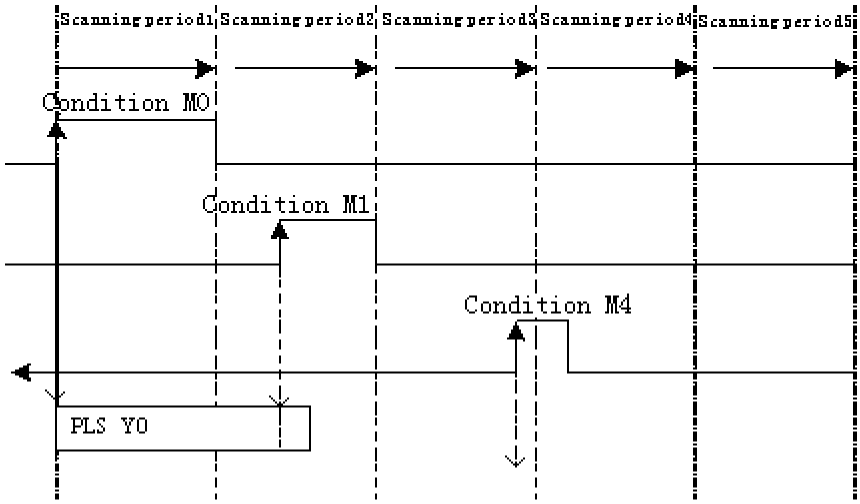




When M0 is OFF→ON, run ‘PLSR HD0 HD100 K1 Y0’ in the BLOCK to output the pulse;
When M2 is OFF→ON, the BLOCK stops running, the pulse output stops at once;
When M3 is OFF→ON, output the rest pulses.

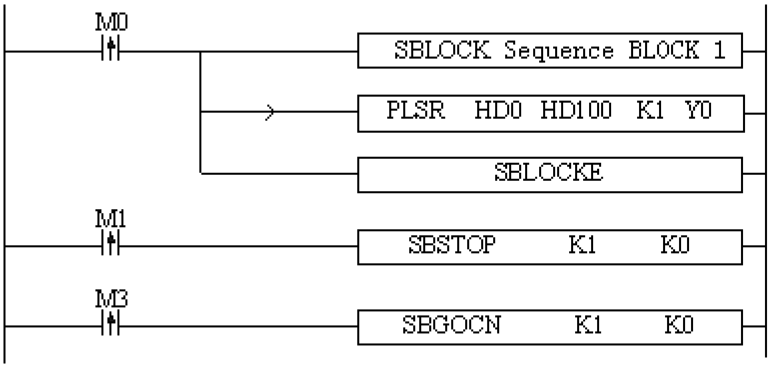
3. SBSTOP(K1 K0)+SBGOON(K1 K1)

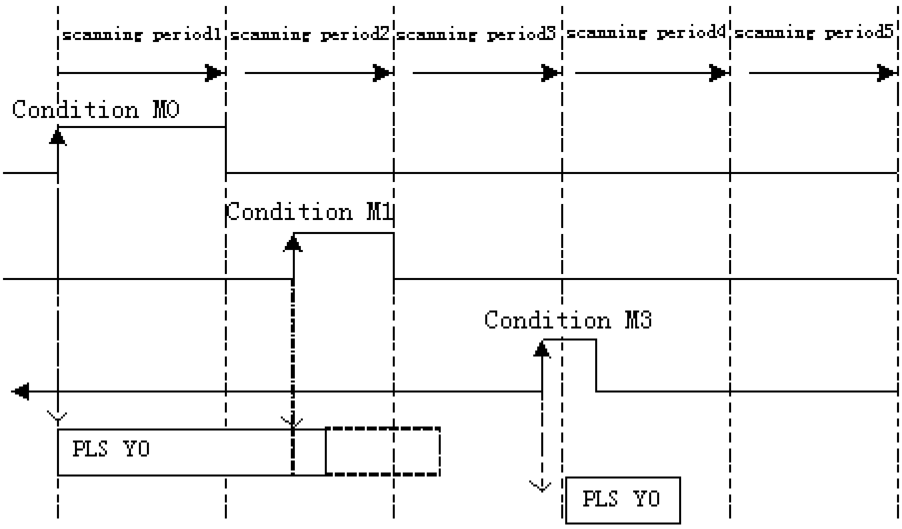




When M0 is from OFF→ON, run 'PLSR HD0 HD100 K1 Y0' in the BLOCK to output the pulse;
When M1 is from OFF→ON, stop running the BLOCK, the pulse will stop slowly with slope;
When M4 is from OFF→ON, abandon the rest pulses.

4. SBSTOP(K1 K0)+SBGOON(K1 K0)





When M0 is from OFF→ON, run ‘PLSR HD0 HD100 K1 Y0’ in the BLOCK to output the pulse;
When M1 is from OFF→ON, suspend running the BLOCK, the pulse will stop slowly with slope;
When M3 is from OFF→ON, output the rest pulses.

Please note that by the SBSTOP stops the pulse with slope, there may be still some pulses; in this case, if run SBGOON K1 K0 again, it will output the rest of the pulses.

9-7. BLOCK flag bit and register

1. BLOCK flag bit:

Address	Function	Explanation
SM300	BLOCK1 running flag	1: running 0: not running
SM301	BLOCK2 running flag	
SM302	BLOCK3 running flag	
…….	…….	
……..	…….	
SM399	BLOCK100 running flag	

2. BLOCK flag register:

Address	Function	Explanation
SD300	BLOCK1 running instruction	BLOCK use this value when monitoring
SD301	BLOCK2 running instruction	
SD302	BLOCK3 running instruction	
…….	…….	
……..	…….	
SD399	BLOCK100 running instruction	

If GBLOCK is used, it will occupy SM399 and SD399.

10 Special Function Instructions

This chapter mainly introduces PWM (pulse width modulation), precise timing, interruption etc.

Special Function Instructions List:

Mnemonic	**Function**	**Circuit and soft components**	**Chapter**
Pulse Width Modulation, Frequency Detection			
PWM	Output pulse with the specified duty cycle and frequency	PWM S1 S2 D	10-1
FRQM	Fixed pulses frequency measurement	FRQM S1 D S2 S3	10-2
Time			
STR	Precise Time	STR D1 D2	10-3
Interruption			
EI	Enable Interruption	EI	10-4-1
DI	Disable Interruption	DI	10-4-1
IRET	Interruption Return	IRET	10-4-1

10-1. Pulse Width Modulation [PWM]

1. Instruction's Summary

Instruction to realize PWM pulse width modulation

PWM pulse width modulation [PWM]			
16 bits instruction	PWM	32 bits instruction	-
execution condition	normally ON/OFF coil	suitable models	XD/XL (except XD1/XL1)
hardware requirement	-	software requirement	-

2. Operands

Operands	Function	Type
S1	specify the duty cycle value or soft component's ID number	32 bits, BIN
S2	specify the output frequency or soft component's ID number	32 bits BIN
D	specify the pulse output port	bit

3. Suitable Soft Components

Word

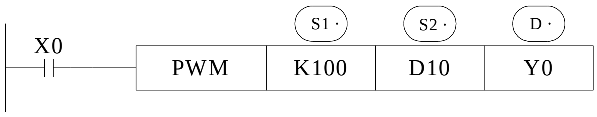
Operands	System									Constant	Module	
	D*	FD	ED	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●					●		
S2	●	●		●	●					●		

Bit

Operands	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●					

*Note: D includes D, HD; TD includes TD HTD; CD includes CD HCD HSCD HSD; DM includes DM DHM; DS includes DS DHS. M includes M HM SM; S includes S HS ; T includes T HT ; C includes C HC

Function and



Duty cycle **n**: 1~65535

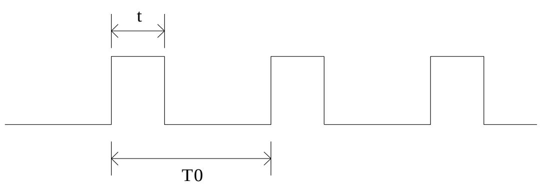
Output pulse **f**: 1~100KHz

XD series PLC PWM output need transistor type terminal:

PLC model	PWM terminal
XD2-16T/RT -24T/RT -32T/RT -48T/RT -60T/RT	Y0、Y1
XD3-16T/RT -24T/RT -32T/RT -48T/RT -60T/RT	Y0、Y1
XD5-16T -24T/RT -32T/RT -48T/RT -60T/RT	Y0、Y1
XD5-24T4 -32T4 -48T6 -60T6	Y0、Y1、Y2、Y3
XDM-24T4 -32T4 -60T4 -60T10	Y0、Y1、Y2、Y3
XDC-24T -32T -48T -60T	Y0、Y1
XD5E-30T4 -60T10	Y0、Y1、Y2、Y3
XDME-60T10	Y0、Y1、Y2、Y3
XL3-16T	Y0、Y1
XL5-32T4、XL5E-32T4、XLME-32T4	Y0、Y1、Y2、Y3

Duty cycle of **PWM** output =n /65535×100%

PWM use the unit of 0.1Hz, so when set S2 frequency, the set value is 10 times of the actual frequency (10f). E.g.: to set the frequency as 72 KHz, and then set value in S2 is 720000. When X0 is ON, output PWM wave; When X0 is OFF, stop output. PMW output doesn’t have pulse accumulation.



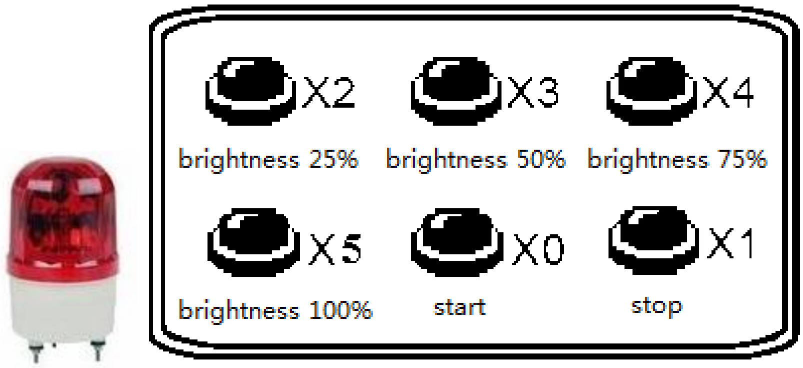
In the left graph:

T0=1/f

t/T0=n/65535

Note: it needs to connect 1K ohm amplification resistor between output terminal and common terminal when using PWM instruction.

Example



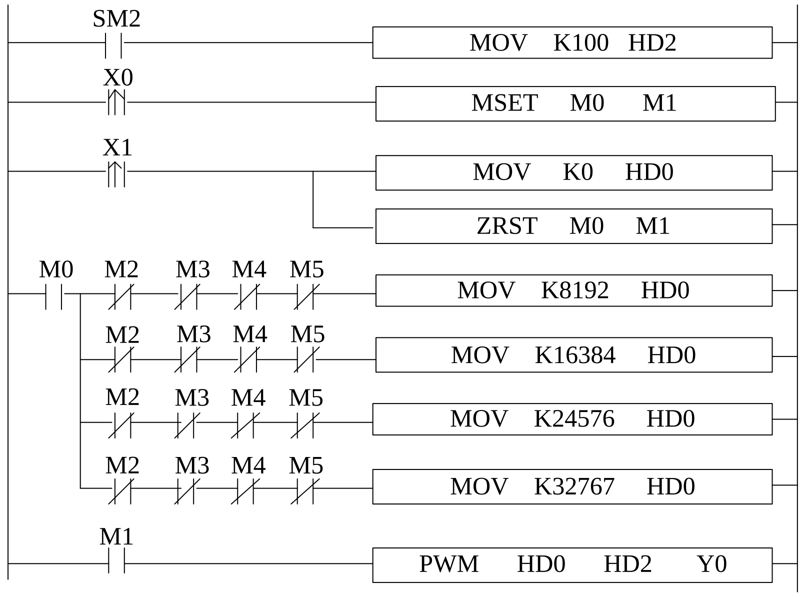
There is a LED drived by DC24V. It needs to control the brightness of the LED. In order to decrease the power loss of wave collector, turn ON the switch at the moment it is OFF, then turn it OFF. This process will cycle. Connet a transistor between the power supply and LED. The pulse signal will input from the transistor base terminal. The current between base and emitter is pulse. The LED input voltage is proportional to the duty ratio. The LED input voltage will be changed by changing the duty ratio. There are many methods to change the value. The normal way is pulse width modulation (PWM) which means only changing the ON holding time but not changing the ON frequency.

This example applies the PWM technology to the LED brightness adjustment. The controller can accpet 24V PWM control signal. The brightness range includes 25%, 50%, 75%, 100%. The brightness is controlled by the PWM duty ratio.

Element explanation:

PLC component	Explanation	Mark
X0	Start button, X0 is ON when pressed.	
X1	Stop button, X1 is ON when pressed.	
X2	25% brightness button, X2 is ON when pressed.	
X3	50% brightness button, X3 is ON when pressed.	
X4	75% brightness button, X4 is ON when pressed.	
X5	100% brightness button, X5 is ON when pressed.	
HD0	PWM duty ratio register	
HD2	PWM frequency register	Defaulted 100Hz

Program:



Program explanation:

1. HD0 will control the LED voltage. The voltage = 24*HD0/32767, pulse output frequency is 100Hz.
2. Press start button, X0 is ON, M0, M1 is ON, the LED brightness adjustment starts.
3. X2 is ON, HD0=8192, HD0/32768=0.25, the LED brightness is 25%.
4. X3 is ON, HD0=16384, HD0/32768=0.5, the LED brightness is 50%.
5. X4 is ON, HD0=24576, HD0/32768=0.75, the LED brightness is 75%.
6. X5 is ON, HD0=32768, HD0/32768=1, the LED brightness is 100%.
7. Press shut down button, X1 is ON, HD0 is reset, shut down the PWM trigger condition, LED voltage is 0V.

10-2. Frequency measurement [FRQM]

1. Instruction list

Measure the frequency.

