VEICHI
DRIVE FOR EVER

Version change record

Version	release time	Update content
A1.0	2019.08.28	First release

CHAPTER 1 PREFACE 1

CHAPTER 2 SUMMARY 4

2.1 INTRODUCTION TO PROGRAMMABLE CONTROLLER 4
2.2 BASIC CONTROL PRINCIPLE 5
2.2.1 How the programmable controller works 5
2.2.2 User program control principle 6
2.3 PROGRAMMING SOFTWARE 6

CHAPTER 3 SOFT COMPONENTS 15

3.1 LIST OF SOFT COMPONENTS 16
3.2 INPUT AND OUTPUT RELAY 18
3.2.1 Input Relay X 18
3.2.1 Output Relay Y 18
3.3 AUXILIARY RELAY M 18
3.4 STATUS RELAY S 19
3.5 TIMER T 19
3.6 COUNTER C 21
3.6.1 16bit counter 22
3.6.2 32bit counter 23
3.6.3 High number counter 24
3.7 REGISTER 24
3.7.1 Data Register D 24
3.7.2Index register V, Z 25
3.7.3 File register R 26
3.8 LABELS AND SUBROUTINES 26
3.9 CONSTANT 27

CHAPTER 4 INSTRUCTION 29

4.1 PROGRAM LOGIC INSTRUCTIONS 29
4.1.1 Contact Instructions 30
4.1.2 Combined instructions 39
4.1.3 Output Instructions 41
4.1.4 Other processing instructions 46
4.2 PROGRAM PROCESS INSTRUCTION 46
4.2.1 Subroutine 47
4.2.2 Interrupt 49
4.2.3 Jump 50
4.2.4 cycle 52
4.3 DATA COMPARISON INSTRUCTIONS 54
4.3.1 Contact comparison 56
4.3.2 Output comparison 69
4.4 DATA OPERATION 74
4.4.1 Arithmetic 75
4.4.2 Data logic operation 84
4.4.3 Trigonometric function 88

4.4.4 Form operation....99
4.4.5 Index operation....107
4.5 DATA PROCESSING....112
4.5.1 Data conversion....113
4.5.2 Data Transfer....130
4.5.3 Table operation....137
4.5.4 Data shift....138
4.5.5 Other data processing....142
4.6 CLOCK INSTRUCTION....145
4.7 BIT INSTRUCTION FOR PULSE LOCATION....149
4.7.1 Pulse output....149
4.7.2 Pulse location....149
4.7.3 Refresh processing....149
4.8 COMMUNICATION....152
4.9 PERIPHERAL DEVICE....153
4.9.1 PID Operation....153
4.9.2 Other peripheral instructions....158
4.10 ELECTRONIC CAM COMMAND....159

CHAPTER 5 HIGH SPEED OUTPUT AND BIT INSTRUCTION....160

5.1 INSTRUCTION OVERVIEW....160
5.1.1 High speed output instruction attribute table....160
5.1.2 Description of pulse output port....161
5.1.3 Special soft element of pulse output port....161
5.1.4 Output frequency and acceleration and deceleration time....163
5.2 LIST OF POSITIONING INSTRUCTIONS....163
5.2.1 List of positioning instructions....163
5.2.2 Plsy pulse output command....164
5.2.3 PLSV variable pulse output command....167
5.2.4 Plsv2 variable pulse output command with acceleration and deceleration....169
5.2.5 PLSR with acceleration and deceleration pulse output....172
5.2.6 DRVA Absolute positional positioning....177
5.2.7 DRVI Relative Positioning....182
5.2.8 ZRN Origin Return....188
5.2.9 DSZRDOG Search Origin Return (Under Development)....191
5.2.10 DVIT Interrupt Location (Not Developed)....197
5.2.11 DPIT Maximum Fixed Length Interrupt Positioning Command....202
5.3 HIGH-SPEED PROCESSING COMMANDS....207
5.3.1 PWM Pulse Width Modulation Output Command....207

CHAPTER 6 ELECTRONIC CAM....207

6.1 INTRODUCTION OF ELECTRONIC CAM (E-CAM)....207
6.2 IMPLEMENTATION OF E-CAM....208
6.2.1 First step: Initial setting....209
6.2.2 Second step: Spindle selection / cam table selection....212

6.2.3 Third step: Start/stop E-CAM 213
6.3 E-CAM KEY POINT MODIFY 217
6.3.1 DCAMWR E-CAM data Modify 217
6.3.2 DCAMRD reads E-CAM data (under development) 218
6.3.3 Full points Modify commands 220
6.3.4 Single Data Modify Command DCAMWR Command 221
6.3.5 Ejector Modify command 222
6.4 FLYING SHEARS 224
6.4.1 Function Description 224
6.4.2 Application examples 228
6.5 CHASING 233
6.5.1 Function Description 233
6.5.2 Application examples 240
6.6 SPECIAL FEATURES 243
6.6.1 Ejector function 243
6.6.2 Calculation of master-slave speed ratio 245
6.6.3 Monitoring of master section position 245
6.6.4 Monitoring of accumulated pulse of spindle 246
6.6.5 Setting the direction of electronic cam 246
6.6.6 Encoder frequency monitoring 246
6.6.7 Calculation of spindle position (Developing) 247
6.6.8 Calculation of slave shaft position 248
6.6.9 Probe 249
6.6.10 Motion superposition 250
6.6.11 Control cycle 252
6.7 LIST OF ELECTRONIC CAM RELATED SOFT COMPONENTS 253

CHAPTER 7 INTERRUPT 255

7.1 OVERVIEW 255
7.1.1 Overview 255
Overview 255
7.1.2 Interrupt type 255
7.2 EXTERNAL INTERRUPTION 256
7.2.1 Overview 256
7.2.2 External interrupt type 256
7.2.3 Examples 256
7.3 TIMER INTERRUPT 257
7.3.1 Overview 257
7.3.2 Timer interrupt type 257
7.3.3 Examples 258
7.4 PULSE COMPLETION INTERRUPT 258
7.4.1 Overview 258
7.4.2 Pulse completion interrupt type 258
7.4.3 Example 259

CHAPTER 8 COMMUNICATION 260

8.1 OUTLINE **260**
8.2 INTRODUCTION **260**
8.3 COMMUNICATION PROTOCOL SETUP INSTRUCTION **261**
8.3.1 COM0 protocol configuration 261
8.3.2 COM2 protocol configuration 264
8.3.3 Serial port communication format 265
8.3.4 Serial port communication format soft component list 265
8.3.5 List of communication error codes 266
8.4 HMI MONITORING PROTOCOL **267**
8.5 MODBUS PROTOCOL **268**
8.5.1 MODBUS protocol specification 268
8.5.2 MODBUS function code and data addressing 269
8.5.3 MODBUS Mailing address 274
8.5.4 MODBUS configuration instructions 275
8.5.5 MODBUS command instructions 282
8.6 CANOPEN COMMUNICATION **287**
8.6.1 Overview 287
8.6.2 Hardware configuration 288
8.6.3 Creat CANopen configuration 290
8.6.4 Master station configuration 292
8.6.5 Slave station configuration 295
8.6.6 CANopen configuration download 301
8.6.7Online debugging 301
8.6.8CAN bus monitoring 302
8.6.9CANopen Communication fault code and elimination 304
8.6.10 CANopen Communication variable 306
8.6.11 V5-CANopen Control SD700 Servo Drive 307

CHAPTER 9 SUBPROGRAM **314**

9.1 SUMMARY **314**
9.1.1 V5 subroutine overview 314
9.1.2V5 Subroutine execution mechanism 314
9.2 GENERAL SUBROUTINES APPLICATION **316**
9.2.1 Creating a subroutine 316
9.2.2 Export subroutine 316
9.2.3 Import subroutine 317
9.2.4 Subroutine property 318
9.2.5 Subroutine call 319
9.3INTERRUPT SUBROUTINE APPLICATION **319**
9.3.1 Interrupt subroutine attribute 319
9.3.2 Interrupt Subroutine call 321

APPENDIX I SPECIAL DEVICE ASSIGNMENT INSTRUCTIONS **322**

SM FLAG BIT ALLOCATION **322**
SD REGISTER ALLOCATION **324**

M8000 FLAG BIT, D8000 REGISTER ALLOCATION 327

APPENDIX II SYSTEM ERROR CODE DESCRIPTION...................................... 342

SYSTEM ERROR CODE D8061 342
SYSTEM ERROR CODE D8062 343
SYSTEM ERROR CODE D8063 344
SYSTEM ERROR CODE D8064 346
SYSTEM ERROR CODE D8065 346
SYSTEM ERROR CODE D8066 346
SYSTEM ERROR CODE D8067 346

Chapter 1 Preface

First of all, thank you for purchasing the V5 series programmable controller! The V5 Series Programmable Controller is a new product developed in conjunction with the world's leading control algorithms. This product supports jog movement, linear interpolation, arc, 3D arc, electronic gear / cam and other functions; embedded a large number of convenient industrial function blocks, such as chasing, flying shears, air defense and other functional blocks, making the application very Simple; integrated synchronous follow-up, full-closed control, multi-axis coordinated motion control and other functional blocks can be applied only by calling.

Project	V5-MC104
Program capacity	64K
Power down storage capacity	40K word
Basic instruction speed	100ns
Interpolation cycle	125us~1ms
Number of axes	4+2 axis [1]
High speed input	4M (4 channels) [2], 200k (2 channels)
High speed output	3M (4 channels) [3], 200k (2 channels)
General purpose input/output	22-point digital input, 14-point digital output
Programmable	Ladder diagram, cam command, MC command, G code
Planning mode	T/S type, symmetrical/asymmetric
Sport mode	Constant speed, dynamic position change, speed change, acceleration change, superposition motion
Interpolation	6-axis linear interpolation, circular interpolation, 3D circular interpolation, helical interpolation
Continuous track	Continuous interpolation, shifting, pause
Electronic cam	6-axis electronic cam, chasing, flying shears, electronic gear, ejector
Probe	3, 5us response time
Analog input	2-channel analog input
Communication	RS485 (2), RS422 (1), USB, Ethernet, CAN [4]
Scalability	Maximum support for expansion of 6 digital input and output local expansion modules (16 in 16 out)

[1] Scalable.

[2] Defined as a 4-axis input, as a handwheel input or feedback input, accepting differential or single-ended inputs.

[3] is defined as 4-axis output, each axis includes 2 sets of differential outputs, which can be used as AB phase output, CW/CCW output mode or pulse plus direction mode.

[4] CAN supports CANopenDS301 and CANopenDS402 master and slave protocols.

☞ Safety Precautions:

Please use this product according to the normal steps by professional operators. Pay attention to the following safety-related precautions during use, otherwise it may cause harm to the human body or property damage. The safety precautions are defined as follows.

⚠ Danger: If you make a mistake, it is very likely to cause death or serious injury;

⚠ Warning: Failure to do so may result in death or serious injury;

⚠ Note: If you make a mistake, it may cause moderate injury or minor injury, or it may cause damage to the equipment.

Design considerations	
⚠ Danger	Install a safety loop outside the controller to ensure that the entire system is operating in a safe state in the event of an external power supply failure or controller failure. Malfunctions or incorrect output may cause an accident. Be sure to install an emergency stop circuit, a protection circuit, an interlock circuit that prevents simultaneous reverse rotation, and other interlocking circuits that prevent mechanical damage, such as the upper and lower limits. When the controller CPU detects an abnormality by a self-diagnosis function such as a watchdog timer error, all outputs are turned off. Further, when the controller CPU cannot detect an abnormality such as the input/output control portion or the like, the output control may be invalid. In this case, design the external circuit and structure to ensure that the machine is operating safely. Due to the failure of the relay, transistor, thyristor, etc. of the output unit, the output may be always turned on or off. To ensure that the machine is operating in a safe state, design external circuits and structures for output signals that can cause major accidents.
⚠Note	Do not bundle the control line with the main circuit or power line, or close to the wiring. In principle, please leave more than 100mm, otherwise it will cause malfunction due to noise. When using, please make sure that the connector connected to the peripheral device is not subjected to external force, otherwise it will cause disconnection and malfunction.

Installation Precautions	
⚠ Danger	When performing the installation operation, be sure to disconnect all power sources externally before operating, otherwise there is a risk of electric shock.
⚠Note	Please use it in the general specifications described in this manual. Do not use in places where there is dust, oil smoke, conductive dust, corrosive gas, flammable gas, or exposed to high temperatures, condensation, wind and rain, and places with vibration and shock. Failure to do so may result in electric shock, fire, malfunction, product damage, and aging. Do not touch the conductive parts of the product directly, otherwise it may cause malfunction or malfunction. When processing and wiring work, please do not drop the chips and wire scraps into the vent hole of the controller. Otherwise, it may cause fire, malfunction and malfunction. When installing the product, please use the DIN rail and install the product on a flat surface. The connection cable for peripheral device connection, input/output, etc., should be securely attached to the specified connector to avoid malfunction due to poor contact. The local expansion module must ensure that the locks on both sides are locked, otherwise it may cause malfunction due to poor contact.

Wiring precautions	
⚠ Danger	When wiring operation, be sure to disconnect all power sources before operation, otherwise there is danger of electric shock and product damage.
⚠ Note	When processing and wiring work, please do not drop the chips and wire scraps into the vent hole of the controller. Otherwise, it may cause fire, malfunction and malfunction. When wiring the European terminal type, please follow the precautions below. Otherwise, it may cause electric shock, malfunction, short circuit, disconnection, malfunction or damage to the product. The end of the stranded wire should be twisted so that no wire is diverged. Do not tin on the end of the wire; Do not connect wires that do not meet the specified size or wires that exceed the specified number.

Start maintenance	
⚠ Danger	Do not touch the terminals while power is on. Be sure to clean and plug in the terminals after disconnecting all external power supplies. Otherwise, there is a risk of electric shock and may cause malfunction. Before changing the program, performing forced output, RUN, STOP, etc. during operation, be sure to read the manual first, and operate under the condition that the safety is fully confirmed. Otherwise, mechanical damage and accident may occur. Do not change the program in the controller from multiple peripheral devices at the same time. Otherwise, the controller's program may be damaged and cause malfunction.
⚠ Note	Do not disassemble or modify the product without authorization, otherwise it may cause malfunction, malfunction or fire. When disassembling the connecting cable such as the extension cable, please operate it after disconnecting the power supply, otherwise it may cause malfunction or malfunction.

Disused and transported	
⚠ Note	When disposing of the product, please dispose of it as industrial waste. When disposing of the battery, dispose of it separately in accordance with the laws and regulations specified by each region. The controller is a precision device, so avoid it from being subjected to the impact of general specifications during transportation. Failure to do so may result in a controller failure. After transportation, please confirm the operation of the controller.

Chapter 2 Summary

2.1 Introduction to Programmable Controller

Programmable Logic Controller (PLC) is an electronic system for digital computing operations designed for industrial applications. It mainly reads the state of external input signals such as buttons, sensors, switches and pulses, and executes the logic, sequence, and microprocessor according to the state or value of these input signals and according to the pre-written program stored internally. Timing, counting and arithmetic operations produce corresponding output signals such as relay switches and control of mechanical equipment operations. The program and monitoring device status can be easily edited/modified by a computer or program writer to perform on-site program maintenance and test machine adjustment.

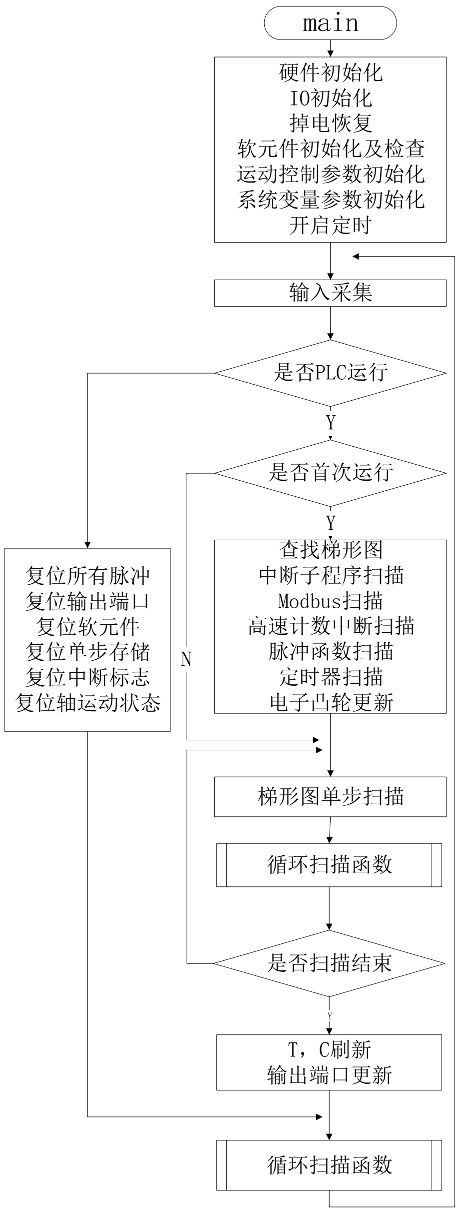
2.2 Basic control principle

2.2.1 How the programmable controller works

The programmable controller adopts the cyclic scan mode, including input point scanning, user program execution, output point refresh, internal processing and communication processing.