Frequency measurement [FRQM]			
16 bits instruction	-	32 bits instruction	FRQM
execution condition	Normally ON OFF coil	suitable models	XD/XL (except XD1/XL1)
hardware requirement	-	software requirements	-

2. Operand

Operands	Function	Type
S1	Sampling pulse numbers	32 bits, BIN
S2	Frequency division option	32 bits, BIN
D	Measurement result	32 bits, BIN
S3	Pulse input terminal	bit

3. Suitable component

Word

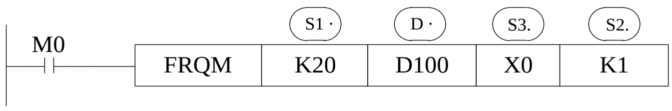
Operand	System									Constant	Module	
	D*	FD	ED	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●					●		
S2	●	●		●	●					●		

Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D		●					

*Note: D includes D HD; TD includes TD HTD; CD includes CD HCD HSCD HSD; DM includes DM DHM; DS includes DS DHS.
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and Action



- The sampling pulse numbers can be adjusted according to the frequency, the higher the frequency, the bigger the sampling pulse numbers
- Measurement result, the unit is Hz
- Display resolution: only can set to 1, 10, 100, 1000, 10000
- When M0 is ON, FRQM collects 20 pulses from X0, and records the sampling time. The result of sampling numbers dividing by sampling time will be saved in D100. The measurement process will repeat. If the measurement freqeuncy is less than the measurement range, the result is 0
- The measurement precision is 0.001%

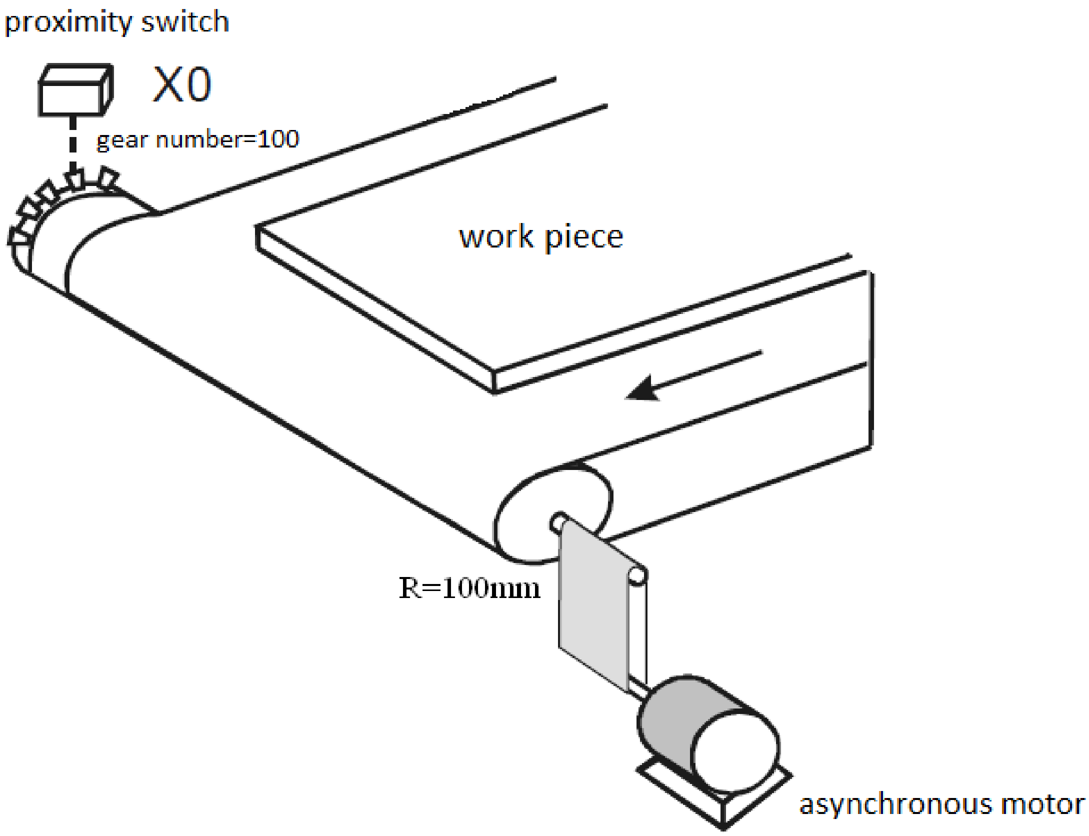
The pulse input terminal for FRQM:

Model		X terminal	Max frequency (Hz)
XD2	16 I/O	X0	10K
		X3	
		X6	
	24/32 I/O	X0	80K
		X3	10K
		X6	
	48/60 I/O	X0	80K
		X3	
		X6	10K
XD3	16/24/32 I/O	X0	80K
		X3	10K
		X6	
	48/60 I/O	X0	80K
		X3	
		X6	10K

XD5	16/24/32 I/O	X0	80K
		X3	10K
		X6	
	24T4/32T4/48T4/60T4 I/O	X0	80K
		X3	
		X6	
		X11	
	48/60 I/O	X0	80K
		X3	
		X6	10K
	48T6/60T6/60T10 I/O	X0	80K
		X3	
		X6	
		X11	
XDM	24T4/32T4/60T4 I/O	X0	80K
		X3	
		X6	
		X11	
	60T10 I/O	X0	80K
		X3	
		X6	
		X11	
XDC	24/32/48/60 I/O	X0	80K
		X3	
		X6	
		X11	
XD5E	30T4/60T10 I/O	X0	80K
		X3	
		X6	
		X11	
XL3	16 I/O	X0	80K
		X3	10K
		X6	
XL5	32T4 I/O	X0	80K
		X3	
		X6	
		X11	
XL5E	32T4 I/O	X0	80K
		X3	
		X6	
		X11	
XLME	32T4 I/O	X0	80K
		X3	
		X6	
		X11	

Example

Asynchronous motor drives the conveyor to transfer the work piece. It needs to real-time display the work piece moving speed. The diameter of the transmission shaft is 100mm, the gear numbers on the transmission shaft are 100, the speed unit is m/min.



Component explanation:

PLC component	Control explanation	Mark
X0	Proximity switch, to count the gear numbers	
M0	Start signal	
D16	Speed register (float number)	

Program:

```
M0
-| |-----+--[FRQM   K20   D0    X0   K1]
         +--[DFLT   D0    D2]
         +--[EDIV   D2    K100  D4]
         +--[EMUL   K0.1  K3.14 D6]
         +--[EMUL   D4    D6    D10]
         +--[EMUL   D10   K60   D16]
```

Program explanation:

1. Set ON the start signal M0, to run the frequency measurement program
2. Transform the frequency to float number, then it is divided by 100 (gear numbers per rotation), the result is shaft rotate numbers per second (float number).

3. Calculate the diameter of the transmission shaft and save in register D6 (float number), then calculate the transfer distance per second and save in D10 (float number).
4. the transfer distance per second multiply by 60 is the speed (m/min).

10-3. Precise Timing [STR]

1. Instruction List

Read and stop precise timing when precise timing is executed

Precise timing[STR]			
16 bits instruction	-	32 bits instruction	STR
execution condition	edge activation	suitable models	XD/XL
hardware requirement	-	software requirements	-

2. Operands

Operands	Function	Type
D1	Timer Number	bit
D2	specify timer’s value or soft component’s ID number	32 bits, BIN

3. Suitable Soft Components

Word

Operands	system									constant	module	
	D*	FD	ED	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
D2	●	●		●	●					●		

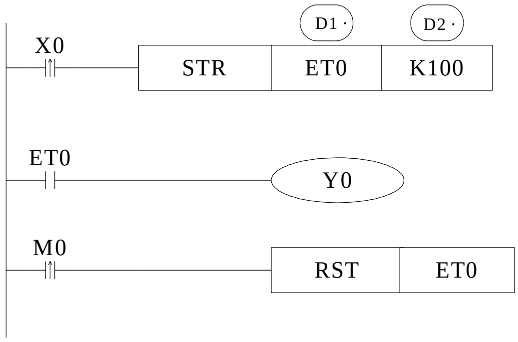
Bit

Operands	system						
	X	Y	M*	S*	T*	C*	Dn.m
D					●		
D1					●		

*Note: D includes D HD; TD includes TD HTD; CD includes CD HCD HSCD HSD; DM includes DM DHM;
DS includes DS DHS.
M includes M HM SM； S includes S HS ； T includes T HT ； C includes C HC.

Function and Action

<Precise timing>, <Precise timing reset>



(D1·) Timer's number. Range: ET0~ET30 (ET0, ET2, ET4……all number should be even)

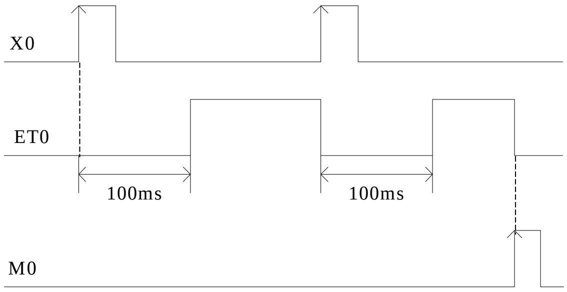
(D2·) Timing value

Precise timer works in unit of 1ms.

Precise timer 32 bits, the counting range is 0~+2,147,483,647.

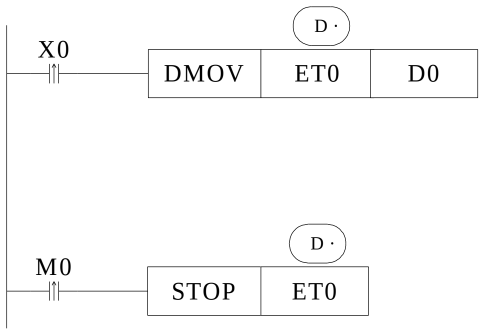
When executing STR, the timer will be reset before start timing.

When X0 turns from OFF to ON, ET0 starts timing. ET0 will be reset and keep its value 100 when accumulation time reaches 100ms; If X0 again turns from OFF to ON, timer T600 turns from ON to OFF， restart to time, when time accumulation reaches 100ms, T600 reset again. See graph below:



When the pre-condition of STR is normally open/closed coil, the precise timer will set ON immediately when the timing time arrives and reset the timing, and cycle back and forth.

<read the precise timing>, <stop precise time>



- When X0 changes from OFF to ON, move the current precise timing value into D0 immediately, it will not be affected by the scan cycle;

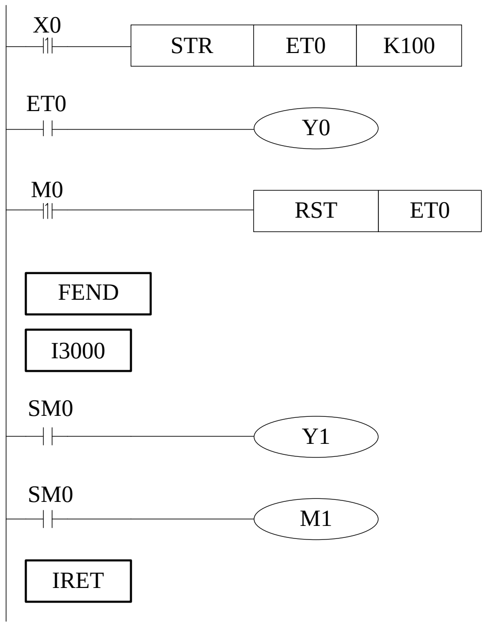
- When M0 changes from OFF to ON, execute STOP instruction immediately, stop precise timing and refresh the count value in ETD0. It will not be affected by the scan cycle;

Precise Timing Interruption

- When the precise timing reaches the count value, it will generate an interruption tag, interruption subprogram will be executed.
- Can start the precise timing in precise timing interruption;
- Every precise timer has its own interruption tag, as shown below:

Interruption Tag corresponding to the Timer:

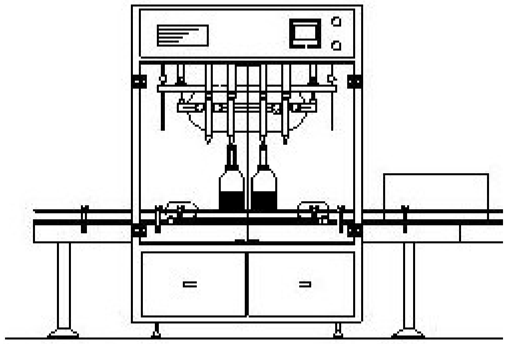
Timer's No	Interruption Tag	Timer's No	Interruption Tag
ET0	I3000	ET10	I3005
ET2	I3001	ET12	I3006
ET4	I3002	……	……
ET6	I3003	ET22	I3011
ET8	I3004	ET24	I3012



When X0 changes from OFF to ON, ET0 will start timing. And ET0 reset when accumulation time is up to 100ms; meantime generates an interruption, the program jumps to interruption tag I3000 and execute the subprogram.

Example 1

The filling machine controls the filling capacity by controlling the liquid valve open time (it is 3000ms in this application). To improve the filling capacity precision, the liquid valve open time can be controlled by precise timing.

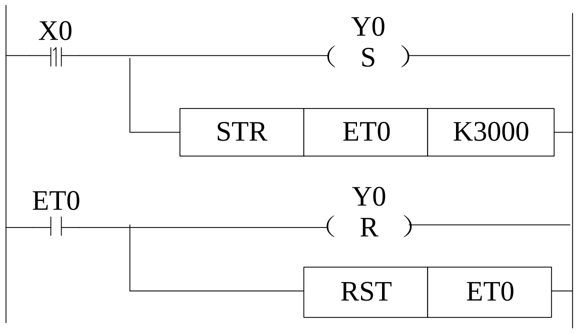


Filling machine

Component explanation:

PLC component	Control explanation	Mark
X0	Start button, X0 is ON when the button is pressed	
ET0	Precise timer	
Y0	Control the liquid valve, Y0 ON when the valve opened, Y0 OFF when the valve closed	

Program:

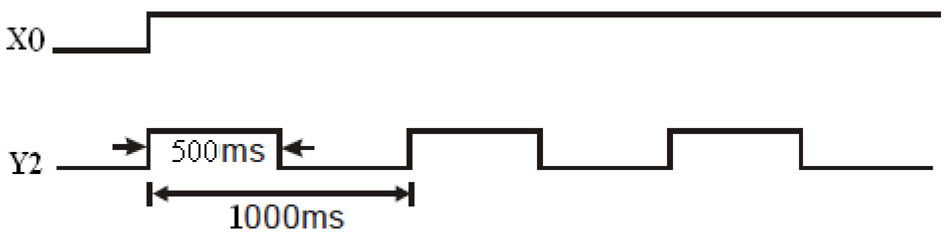


Program explanation:

1. When X0 is ON, the liquid valve Y0 and precise timer ET0 open at once.
2. Shut down the liquid valve Y0 and precise timer ET0 when the time arrived.

Example 2

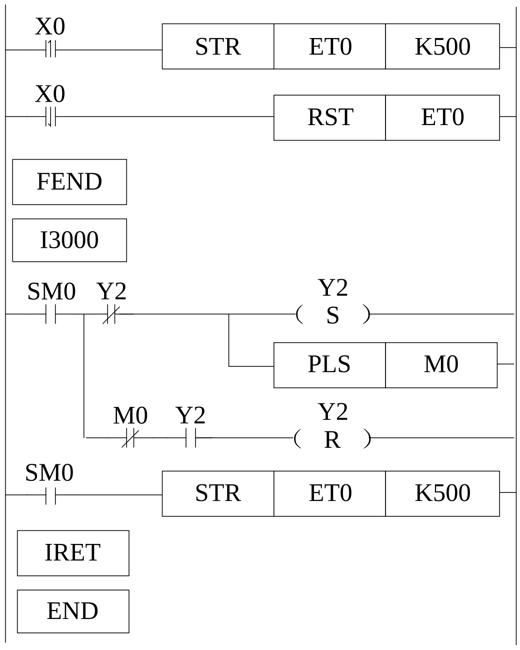
The precise timer interruption can produce the following pulse wave. The Y2 ON time is 500ms, the pulse period is 1000ms.