Before running the programmable controller, you can use the programming software to write the control logic between the input point and the output point and download it to the programmable controller. During the running of the programmable controller, the input point signal is scanned first and read. Take the programmable controller, and then complete the operation and logic processing according to the control program. The operation and logic processing result will change the value of the output point, and finally convert the value in the output point into the electrical signal output and control the operation of various mechanical devices.

In the running process of the programmable controller, the working mode of the cyclic scan is adopted, and the purpose of receiving the control and operating the device is achieved by repeatedly performing the input point scanning, the user program execution, and the output point refreshing work.



2.2.2 User program control principle

In the user program, the input point of the programmable controller is called the contact, and its function is the same as the switch contact in the industrial equipment, which means that the energy flow is turned on or off. In the programmable controller, the input point is stored as a device. When the input point is high, the corresponding device is in the on state, participating in the logic operation in the user program and affecting the value of the output point; the output point is called The coil represents the conduction or turn-off of the output energy flow, and the value of the corresponding device of the output point is determined by the input point and the calculation result of the control logic. When the output is refreshed, the value of the device is converted to the output of the transistor or relay of the electrical signal at the output point, thereby completing the control of the device.

2.3 programming software

The V5 series motion controller is programmed using VCAutoDesignsoft software.

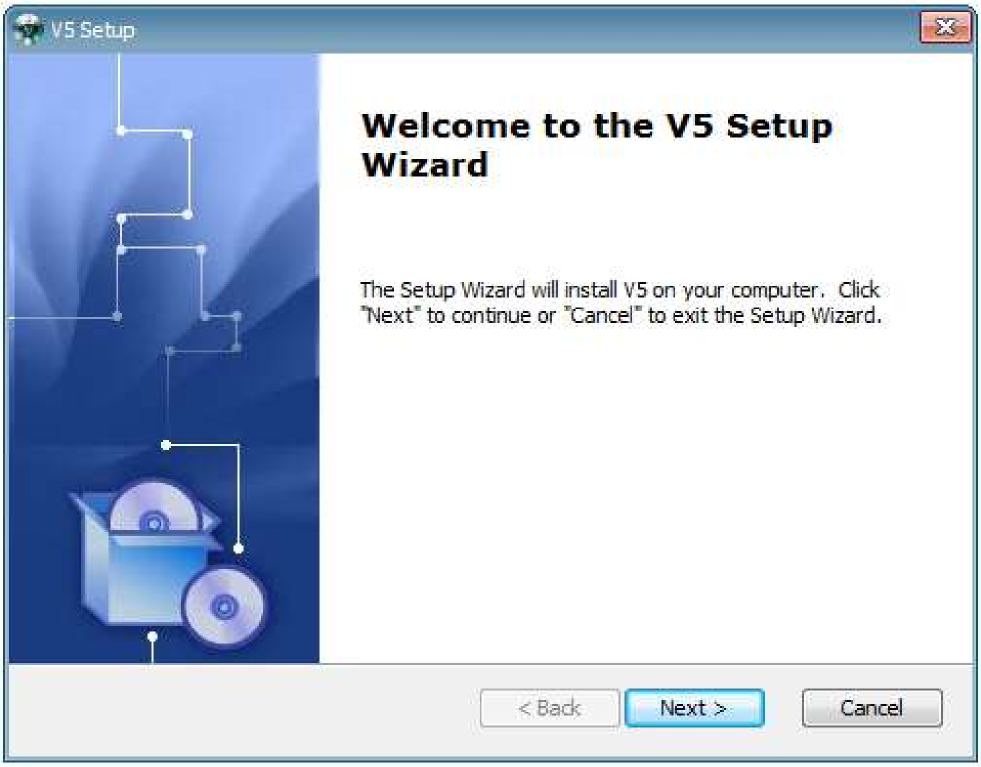
■ Operating environment requirements:

System Configuration	**Claim**
operating system	WindowsXP、Windows2000、Windows7、Windows8、Windows10
System type	32-bit, 64-bit
CPU	Requires 600 megahertz (MHz) Pentium III processor and above
RAM	Minimum RAM requirement 1G
hard disk	Need more than 1G of free space
Graphics card	SuperVGA (800x600) or higher resolution display
drive	USB-SC09-FX communication line driver
other demands	Install Microsoft Office 2003 and above

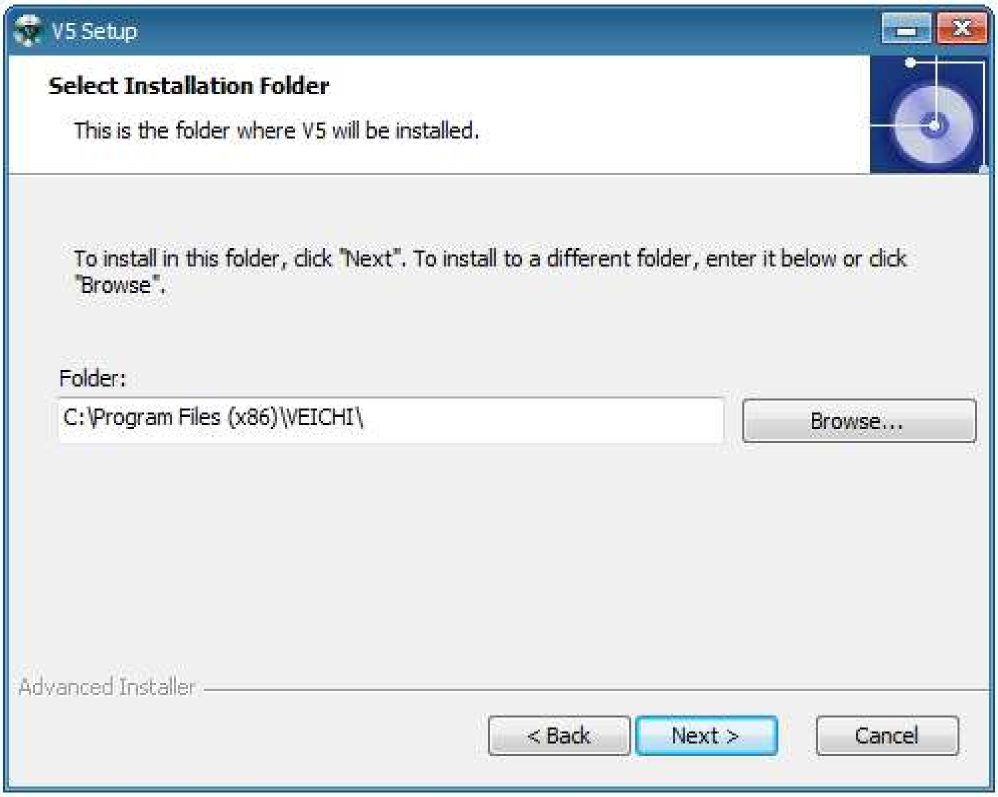
■ VCAutoDesignsoft software installation

1, VEICHI Flextronics official website to download the latest version of the software installation packagehttps://www.veichi.com/, download the installation package, as shown below:

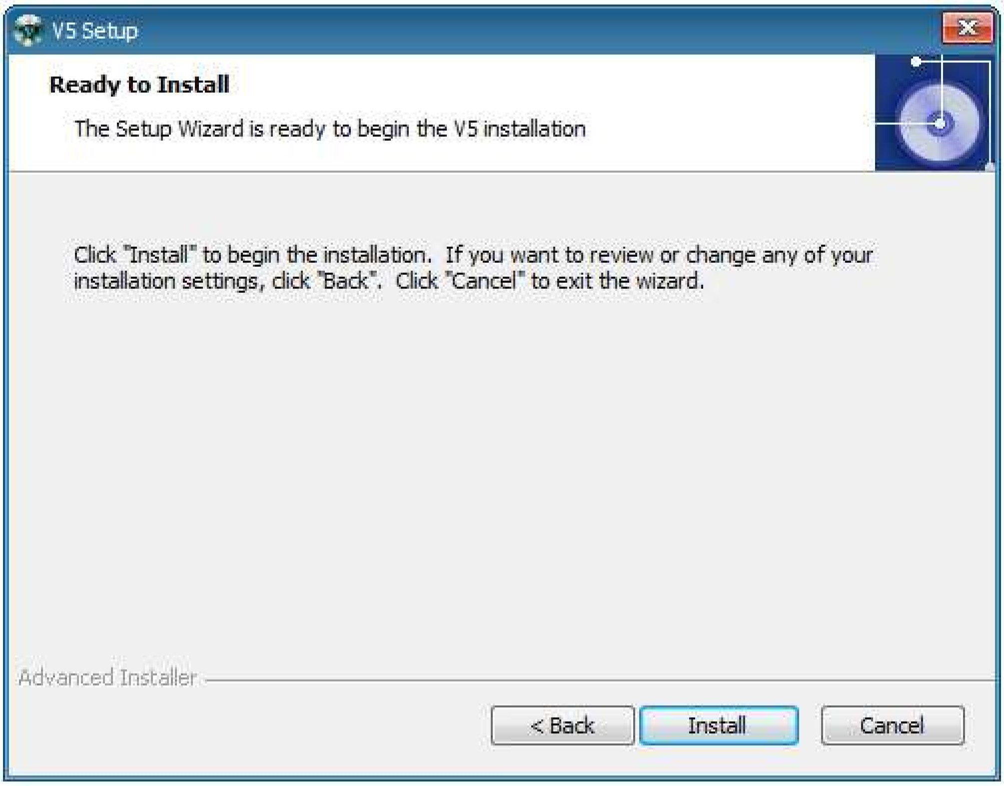
2. Double-click the installation package with the left mouse button to enter the installation interface. As shown in the figure below, click the “Next” button to enter the next step;

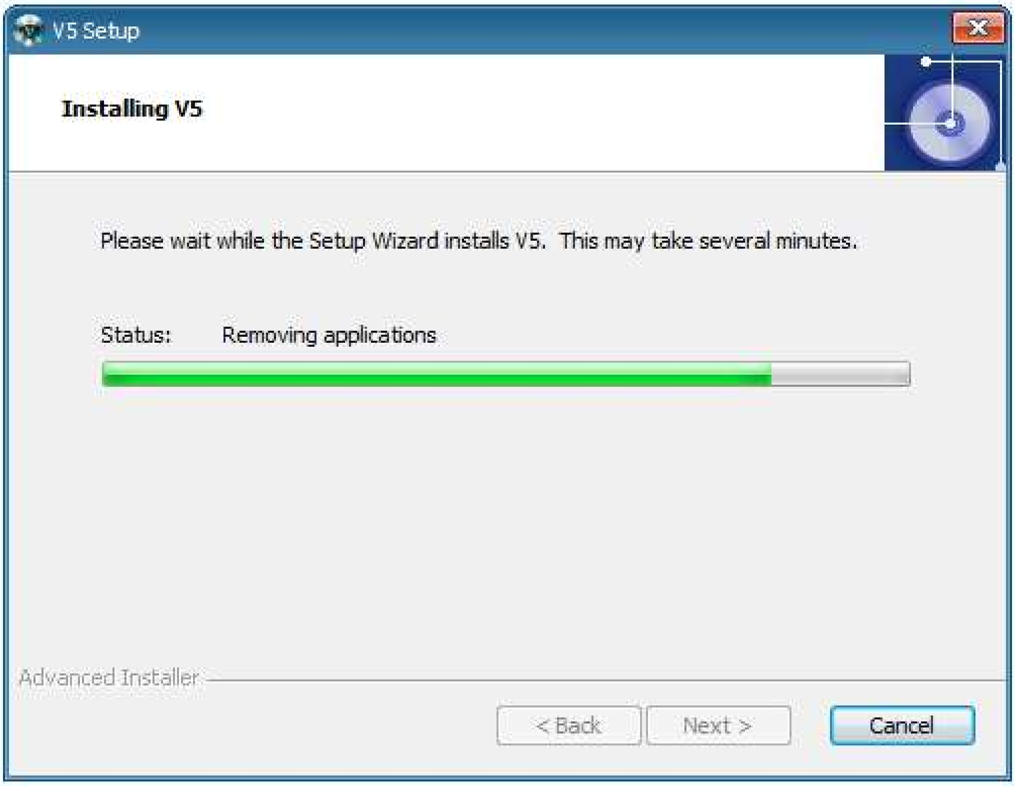


3. Select the software installation location and click the “Next” button to enter the next step, as shown below:

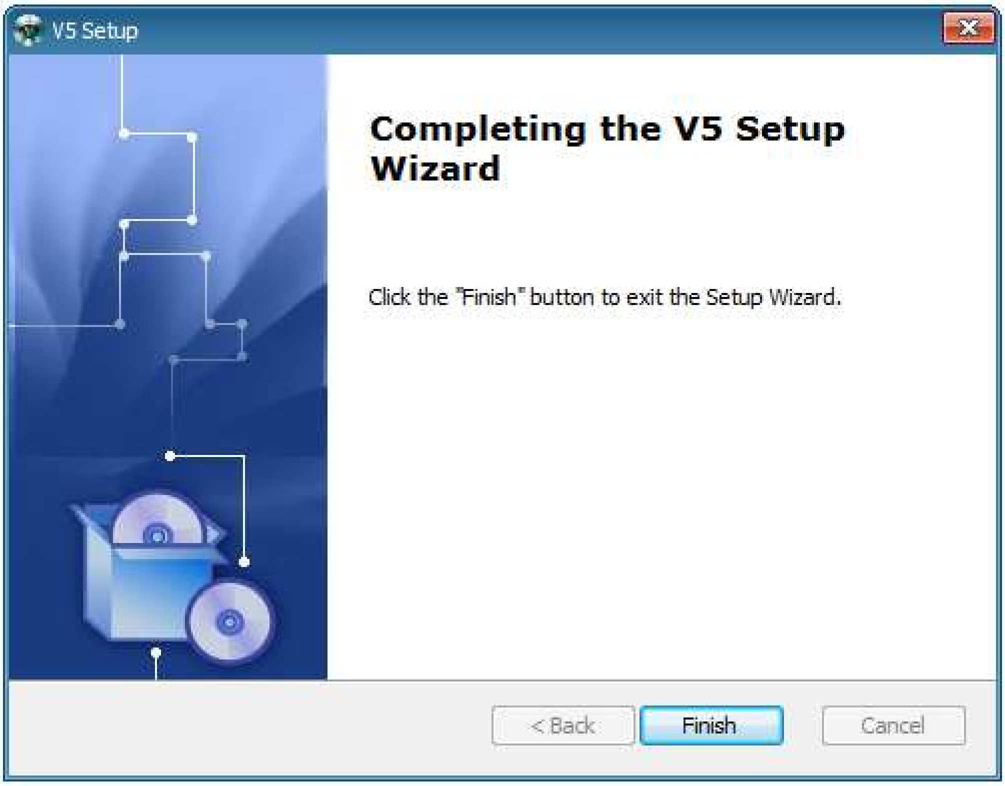


4. Click the “Install” button to start installing the software and wait for the installation to complete, as shown below:





5、After the installation is completed, the interface shown below appears, click the “Finish” button;



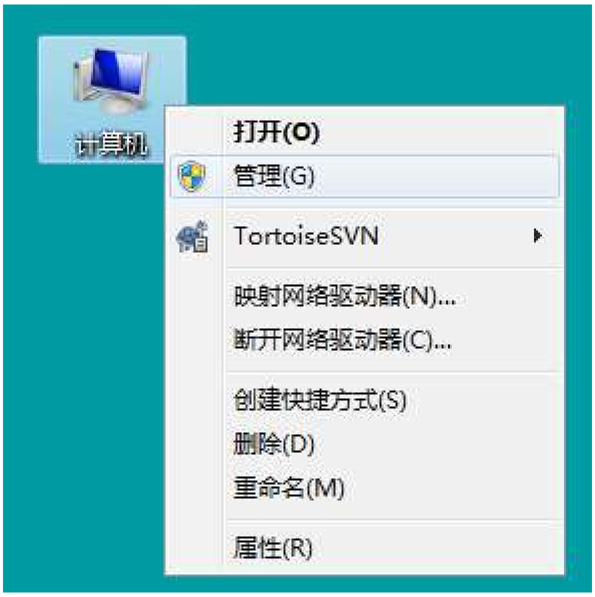
6、Find the auto-generated shortcut icon on the desktop, double-click to open the software and start programming.