Component explanation:

PLC component	Control explanation	Mark
X0	Start button, X0 is ON when button is pressed	
Y2	Pulse output terminal	
M0	Internal auxiliary coil	
ET0	Precise timer	

Program:

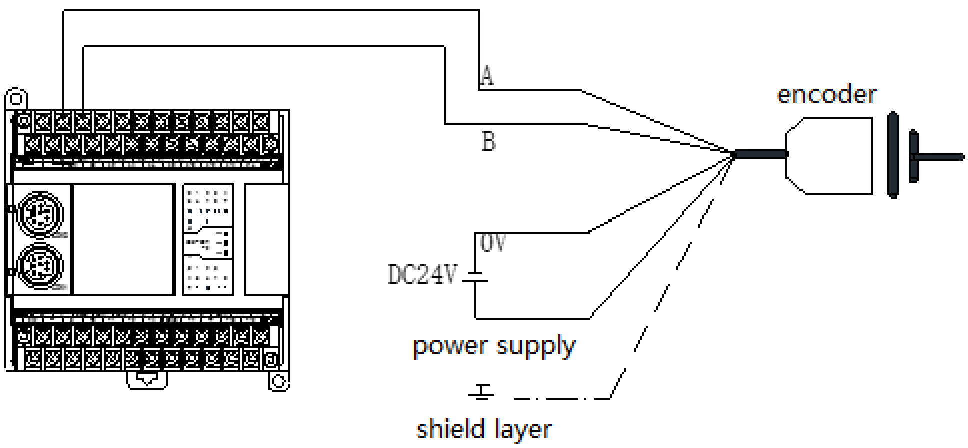


Program explanation:

1. When X0 is ON, the precise timer interruption will work, Y2 will output the pusle wave.
2. When X0 is OFF, shut down the precise timer interruption, Y2 stop outputting.

Example 3

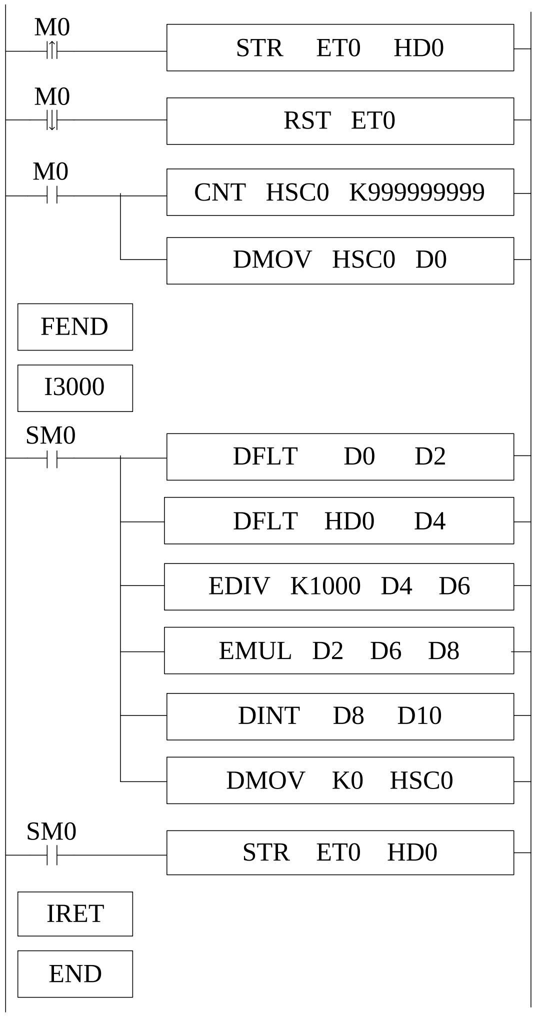
As the FRQM calculating the time for fixed pulse numbers, we will change the way to calculate the pulse numbers in fixed time.



Component explanation:

PLC component	Control explanation	Mark
M0	Start button, X0 is ON when pressed	
ET0	Precise timer	
HD0	Precise timer setting value (unit: ms)	
HSC0	High speed counter	
D10	The measured frequency (unit: s)	

Program:



Program explanation:

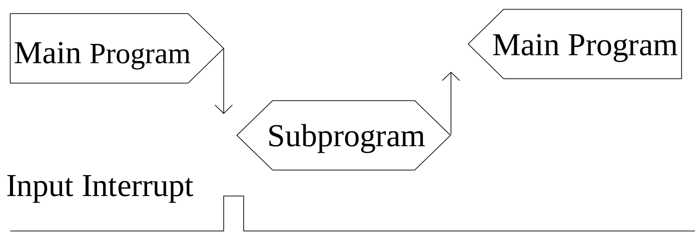
1. Set the high speed counter sampling period register HD0, the unit is ms.
2. Set ON M0 to start the precise timer interruption and high speed counter, calcuate the frequency
3. The frequency range is 0-80KHz, the precision is 0.005%.

10-4. Interruption [EI], [DI], [IRET]

XD/XL series PLC have interruption function, including external interruption and timing interruption. By interruption function we can deal with some special programs. This function is not affected by the scan cycle.

10-4-1. External Interruption

The input terminals X can be used to input external interruption. Each input terminal corresponds with one external interruption. The input's rising/falling edge can activate the interruption. The interruption subroutine is written behind the main program (behind FEND). After interruption generates, the main program stops running immediately, turn to run the correspond subroutine. After subroutine running ends, continue to execute the main program.



Note: The external interruption of XC series PLC cannot be activated by rising edge and falling edge at the same time; but XD/XL series PLC supports rising edge and falling edge activation meantime.

External Interruption's Port Definition

XD1/XD2/XD3/XD5/XL1/XL3 series 16 I/O

Input terminal	Pointer No.		Disable the interruption instruction
	Rising Interruption	Falling interruption	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052
X5	I0300	I0301	SM053
X6	I0400	I0401	SM054
X7	I0500	I0501	SM055

XD1 series 32 I/O, XD2/XD3 series 24/32/48/60 I/O, XD5 series, XDM series, XDC series, XD5E series, XDME series, XL5 series, XL5E, XLME series

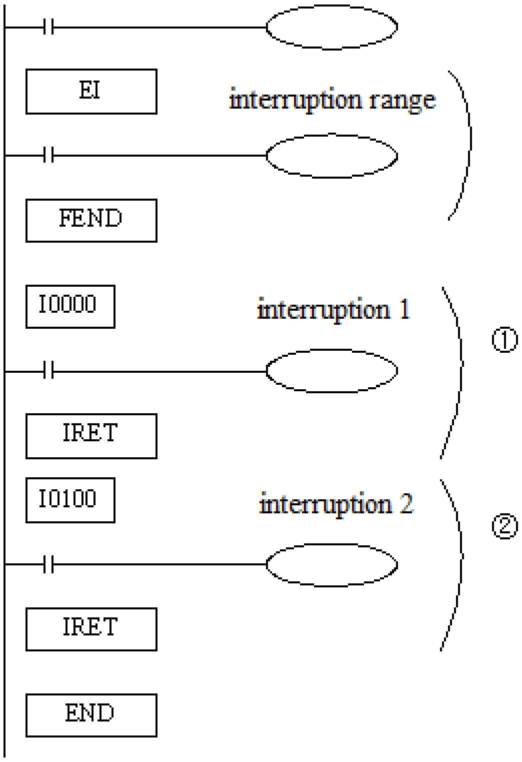
Input terminal	Pointer No.		Disable the interruption instruction
	Rising Interruption	Falling interruption	
X2	I0000	I0001	SM050
X3	I0100	I0101	SM051
X4	I0200	I0201	SM052

X5	I0300	I0301	SM053
X6	I0400	I0401	SM054
X7	I0500	I0501	SM055
X10	I0600	I0601	SM056
X11	I0700	I0701	SM057
X12	I0800	I0801	SM058
X13	I0900	I0901	SM059

Note: when the interruption ban coil is ON, the external interruption will not execute.

Interruption Instruction

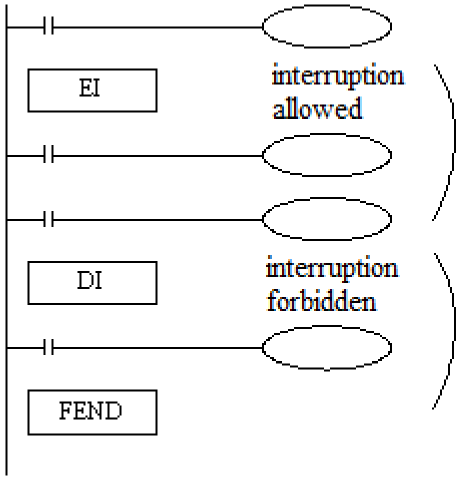
Enable Interruption [EI], Disable Interruption [DI], Interruption Return [IRET]



- If use EI instruction to allow interruption, then when scanning the program, if interruption input changes from OFF to ON, then execute subroutine ①、②. Return to the original main program.
- Interruption pointer (I****) should be behind FEND instruction;
- PLC is usually on the status that allows interruption.

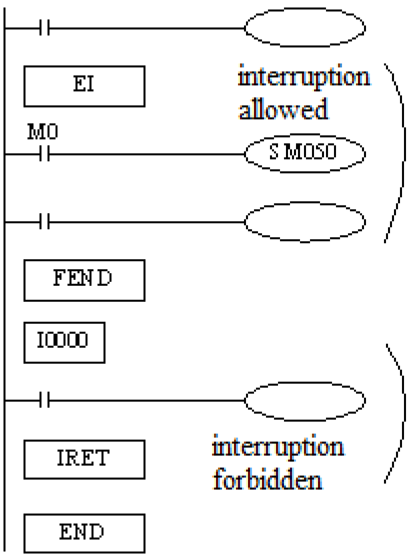
Note: In interrupt subroutine, only simple instructions such as set, reset, transmission and operation can be written, which can be executed in a scanning cycle. Other instructions such as sending pulses, timing (except for precise timing), communication and other instructions that need to be continuously executed are not supported.

Interruption's Range Limitation



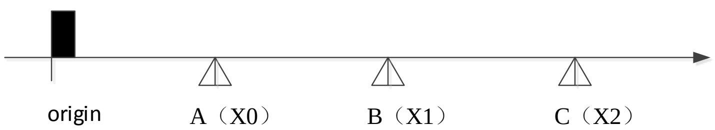
- By programming DI instruction, can set interruption disabled area;
- Allow interruption input between EI~DI
- If interruption forbidden is not required, please program only with EI, and program with DI is not required.

Disable the Interruption



- Every input interruption is equipped with special relays (SM50~SM69) to disable interruption.
- In the left program, if use M0 to set SM50 "ON", then disable the interruption 0.

Example 1



The positions of A, B, C are unknown. The speed of the three segments are different. The application can be perform by PLSF instruction and external interruption. We can install three proximity switch at postion A, B, C, and connect the signal to PLC input terminal X0, X1, X2. (suppose X0, X1, X2 are external interruption terminal, the related rising edge interruption ID are I0000, I0100, I0200. The PLC external interruption terminal please refer to "external interruption terminal definition). The pulse terminal is Y0, the direction terminal is Y2. To improve the speed changing precision, the acceleration and deceleartion time are 0. The speed will switch by external interruption.

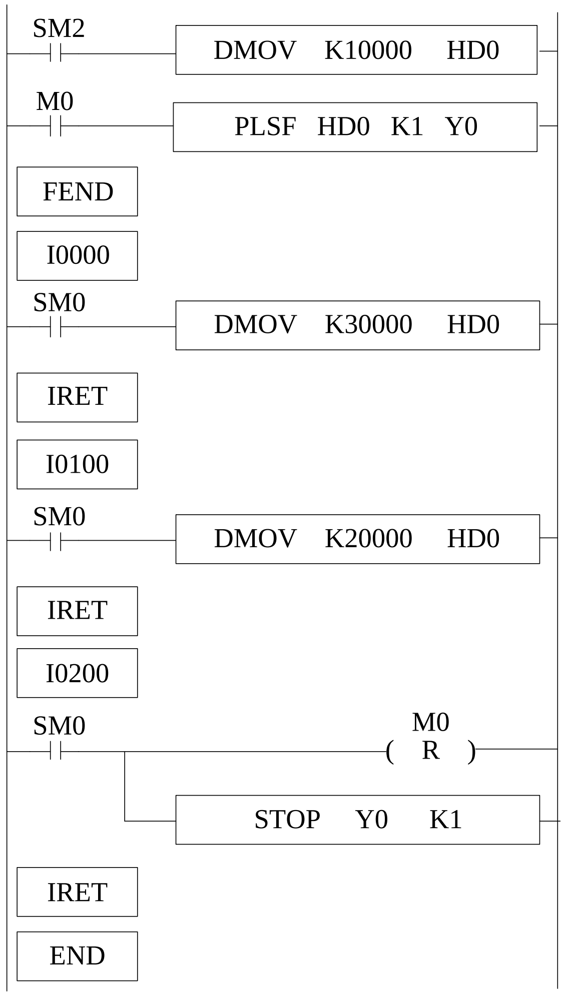
Segment	Frequency setting value (Hz)	Pulse numbers
Origin ---- A	10000	999999999
A---- B	30000	999999999
B ----- C	20000	999999999
Acceleration and deceleratoin time	0	

Note: as the pulse numbers of each segment is unknown, the pulse numbers should set large enough to ensure the object can move to the proximity switch. The STOP instruction will be run by external interruption when the object gets to position C.

Component explanation

PLC component	Control explanation	Mark
M0	Start button, PLSF will send pulse when the button is pressed	
HD0	the PLSF pulse frequency register	

Program

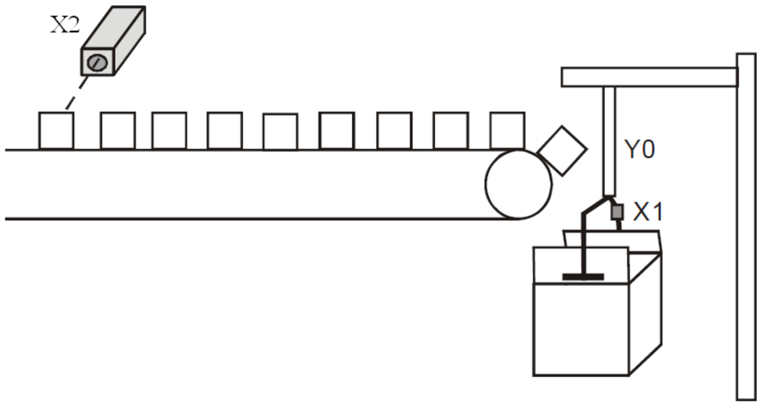


Program explanation

1. SM2 is ON, set HD0 to 10000, set on M0, PLSF instruction will send 10000Hz pulse, the object will move from origin to A.
2. When the object touches A, X0 will be ON at once, the external interruption I0000 will work, HD0 is set to 30000, the object will move from A to B with the speed of 30000Hz.
3. When the object touches B, X1 will be ON at once, the external interruption I0100 will work, HD0 is set to 20000, the object will move from B to C with the speed of 20000Hz.
4. When the object touches C, X2 will be ON at once, the external interruption I0200 will work, M0 is set OFF, the pulse sending will stop at once.