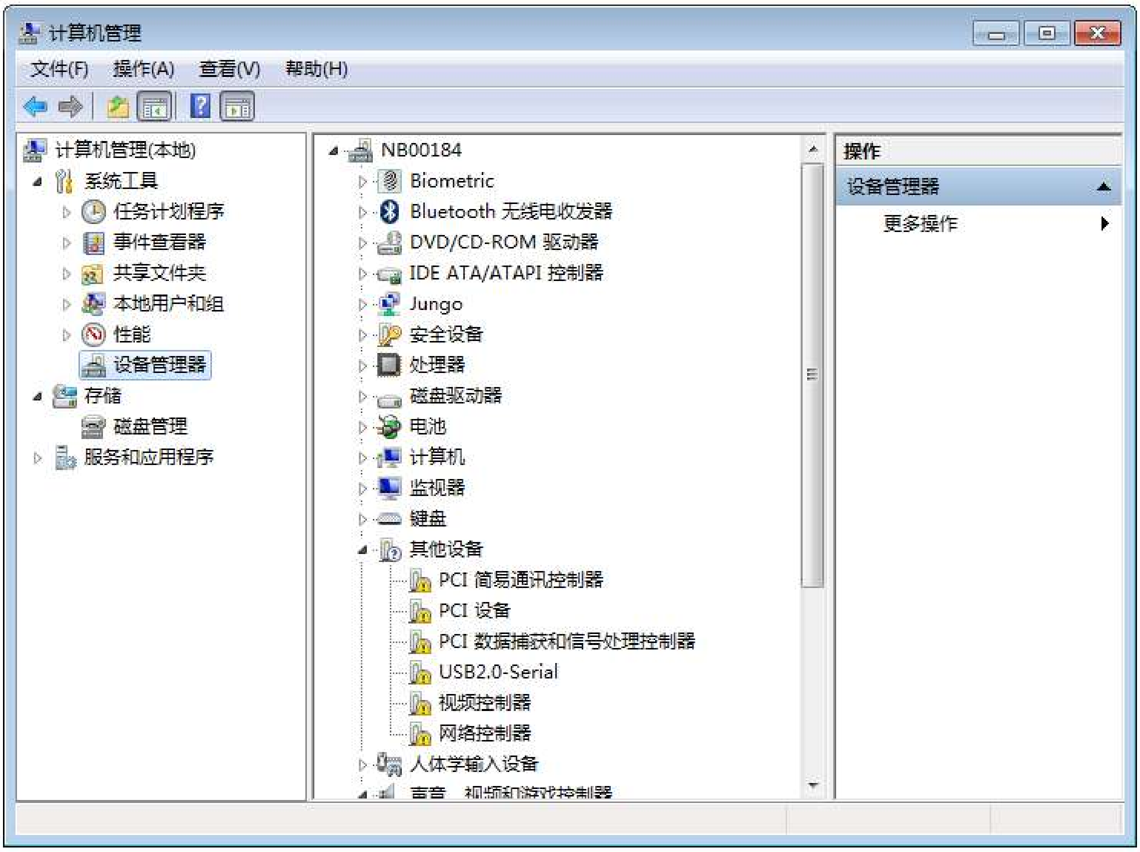
- USB-SC09-FX communication line driver installation

1. The driver of different communication card manufacturers is different. Take the USB-SC09-FX communication line as an example to obtain the driver installation file of the communication line, as shown below:

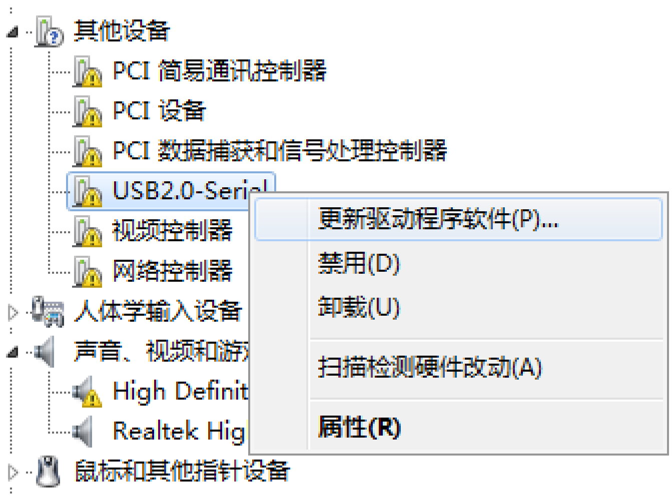
2. Connect the USB-SC09-FX communication line to the USB interface of the background software computer and the RS422 interface of the V5-MC104 controller.

3. Enter the device manager of the computer; take WIN7 as an example (WindowXP is "My Computer" -> "Properties / Device Manager"), select "Computer" from the mouse, right click, select "Manage", click "Device" Manager", as shown below:

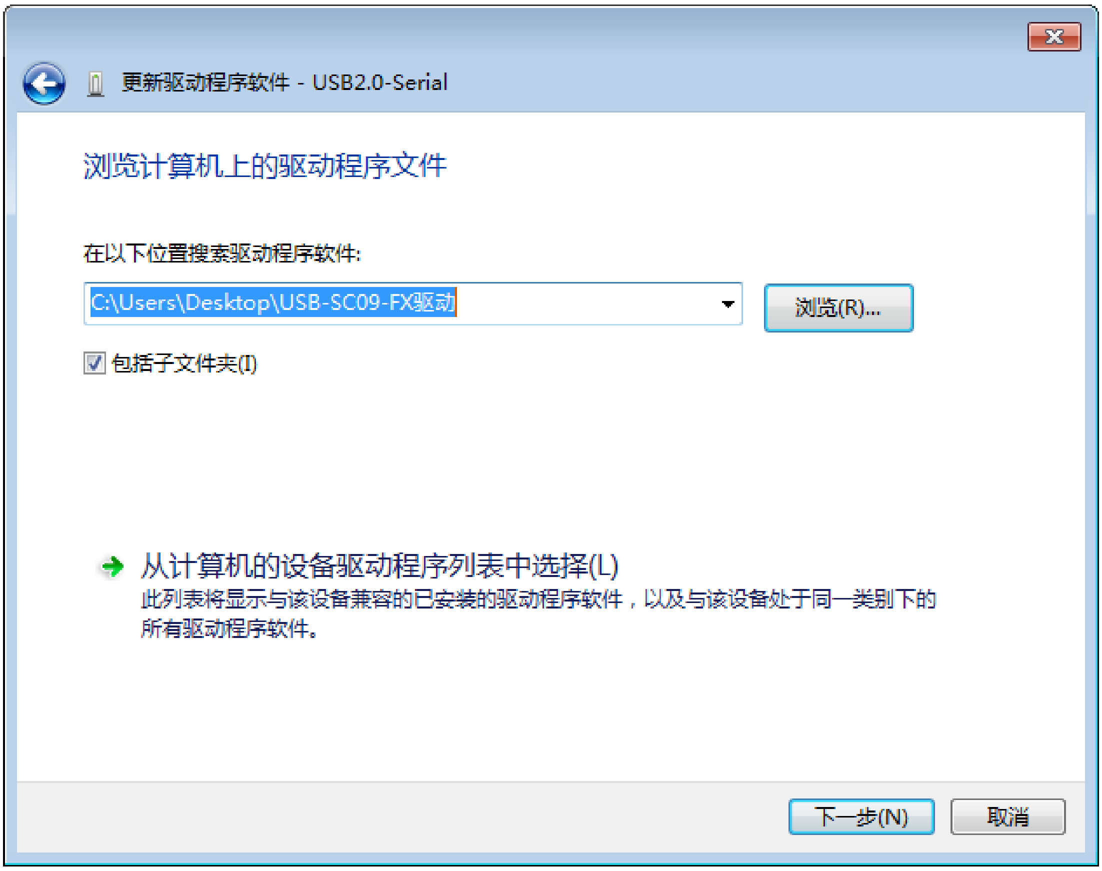

计算机
打开(O)
管理(G)
TortoiseSVN
映射网络驱动器(N)...
断开网络驱动器(C)...
创建快捷方式(S)
删除(D)
重命名(M)
属性(R)