Example 2

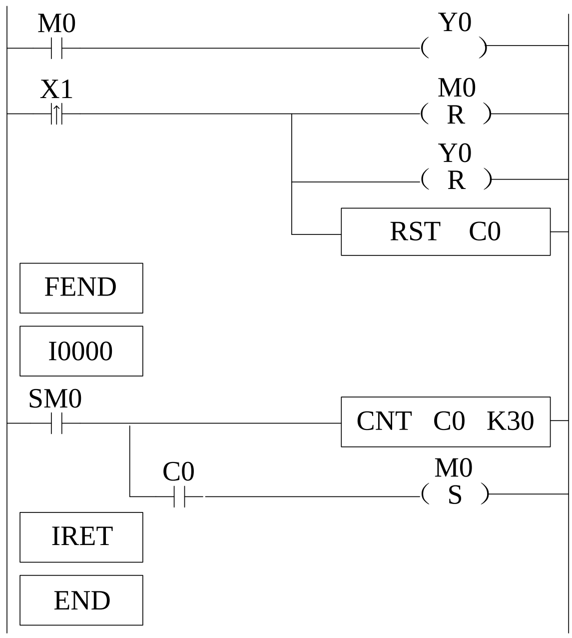
The diagram is the product packing machine. The robot will pack the product when 30 products are detected, the robot and counter will be reset after packing completed. To improve the working efficiency, the product sending speed is very fast, the sensor X2 detects the product time is 8ms, PLC input terminal filter time is 10ms, the normal counter cannot detect the products. We can use the external interruption to count the products.



Component explanation:

PLC component	Control explanation	Mark
X2	Product counting photoelectric sensor, X2 is ON when the product is detected	
X1	Robot action complete sensor, X1 is ON when the action is completed	
C0	16-bit counter	
Y0	Robot	

Program:



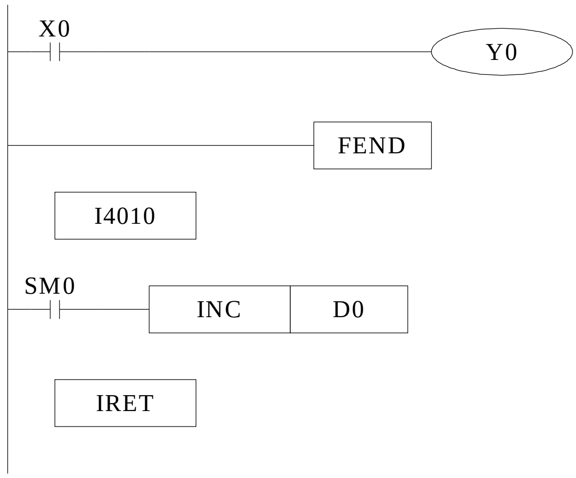
Program explanation:

1. In the external interruption program, count the X2 input, when the X2 is 30, set ON M0
2. In the main program, it controls the Y0 according to the M0 state.
3. When the robot action is completed, X1 changes from OFF to ON once, RST works, Y0 and C0 are reset, M0 is OFF, wait for the next packing process.

10-4-2. Timing Interruption

Function and Action

Under the circumstance that the main program execution cycle is very long, when you have to handle with special program or execute specific program every once in a while when program is scanning in sequence control, the timing interruption is very useful. It is not affected by PLC scan cycle and executes timing interruption subroutine every N ms.



- Timing interruption is open status in default, just like other interruption subroutines, it should be written behind the main program, starts with I40xx, ends with IRET.
- There are 20 channels of timing interruptions, representation: I40**~I59**('**'means interruption time; Unit is ms. E.g: I4010 means executing once the first timing interruption per 10ms.

Interruption No

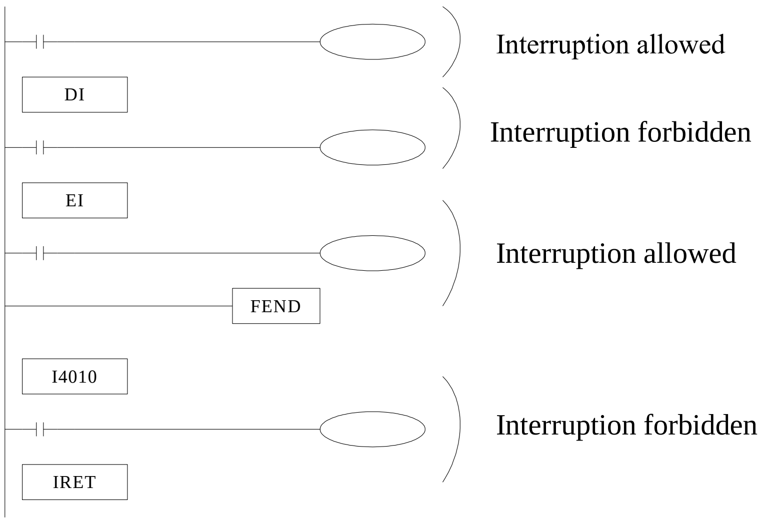
XD, XL series timing interruption:

Interruption number	Interruption ban instruction	Interruption number	Interruption ban instruction	Explanation
I40**	SM070	I50**	SM080	** means the timing interruption time, the range is 1~99, the unit is ms.
I41**	SM071	I51**	SM081	
I42**	SM072	I52**	SM082	
I43**	SM073	I53**	SM083	
I44**	SM074	I54**	SM084	
I45**	SM075	I55**	SM085	
I46**	SM076	I56**	SM086	
I47**	SM077	I57**	SM087	

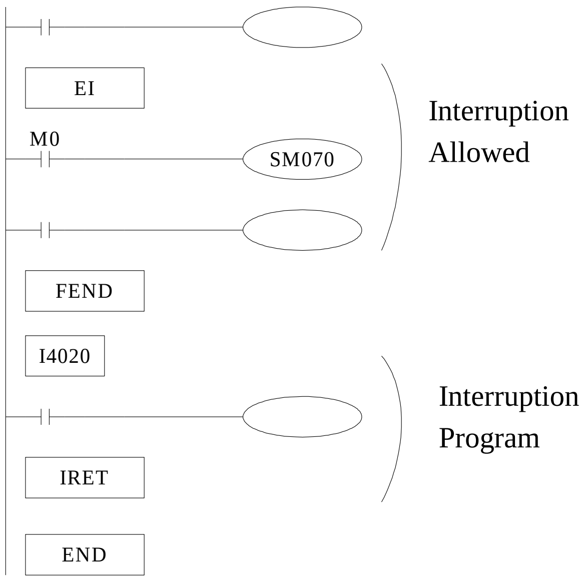
I48**	SM078	I58**	SM088	
I49**	SM079	I59**	SM089	

Interruption range's limitation

- Timing interruption is usually on ‘allow’ status.
- Can set interruption allow and forbidden area with EI、DI instructions. As shown in below pictures, all timing interruptions are forbidden between DI and EI, and allowed beyond DI~EI.



Interruption Forbidden



- The first 3CH timing interruptions are equipped with special relays (SM070~SM079).
- In the left example, if use M0 to set SM070 “ON”, then forbid timing interruption forbidden.

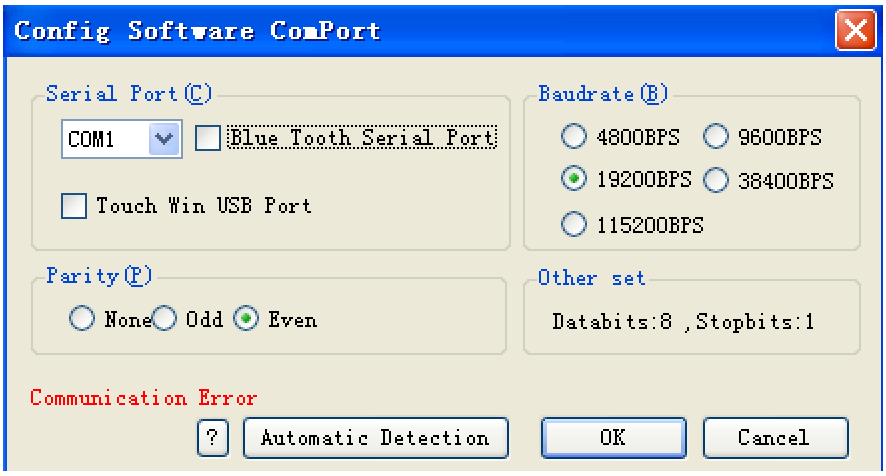
11 Common Questions and Answers

This chapter mainly introduces XD/XL series PLC common questions and answers.

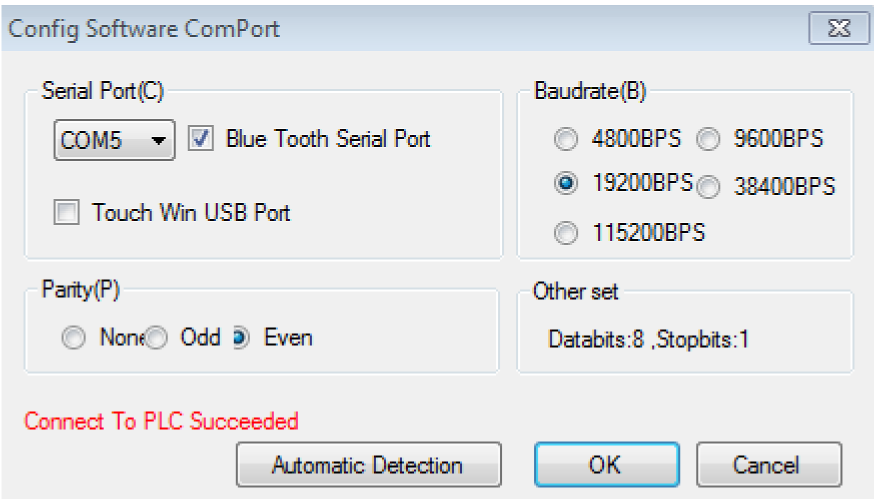
Q1: How to connect PLC with PC?

A1:

If your PC is desktop computer, you can use our company special DVP or XVP cables to connect PC and PLC (Usually PORT1) as general commercial desktop computer has 9 needle serial port. After connecting DVP correctly, power on PLC, click 'Config Software ComPort  , the following window will jump out:



Choose correct communication serial port according to your PC actual serial port.; baud rate selects 19200BPS, parity check selects even parity, 8 data bits, 1 stop bit; you can also click 'check' button directly in the window, and communication parameters will be selected by PLC itself. 'Connect PLC successfully' will be displayed on the left bottom of window as below:



Then it means that PLC has been connected to PC successfully!

Usage method of notebook PC with 9-pin serial port is the same with desktop PC's.

If the notebook does not have 9-pin serial port, users can use USB converter to realize connection between PLC and notebook USB port. Make sure to install USB converter drive software (Xinje special USB converter module COM-USB is recommended， USB converter drive software can be downloaded on Xinje official website)!

Q2: PC cannot connect PLC via RS232 port, it shows offline status?

A2:

Several possible reasons:

Users may changed the communication parameters of PORT1 in PLC (Do not change Port1 communication parameters, or it may lead to connection between PC and PLC failure!)

USB converter driver software was installed incorrectly or USB converter cable is not good

PORT1 communication of PLC is damaged

The download communication cable brand is not Xinje XVP cable.

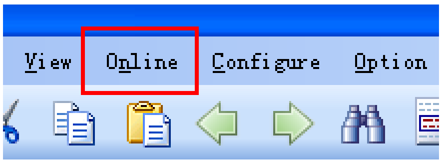
Solutions:

At first, use Xinje XVP cable to connect PC and PLC;

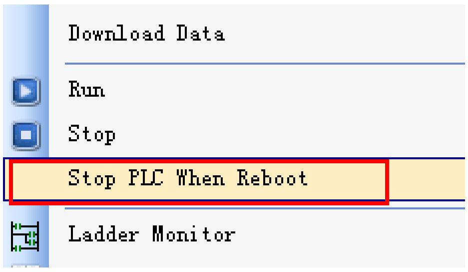
After confirming the connection cable is the Xinje special XVP cable and USB convertor has been used, you can use it to try to connect desktop PC with 9-needle serial port to PLC. If the desktop PC can be connected correctly, please change the USB converter cable with higher performance or install the USB converter serial driver software again.

If PLC can not connect with desktop computer correctly either, you can use ‘stop PLC when reboot’ function to stop PLC and recover the PLC to factory setting, operating method is as follow:

Power on PLC and connect PLC by DVP cables, then click ‘online’ button on PLC editing software menu;

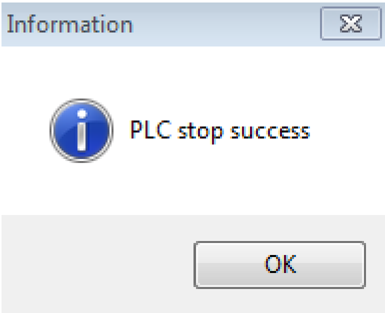


Click ‘Stop when PLC reboot’ from the drop-down menu;

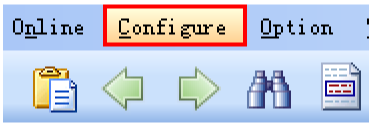


Following window will jump out;

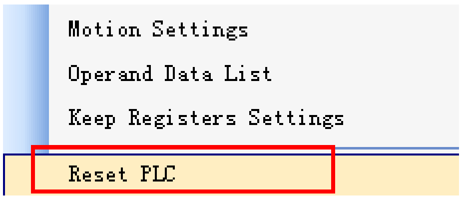
By this time, cut off PLC power for 2-3s and power on again, then a ‘PLC has been stopped successfully’ window will normally jump out; if the window do not jump out after power on, try again a few times until the information window of successful stop jump out.



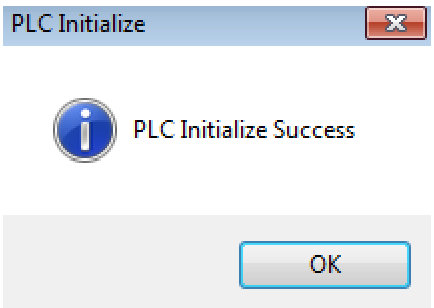
Then click ‘configure’ button ;



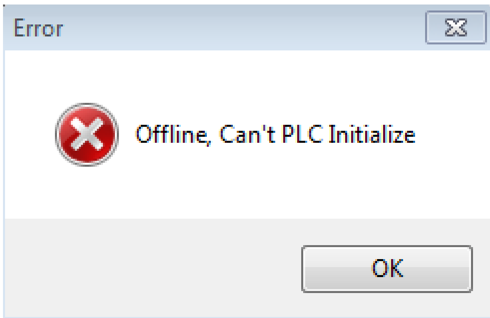
Click ‘Reset PLC’ in the drop-down menu;



By this time, ‘Reset PLC’ information window will jump out and it means that all steps of ‘Stop when PLC reboot’ have been finished.



If initialize PLC unsuccessfully after you trying a few times or the following window jumps out after clicking ‘Reset PLC’:



In both cases, use PLC system update tool to update PLC system, and PLC and PC will be connected successfully if system is updated (For more steps about system update, please refer to Q3 related content).

If update of the desktop computer with 9-pin serial port fails, it is very likely that PLC communication port is damaged, and please contact manufacturer or agent.

Q3：XD/XL series PLC system upgrade

A3：

When does PLC need update usually?

PLC software is in a continuous upgrade stage; if software and hardware version do not match, PLC will not support those upgraded function. About which PLC version the instruction support, please refer to instruction summary in this manual or appendix 2 ‘special function version requirement’;

When users change the communication parameters, PLC and PC can not connect.

When users use ‘program confidential download’ function, however, forget the password (Note：PLC program will disappear after system update！).

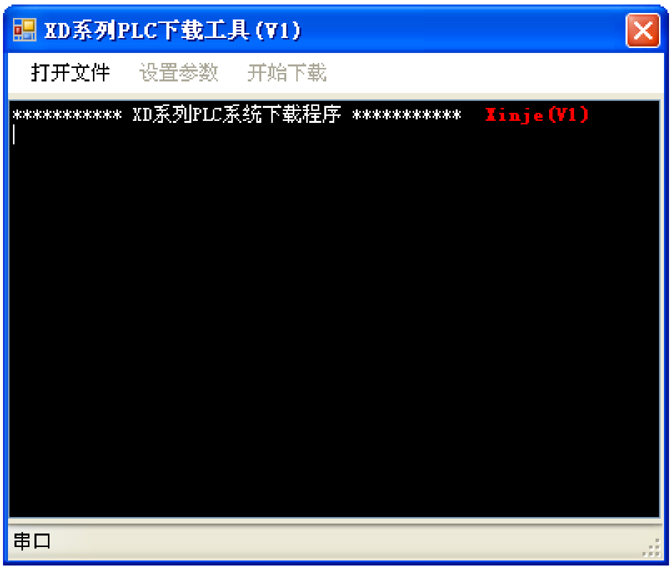
How to update XD/XL series PLC？

PLC update tool:

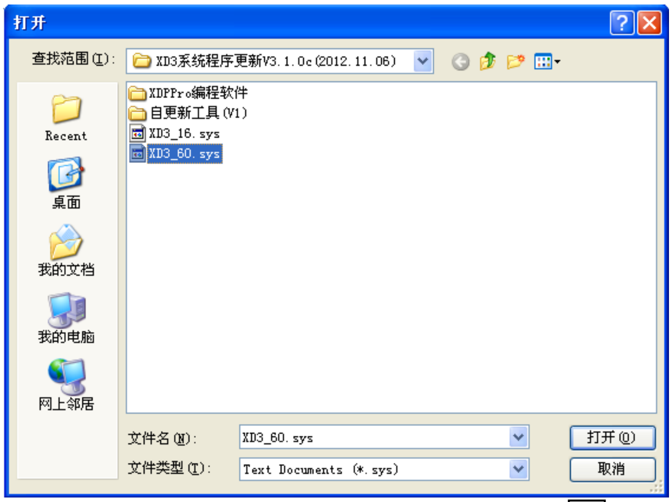
‘XD series PLC download program tool’ and ‘system file’ (*.sys file)

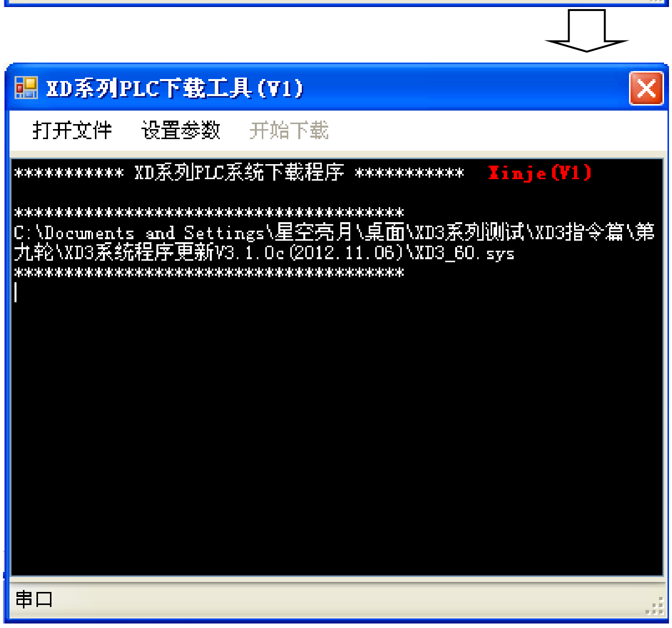
Close all the programs which may occupy the serial port

Cut off the power of PLC, open the XD series update tool (if user use this tool at the first time, please open the enrollment first)



Click "Open File", choose the PLC model for updating. **(Note: XD3_16.sys fit for PLC model XD3-16, XD3_60.sys fit for PLC model XD3-32 and XD3-60):**



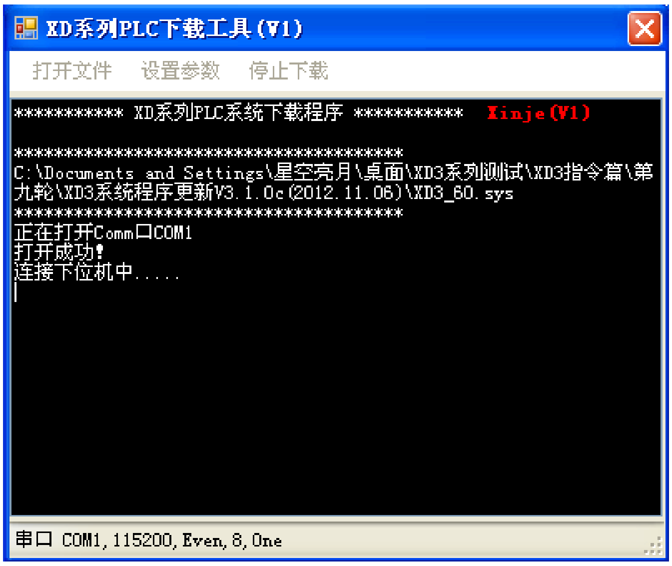


Set the parameters:

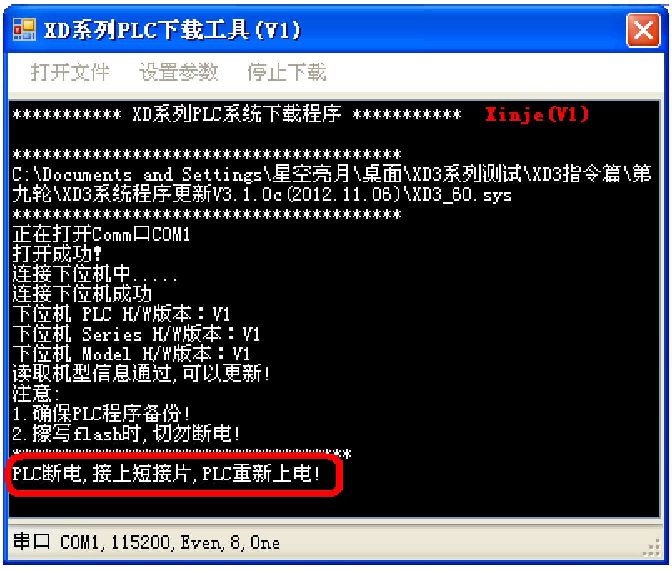
Click “set parameter”, it will show the parameter window:

Note: set the com port, the baud rate is default setting, no need to change.

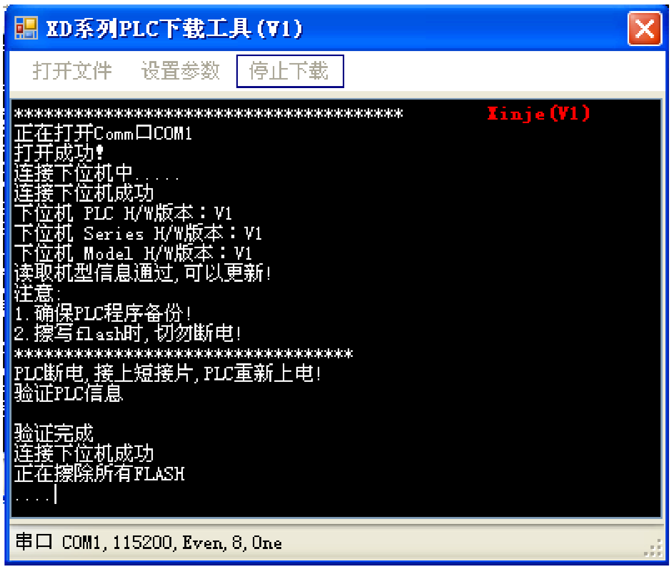
Click “download”, the window will show below words:



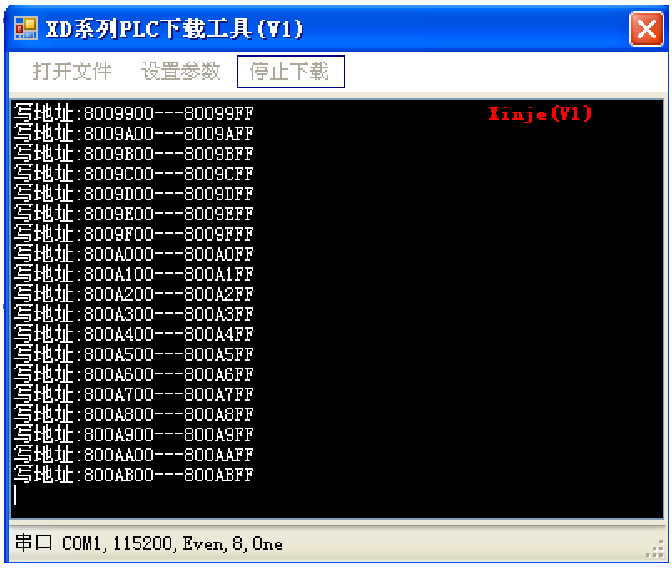
Power on the PLC, the update tool will show below words:



Cut off the power of PLC, connect the short jumper, then power on the PLC again.



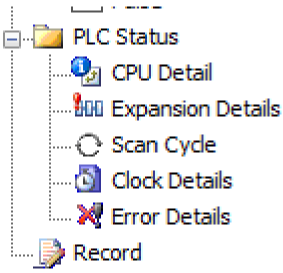
PLC start to update, the updating will take few minutes.

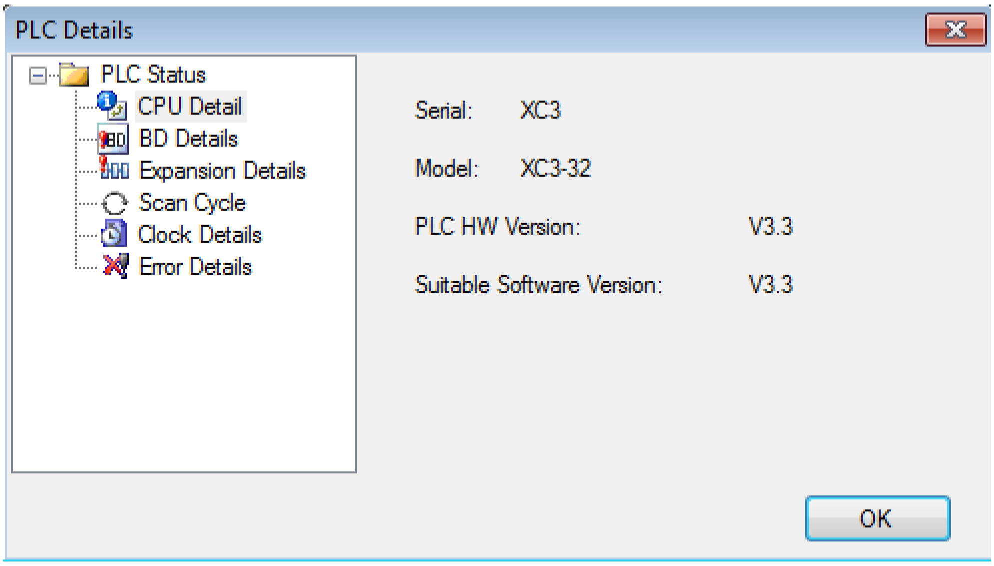


After finishing the update, cut off the PLC power, take off the short jumper, then power on the PLC again.

PLC hardware version

The PLC hardware version can be seen in "CPU detail" on the left window in XDPpro software (PLC online status)





Short jumper

XD, XL series PLC no need to short the jumper when updating.

Note:

Do not cut the power of PLC when it is updating. If it show the error “send data failed, ID not match…) please contact us for help.

The PLC program will be deleted after updating.

Q4: The bit soft component function.

A4:

Continuous 16 coils consist of a word, E.g: DM0 a word consist of 16 coils (bits) M0~M15 is as below:

DM0:

M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

We can use bit in the register directly.

Example 1:

When M100 is from OFF to ON, M0 M1 are ON, M2—M15 are OFF

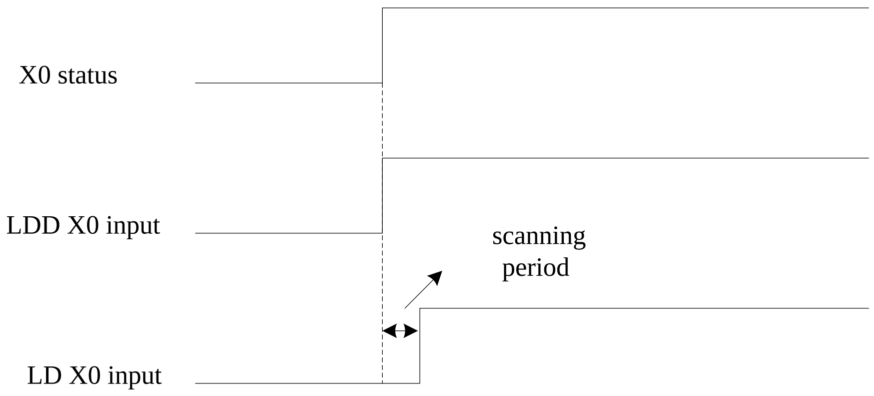
The other mode is bit operation of fixed register. E.g: D0.0 is the first bit of 16 bits in register D0. Similarly, D0.1 is the second bit and so on, as shown below:

D0:

D0.15	D0.14	D0.13	D0.12	D0.11	D0.10	D0.9	D0.8	D0.7	D0.6	D0.5	D0.4	D0.3	D0.2	D0.1	D0.0

Similarly, we can use bit in register D0.

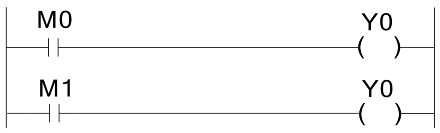
Q5: What's the use of execution instruction LDD/OUTD etc?
A5:
When PLC executes program, state of input point state will map to image register. From then on, PLC will refresh input state at the beginning of every scan cycle; if we use LDD instruction, then the state of input point will not need map to image register; the same with output point (OUTD).
LDD/OUTD instruction usually apply to the occasion that I/O need refresh immediately, which makes the state of input and output avoid the influence of the scan cycle.



Input point X0 sequence chart of LDD and LD

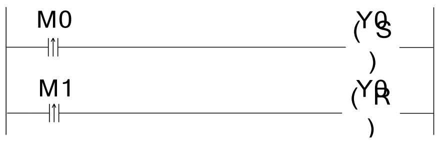
Q6: Why the output LED keeps flashing when using ALT instruction?
A6:
For ALT and many calculation instructions, these instructions will execute every scanning period when the condition is fulfilled (for example, the condition is normal ON coil). We recommend that the condition is rising edge or falling edge.