计算机管理
文件(F) 操作(A) 查看(V) 帮助(H)
计算机管理(本地)
系统工具
任务计划程序
事件查看器
共享文件夹
本地用户和组
性能
设备管理器
存储
磁盘管理
服务和应用程序
NB00184
Biometric
Bluetooth 无线电收发器
DVD/CD-ROM 驱动器
IDE ATA/ATAPI 控制器
Jungo
安全设备
处理器
磁盘驱动器
电池
计算机
监视器
键盘
其他设备
PCI 简易通讯控制器
PCI 设备
PCI 数据捕获和信号处理控制器
USB2.0-Serial
视频控制器
网络控制器
人体学输入设备
操作
设备管理器
更多操作

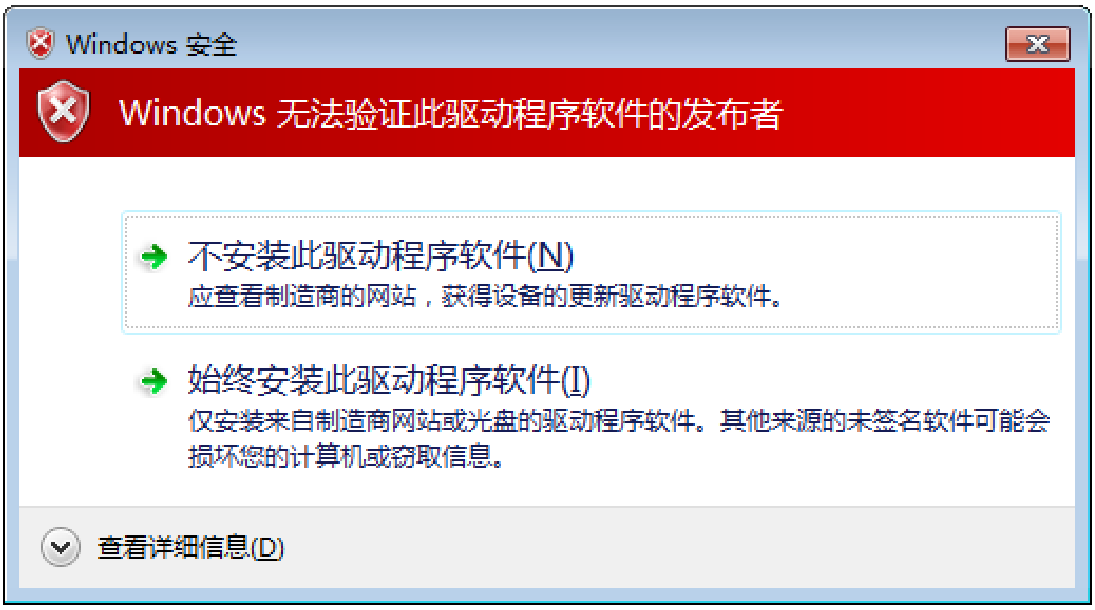
4. Select "USB2.0-Serial" under "Other Devices", right click and click "Update Driver Software":



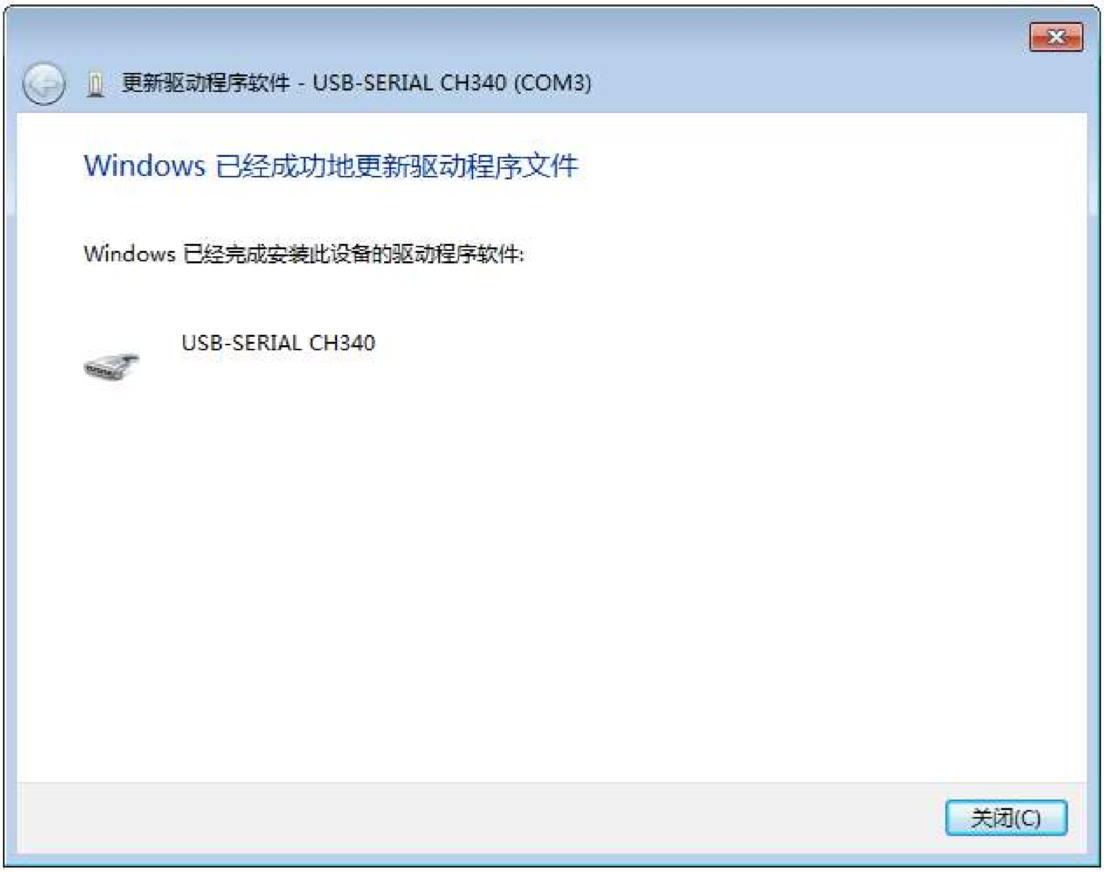
5. Click “Browse my computer to find the driver software”, click the “Browse” button, select the folder where your driver is located, such as the driver folder of the USB-SC09-FX communication line “C:\Users\Desktop\USB” -SC09-FX driver", as shown below:

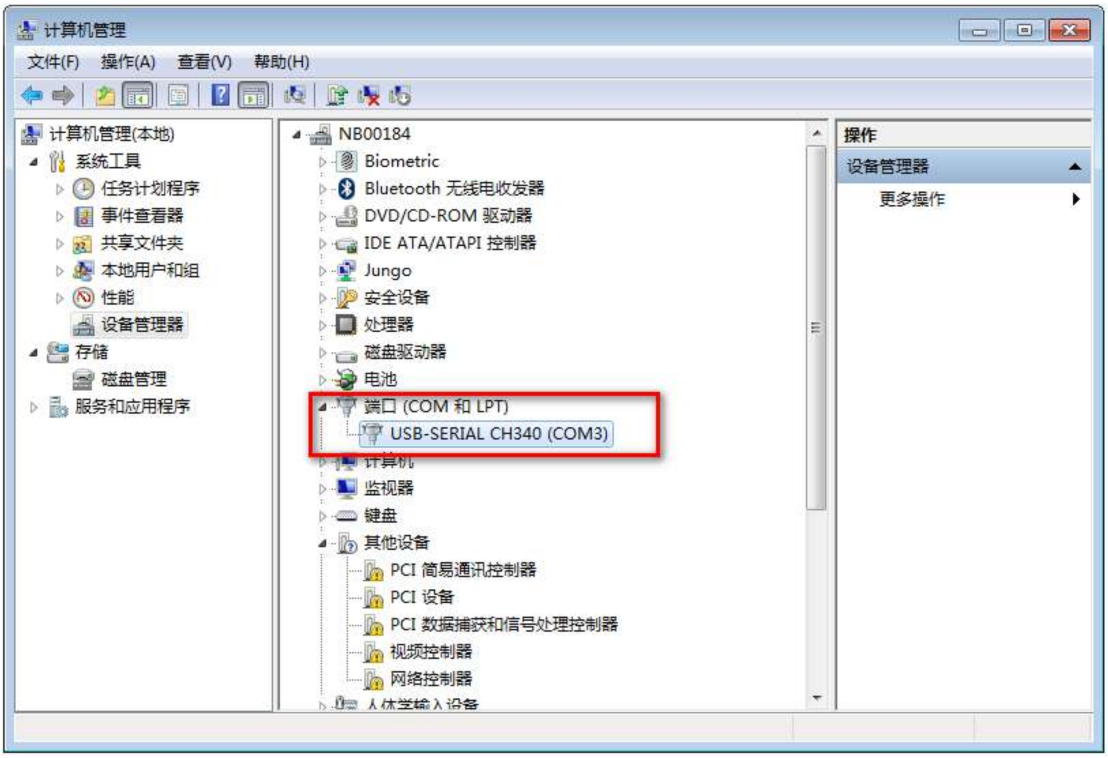


6. Click the “Next” button and follow the prompts to install. If the following warning appears, click “Always install this driver software”:



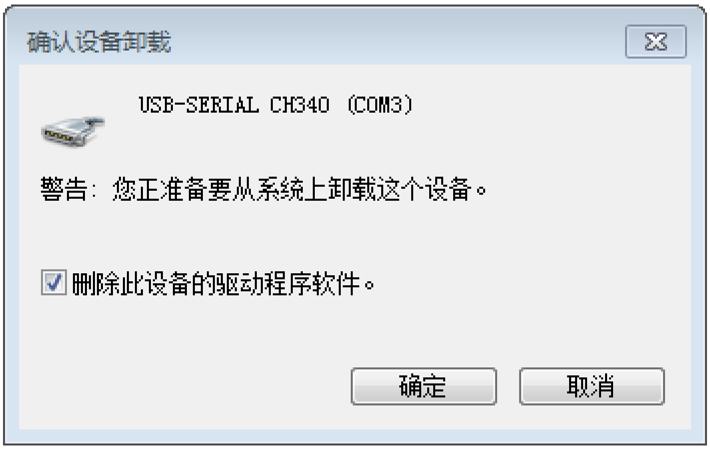
7, after installing the driver, will prompt "Windows has completed the installation of the driver software for this device", click to close; in the device manager "port (COM and LPT)" will appear a new serial port, as follows In the red box of the figure, "COM3", this serial port is the serial port number used by the USB-SC09-FX communication line. The COM port will appear each time you plug in and use the communication line. You only need to select the COM port in the programming software to communicate.





8, the driver uninstall

The driver is uninstalled in order to release the COM port resources for use by other devices, or the driver needs to be uninstalled and reinstalled when the device fails. Follow the steps below to uninstall the driver: Open the device manager in the "Port (COM and LPT)" node. Select the COM port to be uninstalled, select “Uninstall” from the right mouse button menu, select “Delete driver software for this device”, and click “OK” button to complete the uninstall.



Chapter 3 soft components

The system device types are as follows:

Serial No.	Component type	Function and classification
1	Input and output relay	Bit element X corresponding to the hardware switch input of the PLC
		Bit element Y corresponding to the control output of the PLC
2	Auxiliary relay	Ordinary auxiliary relay M
		System special auxiliary relay M
		System special auxiliary relay SM
3	State relay	Step control status flag bit element S
4	Timer	16-bit timer T of 1ms, 10ms, and 100ms steps
5	counter	16bit/32bit increase/decrease counter C
		32bit high speed counter C
6	Data register	Normal data register D
		System special purpose data register D
		System special purpose data register SD
		Data Indirect Addressing Register V, Z
7	File register	File register R
8	Label	Label/jump pointer P
9	Subroutine	Subroutine SBR
		Interrupt subroutine I
10	Nested pointer	Nested pointer N
11	constant	Decimal constant K
		Hexadecimal constant H
		Floating point number E

3.1 list of soft components

Input and output relay			
Input relay	X0～X377	256 points	The device number is octal number, and the input and output totals 512 points.
Output relay	Y0～Y377	256 points	
Auxiliary relay			
General use	M0～M499	500 points	Does not support power-down save
Keep in use	M500～M1023	524 points	Power down save
Keep in use	M1024～M7679	6656 points	Power down save
System special use	M8000～M8511	512 points	Power down save
System special use	SM0～SM1023	1024 points	Power down save
State relay			
Initialization state	S0～S9	10 points	Does not support power-down save
General use	S10～S499	490 points	
Keep in use	S500～S899	400 points	Power down save
Alarm	S900～S999	100 points	Power down save
Keep in use	S1000～S4095	3096 points	Power down save
Timer			
100ms	T0～T191	192points	0.1 to 3, 276.7 seconds
100ms	T192～T199	8points	0.1 to 3, 276.7 seconds, subroutine, interrupt subroutine
10ms	T200～T245	46points	0.01 to 327.67 seconds
1ms cumulative type	T246～T249	4points	0.001 to 32.767 seconds
100ms cumulative type	T250～T255	6points	0.1 to 3, 276.7 seconds
1ms	T256～T511	256points	0.001 to 32.767 seconds
Counter			
General use up count (16 bit)	C0～C99	100points	0 to 32,767, does not support power-down save
Keep up counting (16 bit)	C100～C199	100points	0~32,767, Power down save
Generally used in both directions (32 bit)	C200～C219	20points	-2,147,483,648~+2,147,483,647, does not support power-down save
Keep in both directions (32 bit)	C220～C234	15points	-2,147,483,648~+2,147,483,647, Power down save
High speed counter			
Single phase single count input bidirectional (32 bit)	C235～C245	11points	-2,147,483,648~+2,147,483,647, Power down save
Single-phase double counting input bidirectional (32-bit)	C246～C250	5points	
Dual phase double counting input bidirectional (32 bit)	C251～C255	5points	

Data register			
Generally used (16 digits)	D0～D199	200points	Does not support power-down save
Keep in use (16 bit)	D200～D511	312points	Power down save
Keep in use (16 bit)	D512～D7999	7488points	Power down save
Special use (16 digits)	D8000～D8511	512points	Power down save
Special use (16 digits)	SD0～SD1023	1024points	Power down save
Indexing (16 bit)	V0～V7，Z0～Z7	16points	Power down save
File register			
Extended register (16 bits)	R0～R32767	32768points	Power down save
Label			
CJ instruction	P0～P511	512points	Used in conjunction with the LBL instruction
Subroutine			
CALL instruction	/	512points	Can be set as normal subroutine, encryption subroutine, subroutine with parameters, subroutine with parameter encryption
Input interrupt X000~X007	I00□，I10□ I20□，I30□ I40□，I50□ I56□，I57□	8points	□Indicates: 0 falling edge interrupt, 1 rising edge interrupt. After the interrupt disable flag register is turned ON, the corresponding input interrupt is disabled.
Timed interrupt	I6□□～I8□□	3points	□□=01～99，Time base = 1ms
Count completion interrupt	I010～I080	8points	DHSCS instruction
Pulse completion interrupt	I502～I506	5points	
Motion control subroutine	MC00～MC63	64 points	
Nested pointer			
Main control circuit	N0～N7	8points	MC instruction
constant			
Decimal constant K	16 bits	-32,768～+32,767	
	32 bits	-2,147,483,648～+2,147,483,647	
Hexadecimal constant H	16 bits	0～FFFF	
	32 bits	0～FFFFFFFF	
Real number E	32 bits	-1.0*2e128～-1.0*2e-126， 1.0*2e-126～1.0*2e128(32Bit)	

3.2 Input and output relay

3.2.1 Input Relay X

The input relay X represents the component of the PLC external input signal state, and the external signal state is detected through the X port. 0 represents an external signal open circuit, that is, OFF; 1 represents an external signal closed, that is, ON. The state of the input relay cannot be modified by the program command method, and its contact signal (normally open type, normally closed type) can be used indefinitely in the user program.

The input relay X is numbered in octal, which is X0, X1...X7, X10, X11...X25. After accessing the local expansion module, the number of the X port on the expansion module is numbered sequentially next to the X port on the main module, but the number of the expansion module always starts from the octal digit to 0.

3.2.1 Output Relay Y

The output relay is directly connected to the hardware of the external control port and logically corresponds to the physical output port of the PLC. After each scan of the user program, the PLC will transfer the component status of the Y relay to the hardware port of the PLC. 0 means the output port is open, that is, the output port is OFF; 1 means the output port is closed, that is, the output port is ON. Y relay components can be used indefinitely in the user program. In hardware, depending on the output components, it can be divided into relay type, transistor type, and the like.

The output relay Y is numbered in octal, and is Y0, Y1...Y7, Y10, Y11...Y15. After accessing the local expansion module, the number of the Y port on the expansion module is numbered sequentially next to the Y port on the main module, but the number of the expansion module always starts from the octal digit to 0.