Q7: Why the M and Y cannot output sometime?
A7:
Output mainly has two ways: 1. OUT instruction; 2. SET instruction. The coil will keep outputting if there is no RST instruction.
Usually in the program, one coil M or Y should use the same output way. Otherwise, the coil cannot output.
For example:



M0 is ON, M1 is OFF, Y0 cannot output
M0 is OFF, M0 is ON, Y0 will output
Reason: two different coils drive the same output coil

Y0 will be ON for one scanning period



M0 is ON, Y will keep outputting
M1 is ON, Y0 is OFF

Q8: Check and change the button battery in the PCB of PLC

A8:

The rated voltage of button battery is 3V. The voltage can be measured by multimeter. If the value of power-loss retentive register is very large, it means the battery is low. Please change the button battery. Users can use SM5 and SD5 to detect the power of button batteries in order to facilitate timely replacement of batteries. See Appendix 1 and Appendix 2 for details.

Q9: Communicate with SCADA software

A9:

If there is no choice for XD/XL series PLC in SCADA software, please choose Modbus-RTU protocol and communicate through RS485 port. Please refer to XD/XL series PLC instruction manual chapter 6.

Q10: MODBUS Communication

A10:

First of all, please ensure that the A and B terminals on the PLC are correctly connected with the RS485 communication terminals of other devices. To modify the parameters of the PORT 2 of the PLC, the following methods are adopted:

Method 1: Configuration by configuration parameter instruction

For specific instructions, please refer to Chapter 6, Communication Functions of this manual. The communication parameter settings of different devices are generally different, so it is important to choose the correct frequency setting mode of communication devices, make clear the corresponding MODBUS communication address and function code, and some communication devices need a given operation signal before displaying the setting frequency.

Method 2: Configuration through control panel (refer to Chapter 6 Communication Function of this manual for specific configuration method).

Q11: The LED light of XD/XL series PLC (PWR/RUN/ERR)

A11:

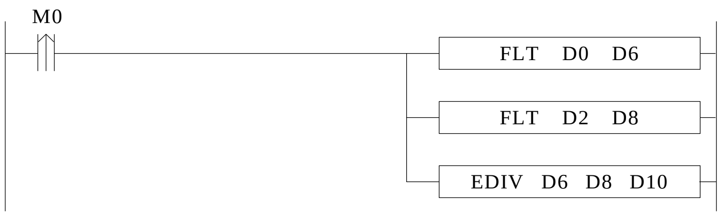
LED light	Problem	Solution
PWR shining, other LED off.	1. I/O PCB has short circuit 2. load is too large for 24V 3. not click RUN for program	Check I/O terminal, if there is short circuit. If the load is too large for 24V power supply. Make sure the program is running inside PLC. Contact us for help.
Three LED all OFF	1. PLC input power supply has short circuit 2. PLC power PCB damaged	Check the input power supply of PLC. Contact us for help.

PWR and ERR light	1. PLC input voltage is not stable 2. there is dead loop in the program 3. PLC system has problem	Check the power supply voltage, check if there is dead loop in the program. Update the hardware of PLC. Contact us for help.

Q12: the result is not correct when doing floating operation

A12:

Please transform the integer to floating number. For example: EDIV D0 D2 D10. If the value of D0 and D2 is integer, the result will has error (D10). Please use below instruction to transform the integer to floating number.

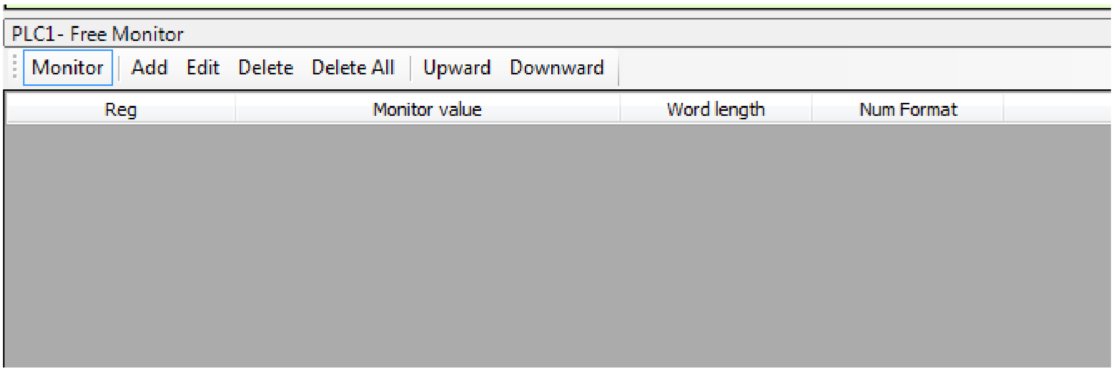


Q13: Why the floating numbers become messy code in online ladder monitor window?

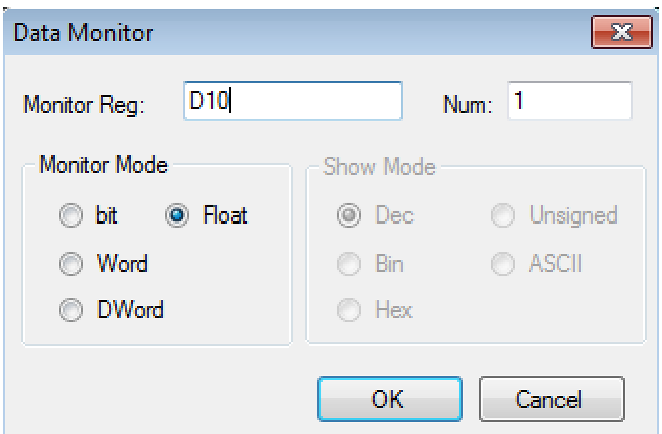
A13:

As the floating number cannot be displayed in online ladder monitoring, please monitor the floating number in free monitor function.

Open XDPpro software, click online/free monitor. The following window will pop up:



Click “add” in the window, the following window will pop up. Set the monitor mode to “float”. Monitor register set to D10. Then click ok.



Q14: Why data errors after using DMUL instructions?

A14:

DMUL operation instruction is 32 bit*32 bit=64 bit operation, the result occupies 4 words, such as: EMUL D0 D2 D10, two multiplier both are 32bit (D1,D0) and (D3, D2), the result is 64 bit (D13, D12, D11, D10), so D10~D13 will be occupied. If these data registers are used latter, operation will error.

Q15: Why the output point action errors after PLC running for a while？

A15:

It’s possible that output terminal is loose, please check.

Q16: Why expansion module does not work while power indicator is ON?

A16:

It is likely the connection of module strips and PLC pins or CPU is not good. Compare the CPU and expansion in cross contrast way to find the problems.

Q17: Why the signal input but cannot see the high speed counter working？

A17:

If high-speed counting is to be carried out, in addition to connecting high-speed pulse to the input of high-speed counting of PLC, the corresponding high-speed counting program should be written with functional instructions. For details, please refer to the relevant content of Chapter 5 of this manual.

Q18: C language advantages compared to ladder chart?

A18:

(1) XD/XL series PLC supports almost all C language functions. When it comes to complex mathematical operations, the advantage of C language is more obvious.

(2) Enhance the confidentiality of the program (when using file-advanced storage mode, C language can not upload);

(3) C language function block can be called in many places and different files, which greatly improves the efficiency of programmers.

Q19: What’s PLC output terminal A, B？

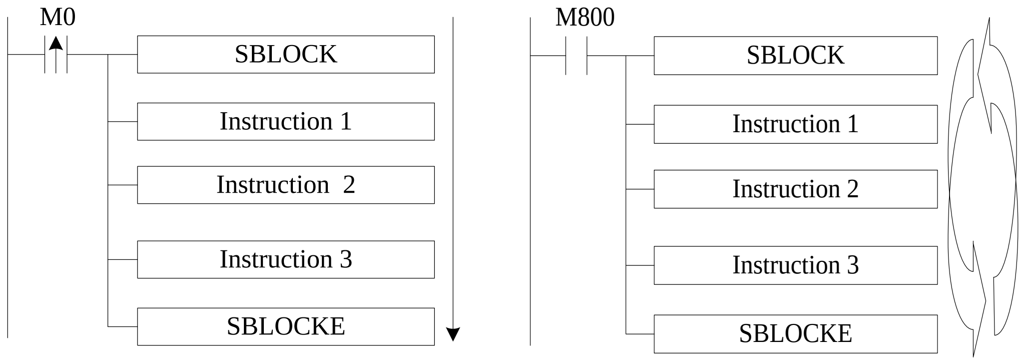
A19:

PLC output terminal A, B are RS485 terminals of PORT2 on PLC.

Q20: What’s the difference of sequence function BLOCK trigger condition: rising edge triggered and normally closed conduction?

A20:

Rising edge triggered: when the condition is triggered, block executes in order from top to bottom; Normally closed conduction: when the condition is triggered, Block will execute in order from top to bottom, return to the top and execute again until the normally closed conduction breaks off. The cycle stops when the last one finished.



From up to down, run the instruction one by one

from up to down, cyclic run the instruction

Q21: What are the download modes of XD/XL series PLC and what are their characteristics?

A21:

XD/XL series PLC has three download modes, which are:

Common download mode

In this mode, you can easily download the program from the computer to the PLC or upload the program from the PLC to the computer. It will be very convenient to use this mode when debugging the equipment.

Password Download Mode

You can set a password for the PLC. When you upload the program from the PLC to the computer, you need to enter the correct password. In the advanced password option, you can also check the function of "download the program needs to be decrypted first" (Note: This operation is dangerous, if you forget the password, your PLC will be locked!). This download mode is suitable for users when they need to keep the device program secret and they can call out the device program at any time.

Secret download mode

In this mode, the program on the computer can be downloaded to the PLC, no matter what way the user can upload the program in the PLC to the computer; at the same time, the user program can be downloaded confidentially, which can occupy less internal resources of the PLC, greatly increase the program capacity of the PLC, and can have a faster download speed; after using this download mode, the program will be completely unable to recover.

Q22: What kinds of confidentiality methods do XD/XL series PLCs have?

A22:

Xinje PLC has three methods of confidentiality: (1) importing and exporting downloaded files; (2) secret downloading; (3) password downloading.

Import and export download files: After saving the PLC program in this way, users can download and use the program, but they can not view and edit the program.

Secret download: After secret downloading to PLC, the program and data in PLC will not be uploaded, indicating that "the program does not exist".

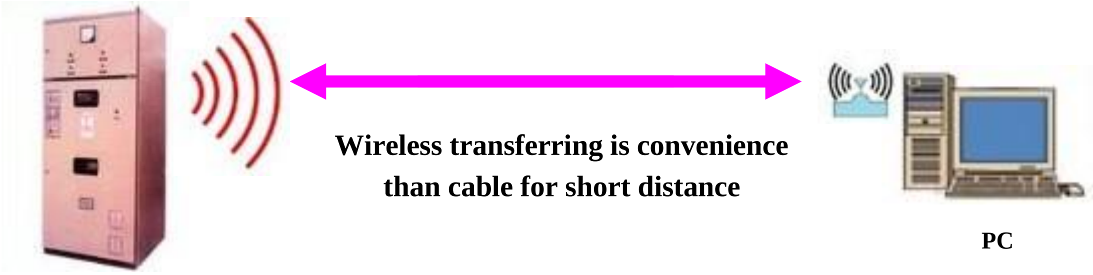
Password download: If you download the program that has set the password to the PLC, you need to input the correct password when uploading the PLC program; if you check "download program needs to be decrypted first", you also need to input the correct password when downloading the new program to the PLC. Under this mode, you can not modify the clock information of the PLC, and the confidentiality is stronger.

Q23: what’s the advantage that XD series PLC replaces DVP download cable with Bluetooth?

A23:

XD series PLC Bluetooth function can perform PLC program download and upload, monitor and Twin configuration software online simulation. The Bluetooth can replace the cable to transfer the data.

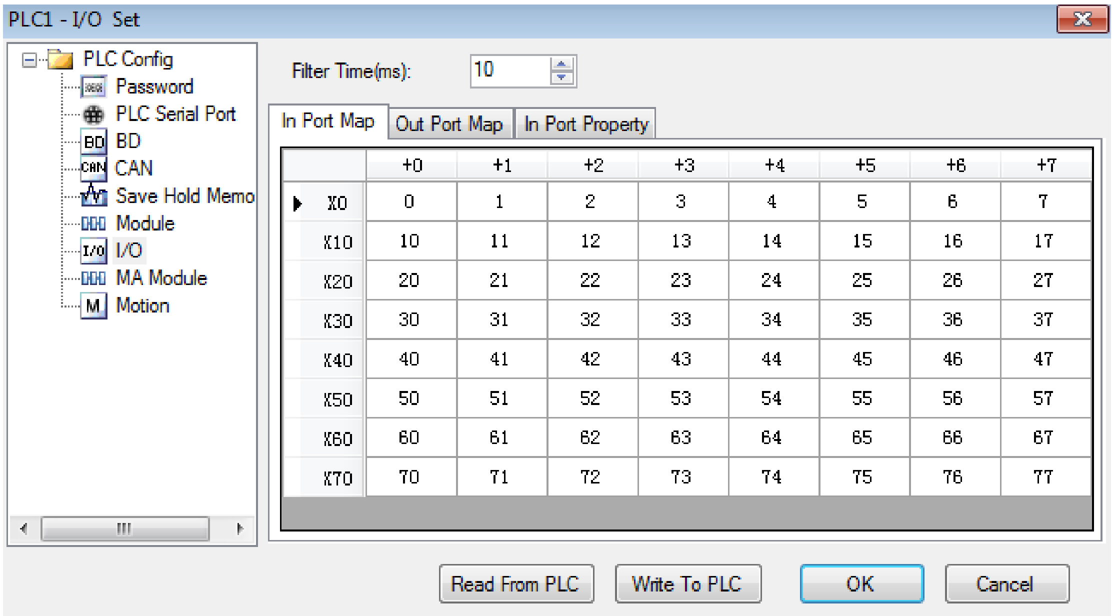
Note: COM-Bluetooth only fit for XINJE PLC.

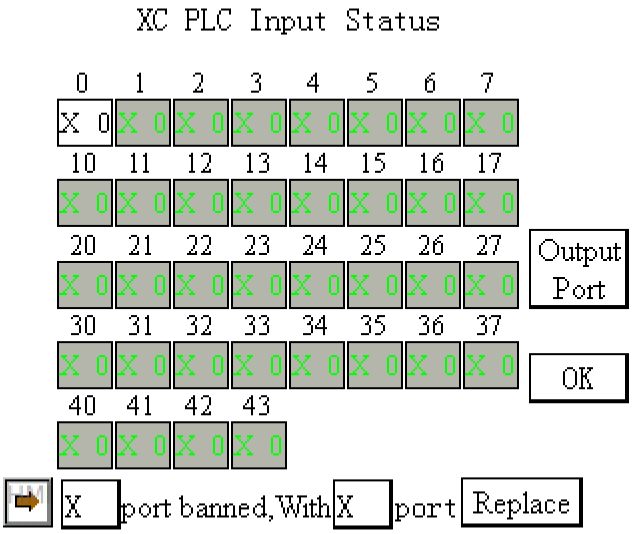


Q24: PLC I/O terminal exchanging

A24:

Sometime the PLC I/O terminals are broken. User don’t have to change the program, PLC I/O terminal exchanging function can solve the problem. User can exchange the terminal through XINJE Touchwin HMI. Open Touchwin software, jump to screen no. 60004 (X terminals) or screen no. 60005 (Y terminals) to set the I/O exchanging.





Touchwin HMI I/O terminal exchanging screen

Q25: What's the function of XD/XL series PLC indirect addressing?

A25:

Adding offset suffix after coils and data registers (Such as X3[D100], M10[D100], D0[D100]) can realize indirect addressing function; such as D100=9, X3[D100] represents X14， M10[D100] represents M19, D0[D100] represents D9; It usually applies to large number of bit and register operation and storage.

Q26: How does XD/XL series PLC connect to the network?

A26:

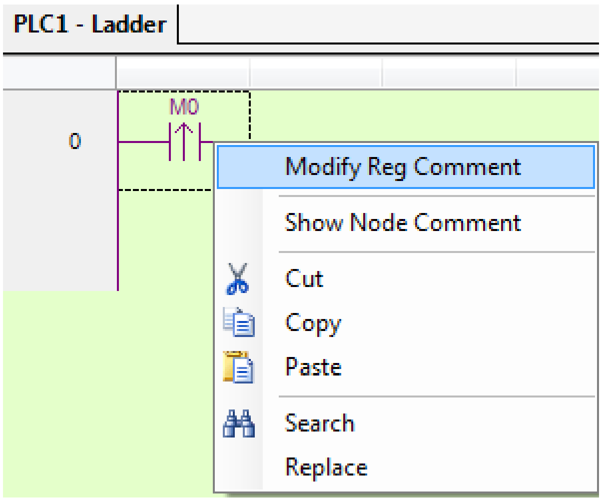
XD/XL series PLC can connect to network by Xinje T-BOX, G-BOX, W-BOX, S-BOX, A-BOX expansion modules or expansion BD boards which have their own communication characteristics. Details please refer to the user manual of communication module or BD board.

Q27: how to add soft element and line note in XDppro software?

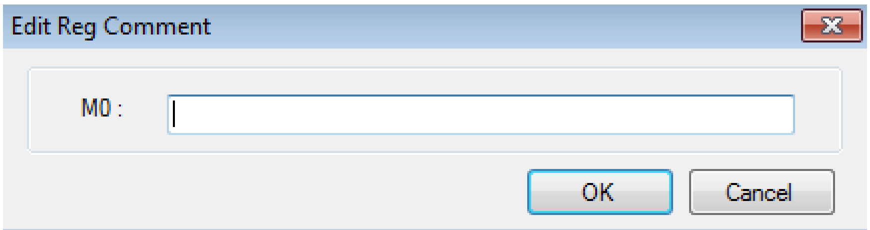
A27:

Soft element note

Open XDPpro software, and move the mouse to the corresponding soft element and right click the mouse, then menu will pop out:

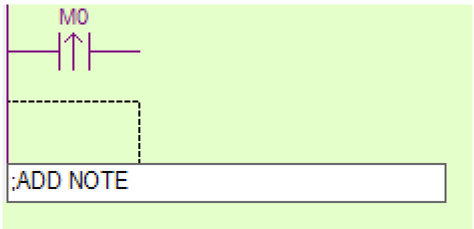


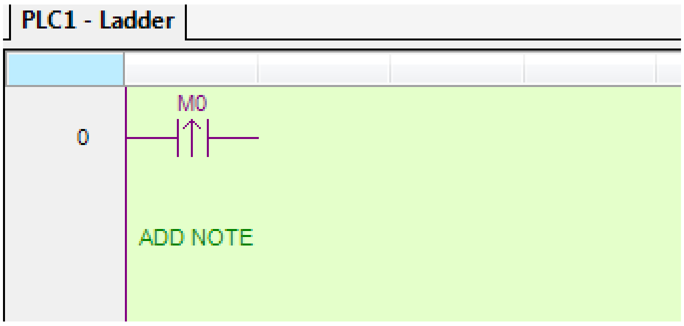
Click “Modify reg comment” to add element notes in below window:



Line note

Line note starts from “;”. Double click the line, then input semicolon and the contents.





Q28: do not have clock function? Why is the clock inaccurate?

A28:

XD/XL series PLC clock function is optional, and if you want to buy the PLC with clock function, please confirm when purchasing. Otherwise, the default PLC when it leaves factory does not have clock function.

If you use a PLC with clock function, check whether the value in register SD13-SD19 is decimal. If not, you need to convert it into decimal through BIN or TRD instructions.

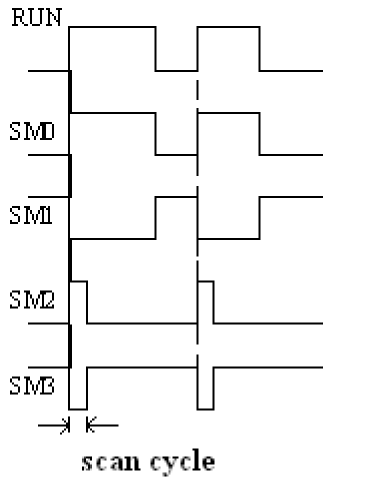
There are some errors in the clock of XD/XL series PLC. The error is about ±5 minutes per month. Please calibrate it by HMI or directly in the PLC program.

Appendix Special soft components

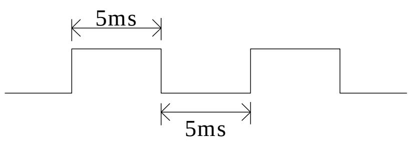
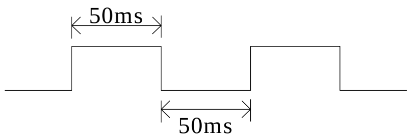
Appendix mainly introduces the functions of XD/XL series PLC special soft element, data register, FlashROM and the address distribution of expansions for users to search.

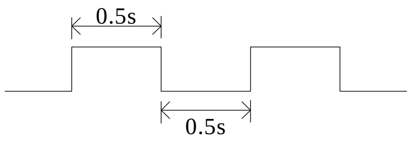
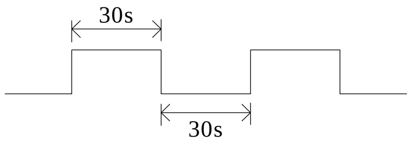
Appendix 1. Special Auxiliary Relay

Initial Status（SM0-SM7）

ID	Function	Description	
SM000	Coil ON when running		SM000 keeps ON when PLC running
SM001	Coil OFF when running		SM001 keeps OFF when PLC running
SM002	Initial positive pulse coil		SM002 is ON in first scan cycle
SM003	Initial negative pulse coil		SM003 is OFF in first scan cycle
SM004	PLC running error	When SM4 sets ON, it indicates that there is an error in the operation of PLC. (Firmware version V3.4.5 and above supports this function by PLC)	
SM005	Battery low alarm coil	When the battery voltage is less than 2.5V, SM5 will put ON (at this time, please replace the battery as soon as possible, otherwise the data will not be maintained)	
SM007	Power-off memory data error		

Clock（SM11-SM14）

ID	Function	Description
SM011	10ms frequency cycle	
SM012	100ms frequency cycle	

SM013	1s frequency cycle	
SM014	1min frequency cycle	

Mark（SM20-SM22）

ID	Function	Description
SM020	Zero bit	SM020 is ON when plus/minus operation result is 0
SM021	Borrow bit	SM021 is ON when minus operation overflows
SM022	Carry bit	SM022 is ON when plus operation overflows

PC Mode（SM32-SM34）

ID	Function	Description
SM032	Retentive register reset	When SM032 is ON, ON/OFF mapping memory of HM、HS and current values of HT、HC、HD will be reset.
SM033	Clear user's program	When SM033 is ON, all PLC user's program will be cleared.
SM034	All output forbidden	When SM034 is ON, all PLC external contacts will be set OFF.

Stepping Ladder

ID	Function	Description
SM040	The process is running	Set ON when the process is running

Interruption ban（SM50-SM90）

ID	Address	Function	Description
SM050	I0000/I0001	Forbid input interruption 0	After executing EI instruction, the input interruption couldn’t act independently when M acts, even if the interruption is allowed. E.g.： when SM050 is ON, I0000/I0001 is forbidden.
SM051	I0100/I0101	Forbid input interruption 1	
SM052	I0200/I0201	Forbid input interruption 2	
SM053	I0300/I0301	Forbid input interruption 3	
SM054	I0400/I0401	Forbid input interruption 4	
……	……	……	
SM069	I1900/I1901	Forbid input interruption 19	
SM070	I40**	Forbid timing interruption 0	After executing EI instruction, the timing interruption couldn’t act independently when M acts, even if the interruption is allowed.
SM071	I41**	Forbid timing interruption 1	
SM072	I42**	Forbid timing interruption 2	
SM073	I43**	Forbid timing interruption 3	
SM074	I44**	Forbid timing interruption 4	
……	……	……	
SM089	I59**	Forbid timing interruption 19	
SM090		Forbid all interruptions	Forbid all interruptions

High Speed Ring Counter（SM99）

address	Function	Note
SM099	High Speed Ring Counting enable	SM99 set ON， SD99 add one per 0.1ms, cycle between 0 and 32767

High speed count complete（SM100-SM109）

Address	Function	Note
SM100	HSC0 count complete flag（100 segments）	
SM101	HSC2 count complete flag（100 segments）	
SM102	HSC4 count complete flag（100 segments）	
SM103	HSC6 count complete flag（100 segments）	
SM104	HSC8 count complete flag（100 segments）	
SM105	HSC10 count complete flag（100 segments）	
SM106	HSC12 count complete flag（100 segments）	
SM107	HSC14 count complete flag（100 segments）	
SM108	HSC16 count complete flag（100 segments）	
SM109	HSC18 count complete flag（100 segments）	

High speed counter direction（SM110-SM119）

Address	Function	Note
SM110	HSC0 direction flag	
SM111	HSC2 direction flag	
SM112	HSC4 direction flag	
SM113	HSC6 direction flag	
SM114	HSC8 direction flag	
SM115	HSC10 direction flag	
SM116	HSC12 direction flag	
SM117	HSC14 direction flag	
SM118	HSC16 direction flag	
SM119	HSC18 direction flag	

High speed counter error（SM120-SM129）

address	Function	Note
SM120	HSC0 error flag	
SM121	HSC2 error flag	
SM122	HSC4 error flag	
SM123	HSC6 error flag	
SM124	HSC8 error flag	
SM125	HSC10 error flag	
SM126	HSC12 error flag	
SM127	HSC14 error flag	
SM128	HSC16 error flag	
SM129	HSC18 error flag	

Communication（SM140-SM193）

	Address	Function	Note
Serial port 0	SM140	Modbus instruction execution flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM141	X-NET instruction execution flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM142	Free format communication sending flag	When the instruction starts to execute, set ON When execution is complete, set OFF
	SM143	Free format communication receive complete flag	When receiving a frame of data or receiving data timeout, set ON. Require user program to set OFF
Serial port 1	SM150	Modbus instruction execution flag	Same to SM140

	SM151	X-NET instruction execution flag	Same to SM141
	SM152	Free format communication sending flag	Same to SM142
	SM153	Free format communication receive complete flag	Same to SM143
Serial port 2	SM160	Modbus instruction execution flag	Same to SM140
	SM161	X-NET instruction execution flag	Same to SM141
	SM162	Free format communication sending flag	Same to SM142
	SM163	Free format communication receive complete flag	Same to SM143
Serial port 3	SM170	Modbus instruction execution flag	Same to SM140
	SM171	X-NET instruction execution flag	Same to SM141
	SM172	Free format communication sending flag	Same to SM142
	SM173	Free format communication receive complete flag	Same to SM143
Serial port 4	SM180	Modbus instruction execution flag	Same to SM140
	SM181	X-NET instruction execution flag	Same to SM141
	SM182	Free format communication sending flag	Same to SM142
	SM183	Free format communication receive complete flag	Same to SM143
Serial port 5	SM190	Modbus instruction execution flag	Same to SM140
	SM191	X-NET instruction execution flag	Same to SM141
	SM192	Free format communication sending flag	Same to SM142
	SM193	Free format communication receive complete flag	Same to SM143

Sequence Function BLOCK（SM240-SM399）

ID	Function	Description
SM300	BLOCK1 running flag	SM300 will be ON when block1 is running
SM301	BLOCK2 running flag	SM301 will be ON when block2 is running
SM302	BLOCK3 running flag	SM302 will be ON when block3 is running
SM303	BLOCK4 running flag	SM303 will be ON when block4 is running
SM304	BLOCK5 running flag	SM304 will be ON when block5 is running
SM305	BLOCK6 running flag	SM305 will be ON when block6 is running
……	……	

SM396	BLOCK97 running flag	SM396 will be ON when block97is running
SM397	BLOCK98 running flag	SM397 will be ON when block98 is running
SM398	BLOCK99 running flag	SM398 will be ON when block99 is running
SM399	BLOCK100 running flag	SM399 will be ON when block100 is running

Error check（SM400-SM412）

ID	Function	Description
SM400	I/O error	ERR LED keeps ON, PLC don not run and output, check when power on
SM401	Expansion module communication error	
SM402	BD communication error	
……		
SM405	No user program	Internal code check wrong
SM406	User program error	Implement code or configuration table check wrong
SM407	SSFD check error	ERR LED keeps ON, PLC don not run and output, check when power on
SM408	Memory error	Can not erase or write Flash
SM409	Calculation error	
SM410	Offset overflow	Offset exceeds soft element range
SM411	FOR-NEXT overflow	Reset when power on or users can also reset by hand.
SM412	Invalid data fill	When offset of register overflows, the return value will be SM372 value

Error Message（SM450-SM463）

ID	Function	Description
SM450	System error check	
SM451	Hardfault interrupt flag	
SM452		
SM453	SD card error	
SM454	Power supply is cut off	
……		
SM460	Extension module ID not match	
SM461	BD/ED module ID not match	
SM462	Extension module communication overtime	
SM463	BD/ED module communication overtime	

Expansion Modules, BD Status（SM500）

ID	Function	Description
SM500	Module status read is finished	

Appendix 2． Special Data Register

Battery （SD5~SD7）

ID	Function	Description
SD005	Battery register	It will display 100 when the battery voltage is 3V, if the battery voltaeg is lower than 2.5V, it will display 0, it means please change new battery at once, otherwise the data will lose when PLC power off.
SD007	Power-off memory data error type	

Clock（SD10-SD019）

ID	Function	Description
SD010	Current scan cycle	100us，us is the unit
SD011	Min scan time	100us，us is the unit
SD012	Max scan time	100us，us is the unit
SD013	Second (clock)	0~59 (BCD code)
SD014	Minute (clock)	0~59 (BCD code)
SD015	Hour (clock)	0~23 (BCD code)
SD016	Day (clock)	0~31 (BCD code)
SD017	Month (clock)	0~12 (BCD code)
SD018	Year (clock)	2000~2099 (BCD code)
SD019	Week (clock)	0(Sunday)~6(Saturday)(BCD code)

Flag （SD020-SD031）

ID	Function	Note
SD020	Model type	
SD021	model（low-8）series（high-8）	
SD022	Compatiable system version（low）system version（high）	
SD023	Compatiable model version（low）model version（high）	
SD024	Model info	
SD025	Model info	

SD026	Model info	
SD027	Model info	
SD028	Suitable software version	
SD029	Suitable software version	
SD030	Suitable software version	
SD031	Suitable software version	

Step ladder（SD040）

ID	Function	Description
SD40	Flag of the executing process S	

High Speed Counting（SD100-SD109）

ID	Function	Description	
SD100	Current segment (No. n segment)		HSC00
SD101	Current segment (No. n segment)		HSC02
SD102	Current segment (No. n segment)		HSC04
SD103	Current segment (No. n segment)		HSC06
SD104	Current segment (No. n segment)		HSC08
SD105	Current segment (No. n segment)		HSC10
SD106	Current segment (No. n segment)		HSC12
SD107	Current segment (No. n segment)		HSC14
SD108	Current segment (No. n segment)		HSC16
SD109	Current segment (No. n segment)		HSC18

High speed counter error（SD120-SD129）

ID	Function	Note
SD120	HSC0 error info	
SD121	HSC2 error info	
SD122	HSC4 error info	
SD123	HSC6 error info	
SD124	HSC8 error info	
SD125	HSC10 error info	
SD126	HSC12 error info	
SD127	HSC14 error info	
SD128	HSC16 error info	
SD129	HSC18 error info	

communication（SD140~SD199）

	ID	Function	Note
Serial port 0	SD140	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD141	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD142	Free format communication send result	0: correct 410: free format send buffer overflow
	SD143	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD144	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD149		
Serial port 1	SD150	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD151	X-Net communication result	0: correct 1: communication overtime

			2: memory error 3: receive CRC error
	SD152	Free format communication send result	0: correct 410: free format send buffer overflow
	SD153	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD154	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD159		
Serial port 2	SD160	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error（erase FLASH）
	SD161	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD162	Free format communication send result	0: correct 410: free format send buffer overflow
	SD163	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start character 416: no end character
	SD164	Free format communication receive data numbers	In bytes, there are no start and stop characters
	……		
	SD169		
Serial port 3	SD170~SD179		

Serial port 4	SD180~SD189		
Serial port 5	SD190~SD199		

Sequence Function Block（SD300-SD399）

ID	Function	Description
SD300	Executing instruction of BLOCK1	The value will be used when BLOCK monitors
SD301	Executing instruction of BLOCK2	The value will be used when BLOCK monitors
SD302	Executing instruction of BLOCK3	The value will be used when BLOCK monitors
SD303	Executing instruction of BLOCK4	The value will be used when BLOCK monitors
SD304	Executing instruction of BLOCK5	The value will be used when BLOCK monitors
SD305	Executing instruction of BLOCK6	The value will be used when BLOCK monitors
……	……	……
SD396	Executing instruction of BLOCK97	The value will be used when BLOCK monitors
SD397	Executing instruction of BLOCK98	The value will be used when BLOCK monitors
SD398	Executing instruction of BLOCK99	The value will be used when BLOCK monitors
SD399	Executing instruction of BLOCK100	The value will be used when BLOCK monitors

Error Check（SD400-SD413）

ID	Function	Note
SD400		
SD401	Extension module no. of communication error	Means module no.n is error
SD402	BD/ED module no. of communication error	
SD403	FROM/TO error type	
SD404	PID error type	
……		
SD409	Calculation error code	1: divide by 0 error 2: MRST, MSET front operand address less than back operand 3: ENCO, DECO data bits of encoding and decoding instructions exceed the limit. 4: BDC code error 7: Radical sign error
SD410	The number of offset register D when offset crosses the boundary	
SD411		

SD412	Invalid data fill value (low 16 bits)	
SD413	Invalid data fill value (high 16 bits)	

Error Check（SD450-SD452）

ID	Function	Description
SD450	1：Watchdog act (Default 200ms) 2：Control block application fail 3：Visit illegal address	
SD451	Hardware error type: 1：Register error 2：Bus error 3：Usage error	
SD452	Hardware error	
SD453	SD card error	
SD454	Power-off time	
SD460	Extension module ID not match	
SD461	BD/ED module ID not match	
SD462	Extension module communication overtime	
SD463	BD/ED module communication overtime	

Expansion Modules, BD Status（SD500-SD516）

ID	Function	Description	
SD500	Module number Expansion modules： #10000～10015 BD：#20000～20001 ED: #30000		
SD501~516	Expansion module, BD /ED status		16 registers

Module info（SD520-SD823）

ID	Function	Explanation	Note
SD520~SD535	Extension module info	Extension module 1	Each extension module, BD, ED occupies 16 registers
……	……	……	
SD760~SD775	Extension module info	Extension module 16	
SD776~SD791	BD module info	BD module 1	
SD792~SD807	BD module info	BD module 2	
SD808~SD823	ED module info	ED module 1	

Expansion Module Error Information

ID	Function	Description	
SD860	Error times of module read		Expansion module 1
SD861	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD862	Error times of module write		
SD863	Error types of module write		
SD864	Error times of module read		Expansion module 2
SD865	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD866	Error times of module write		
SD867	Error types of module write		
……			
SD920	Error times of module read		Expansion module 16
SD921	Error types of module read	Module address error. Module accepted data length error. Module CRC parity error when PLC is accepting data. Module ID error. Module overtime error.	
SD922	Error times of module write		
SD923	Error types of module write		
SD924	Error times of module read		BD module 1
SD925	Error types of module read		
SD926	Error times of module write		
SD927	Error types of module write		
SD928	Error times of module read		BD module 2
SD929	Error types of module read		
SD930	Error times of module write		
SD931	Error types of module write		
SD932	Error times of module read		ED module 1
SD933	Error types of module read		
SD934	Error times of module write		
SD935	Error types of module write		

Version info（SD990~SD993）

ID	Function	Explanation	Note
SD990	Firmware version date	Low 16-bit	
SD991	Firmware version compilation date	High 16-bit	
SD992	FPGA version compilation date	Low 16-bit	
SD993	FPGA version compilation date	High 16-bit	

Appendix 3. Special Flash Register

Special FLASH data register SFD

*** means it works only after repower on the PLC**

I filtering

ID	Function	Description
SFD0*	Input filter time	
SFD2*	Watchdog run-up time, default value is 200ms	

I Mapping

ID	Function	Description	
SFD10*	I00 corresponds to X**	Input terminal 0 corresponds to X** number	0xFF means terminal bad，0xFE means terminal idle
SFD11*	I01 corresponds to X**		
SFD12*	I02 corresponds to X**		
……	……		
SFD73*	I77 corresponds to X**	Default value is 77（Octonary）	

O Mapping

ID	Function	Description	
SFD74*	O00 corresponds to Y**	Output terminal 0 correspond to Y** number	0xFF means terminal bad，0xFE means terminal idle
		Default value is 0	
……	……		

SFD134*	O77 corresponds to Y**	Default value is 77 （Octonary）	

I Attribute

ID	Function	Description	
SFD138*	I00 attribute	Attribute of input terminal 0	0：positive logic others：negative logic
SFD139*	I01 attribute		
……	……		
SFD201*	I77 attribute		

High Speed Counting

ID	Function	Description
SFD320	HSC0 frequency times	2：2 times frequency；4: 4 times frequency(effective at AB phase counting mode)
SFD321	HSC2 frequency times	Ditto
SFD322	HSC4 frequency times	Ditto
SFD323	HSC6 frequency times	Ditto
SFD324	HSC8 frequency times	Ditto
SFD325	HSC10 frequency times	Ditto
SFD326	HSC12 frequency times	Ditto
SFD327	HSC14 frequency times	Ditto
SFD328	HSC16 frequency times	Ditto
SFD329	HSC18 frequency times	Ditto
SFD330	Bit selection of HSC absolute and relative（24 segment）	bit0 corresponds to HSC0，bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：relative 1：absolute
SFD331	Interrupt circulating of 24 segments high speed counting	bit0 corresponds to HSC0，bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：single 1：loop
SFD332	CAM function	bit0 corresponds to HSC0，bit1corresponds to HSC2，and so on, bit9 corresponds to HSC18 0：do not support CAM function 1：support CAM function

Expansion Module Configuration

ID	Function	Explanation
SFD340	Extension module configuration status（#1#2）	Configuration Status of Extension Modules 1 and 2
SFD341	Extension module configuration status（#3#4）	Configuration Status of Extension Modules 3 and 4
……	……	……
SFD347	Extension module configuration status（#15#16）	Configuration Status of Extension Modules 15 and 16

SFD348	BD module configuration status （#1#2）	Configuration Status of BD Modules 1 and 2
SFD349	ED module configuration status （#1）	Configuration Status of ED Module 1
SFD350 ： SFD359	Extension module configuration	Configuration of Extension Module 1
SFD360 ： SFD369	Extension module configuration	Configuration of Extension Module 2
：	：	
SFD500 ： SFD509	Extension module configuration	Configuration of Extension Module 16
SFD510 ： SFD519	BD module configuration	Configuration of BD Module 1
SFD520 ： SFD529	BD module configuration	Configuration of BD Module 2
SFD530 ： SFD539	ED module configuration	Configuration of ED Module 1

Communication

ID	Function	Note
SFD600	COM1 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD610	COM2 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD620	COM3 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD630	COM4 free format communication buffer bit numbers	0: 8-bit 1: 16-bit
SFD640	COM5 free format communication buffer bit numbers	0: 8-bit 1: 16-bit

Appendix 4. PLC resource conflict table

When PLC is used in practice, conflicts may arise because some resources are used at the same time. This section will list the resources that may cause conflicts in each PLC model. This part mainly refers to high-speed counting, accurate timing and pulse output.

	Accurate timing	High speed counting				Pulse output	
XD2-16, XD3-16, XD5-16, XL3-16							
	ET0	-	-	-	-	-	-
	ET2						
	ET4						
	ET6						
	ET8	HSC0					
	ET10		HSC2				
	ET12			HSC4			
	ET14					Y0	
	ET16					Y0	
	ET18					Y1	
	ET20					Y1	
	ET22						
	ET24						
XD3-24/32/48/60, ZG3-30							
	ET0						
	ET2						
	ET4						
	ET6						
	ET8						
	ET10						
	ET12	HSC0					
	ET14		HSC2				
	ET16			HSC4			
	ET18					Y0	
	ET20					Y0	
	ET22					Y1	
	ET24					Y1	
XD5-24/32/48/60, XDM-24/32/48/60, XD5E-30/60, XDME-60, XL5-32, XL5E-32							
	ET0	-	-	-	-	-	-
	ET2				HSC6		
	ET4			HSC4			
	ET6		HSC2				
	ET8	HSC0					
	ET10					Y3	
	ET12					Y3	
	ET14					Y2	
	ET16					Y2	
	ET18					Y1	
	ET20					Y1	
	ET22					Y0	
	ET24					Y0	
XDC-24/32/48/60							

	ET0	-	-	-	HSC6	-	-
	ET2			HSC4			
	ET4		HSC2				
	ET6	HSC0					
	ET8					Y3	
	ET10					Y3	
	ET12					Y2	
	ET14					Y2	
	ET16					Y1	
	ET18					Y1	
	ET20					Y0	
	ET22					Y0	
	ET24						

※1：This form should be read horizontally. Any two resources in each row cannot be used at the same time. Otherwise, it will cause conflict.

Appendix 5. PLC function configuration list

This part is used to check each model's configurations. Via this table, we can judge products type easily.

○ Selectable × Not support √ Support

series	USB port	232 port	485 port	RJ 45	Ex module	BD	High speed counter		Pulse output Channel(T /RT)	External interruption
							Incremental mode	AB phase		
XD1										
XD1-16	×	2	×	×	×	×	×	×	×	6
XD1-32	×	2	√	×	×	×	×	×	×	10
XD2										
XD2-16	×	2	√	×	×	×	3	3	2	6
XD2-24	×	2	√	×	×	1	3	3	2	10
XD2-32	×	2	√	×	×	1	3	3	2	10
XD2-48	×	2	√	×	×	2	3	3	2	10
XD2-60	×	2	√	×	×	2	3	3	2	10
XD3										
XD3-16	1	1	√	×	10	×	3	3	2	6
XD3-24	1	1	√	×	10	1	3	3	2	10
XD3-32	1	1	√	×	10	1	3	3	2	10
XD3-48	1	1	√	×	10	2	3	3	2	10
XD3-60	1	1	√	×	10	2	3	3	2	10
XD5										
XD5-16	1	1	√	×	16	×	3	3	2	10

XD5-24	1	1	√	×	16	1	3	3	2	10
XD5-32	1	1	√	×	16	1	3	3	2	10
XD5-48	1	1	√	×	16	2	3	3	2	10
XD5-60	1	1	√	×	16	2	3	3	2	10
XD5-24T4	1	1	√	×	16	1	4	4	4	10
XD5-32T4	1	1	√	×	16	1	4	4	4	10
XD5-48T4	1	1	√	×	16	2	4	4	4	10
XD5-48T6	1	1	√	×	16	2	6	6	6	10
XD5-60T4	1	1	√	×	16	2	4	4	4	10
XD5-60T6	1	1	√	×	16	2	6	6	6	10
XD5-60T10	1	1	√	×	16	2	10	10	10	10
XDM										
XDM-24T4	1	1	√	×	16	1	4	4	4	10
XDM-32T4	1	1	√	×	16	1	4	4	4	10
XDM-60T4	1	1	√	×	16	2	4	4	4	10
XDM-60T4L	1	1	√	×	16	2	4	4	4	10

series	USB port	232 port	485 port	RJ 45	Ex module	BD	High speed counter		Pulse output Channel(T /RT)	External interruption
							Incremental	AB phase		
XDM										
XDM-60T10	1	1	√	×	16	2	10	10	10	10
XDC										
XDC-24	×	2	√	×	16	1	4	4	2	10
XDC-32	×	2	√	×	16	1	4	4	2	10
XDC-48	×	2	√	×	16	2	4	4	2	10
XDC-60	×	2	√	×	16	2	4	4	2	10
XD5E										
XD5E-30T4	1	1	√	1	16	1	4	4	4	10
XD5E-60T10	×	1	√	2	16	2	10	10	10	10
XDME										
XDME-60T10	×	1	√	2	16	2	10	10	10	10
XL1										
XL1-16	×	1	√	×	×	×	×	×	×	6
XL3										
XL3-16	1	1	√	×	10	×	3	3	2	6
XL5										
XL5-32T4	1	1	√	×	16	1	4	4	4	10
XL5E										
XL5E-32T4	×	1	√	2	16	1	4	4	4	10
XLME										
XLME-32T4	×	1	√	2	16	1	4	4	4	10

WUXI XINJE ELECTRIC CO., LTD.
4th Floor Building 7,No. 100 Dicui
Road,Wuxi, China
214072
Tel: (510) 85134139
Fax: (510) 85111290
Email: fiona.xinje@vip.163.com
Web: www.xinje.com