3.3 Auxiliary relay M

The auxiliary relay M component is used as an intermediate variable in the execution of the user program. Like the auxiliary relay in the actual electronic control system, it is used for the transmission of status information. There is no direct connection with the external port, but X can be copied to the M through the program statement., or the way M is copied to Y to contact the outside world. Multiple M variables can be used as word variables, and one M variable can be used indefinitely.

General	Use for keep	Use for keep	Special relay	Special relay
M0~M499 500 bits[1]	M500~M1023 524 bits [2]	M1024~M7679 6656 bits [3]	M8000~M8511 512 bits	SM0~SM1023 1024 bits

[1] Non-blackout holding area. Use the parameter setting to change to the power failure holding area.

[2] Power outage retention area. Use the parameter setting to change to the non-power-off holding area.

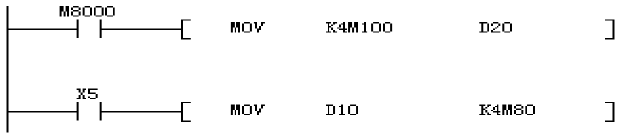
[3] Power failure holding area cannot be changed by parameters.

The auxiliary relay M is numbered in decimal mode, and the variable above M8000 is a system-specific variable for the interaction between the PLC user program and the system state; some M variables also have a power-down save feature.

The special auxiliary relay SM is a system-specific variable, numbered in decimal, for the interaction between the PLC user program and the system state.

There are a large number of special auxiliary relays in the programmable controller, each of which has its own specific function. Please see the appendix for specific functions. Note that the user cannot use special auxiliary relays that have not been defined.

Continuous M variables can be accessed in bytes or words, for example:



Among them, K4M100 means that 16 units of M100, M101, M102...M115 are combined to form a unit of one word for reading operation (M100 is used as bit 0 of the word... M115 is used as bit 15 of the word), which can improve programming efficiency.

3.4 Status Relay S

The state relay S is used for the design and execution processing of the step program, and uses the STL step instruction to control the shift of the step state S, simplifying the programming design.

If STL programming is not used, S can be used as a normal bit component, just like the M variable. The state S variable is identified by symbols such as S0, S1 ... S999, and its serial number is numbered in decimal. Some S variables have a power-down save function.

General use		Use for keep	Use for warn	Use for keep
S0~S9 10 points[1]	S10~S499 490 points [1]	S500~S899 400 points [2]	S900~S999 100 points	S1000~S4095 3096 points [3]

[1] Non-blackout holding area. Use the parameter setting to change to the power failure holding area.

[2] Power outage retention area. Use the parameter setting to change to the non-power-off holding area.

[3] Power failure holding area cannot be changed with parameters.

3.5 Timer T

Timer function: timing function. The composition of the timer: the coil of the timer, the power-on of the timer, and the value register of the count time are composed of three parts.

The operation principle of the timer: When the coil of the timer is "powered" (the energy flow is valid), the timer starts counting. If the timing value reaches the preset time value, the contact action, a contact (NO contact) is closed, b The contact (NC contact) is disconnected. If the coil is "powered out" (the energy flow is invalid), the contact of the timer returns to the initial state, and the timing value is automatically cleared. Some timers also have the characteristics of accumulation, power-down retention, etc., and maintain the value before power-down after power-on.

Timer number: numbered in decimal, identified by symbols such as T0, T1...T10, T11....

The length of the timer: there are 1ms, 10ms, 100ms, etc., and some have power-down retention characteristics.

100ms		10ms	1msCumulative type	100msCumulative type	1ms
T0~T191 192 points[1]	T192~T199 8 points [2]	T200~T245 46 points [1]	T246~T249 4 points [3]	T250~T255 6 points [4]	T256~T511 256 points [1]

[1] Non-power-off holding area.

[2] Non-power-off holding area, subroutine, interrupt subroutine.

[3] Power failure holding area, cumulative type.

[4] Power failure holding area, cumulative type.

Timer setting value: The constant (K) in the program memory is used as the setting value, and can also be specified indirectly by the contents of the data register (D).

When the data register D is used as the set value, the content of D must be set before starting the timing. When the counting starts, the data change of D can only take effect when the next start timing.

There is no timer number used as a timer, and it can also be used as a data register for value storage.

When the accumulated timer reaches the set time, the output contact can only be operated when the coil command or END command is executed.

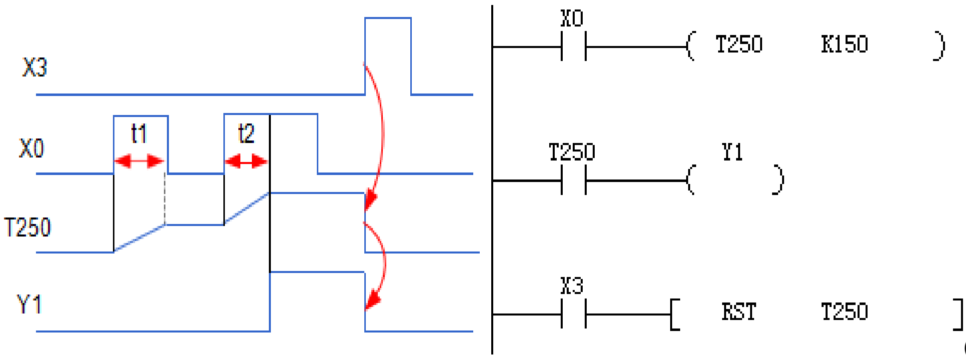
From the start of the coil that drives the timer to the contact action of the timer, the possible timing lengths are as follows:

(1) The longest case is (T+T0+a), where: T is the set timing time; T0 is the program scan execution time; a is the timer's timing step.

(2) The shortest case is (T-a).

(3) If the contact command of the timer is before the coil command, the least ideal timing length is (T+2T0).

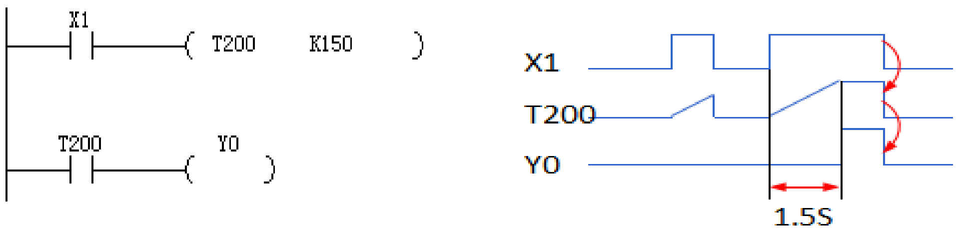
(4) Using the b-contact of the timer, the output signal of the time-delayed disconnection and self-oscillation can be realized.



(5)

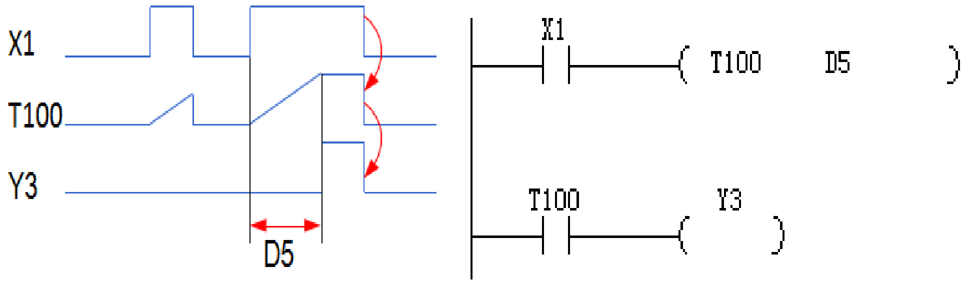
PLC also provides special timer instructions, such as TTMR, STMR, etc., please refer to the description of the corresponding instructions.

【Example 1】 The ordinary timer T200 is a counter of 10ms step, and the actual operation delay is 150×10ms=1500ms, that is, 1.50s. The operation principle is:



【Example 2】For the cumulative timer T250 with power-down hold, the drive signal is OFF, or when the PLC is powered down, the internal count value remains unchanged. When the next drive signal is ON, continue counting until the timing is satisfied. When the set value is reached, the output contact closes. When the timer coil is reset, the timing value is cleared and the output contact is broken, as shown below. Since the counter T250 is 100ms step, the actual action delay is 150×100ms=15000ms, which is 15.0s, which is the (t1+t2) time in the figure:

【Example 3】 The setting operation value of the timer can be set by register D, as shown in the figure below. (In the counter timing process, if the value in register D changes, it will take effect the next time the timer starts.)



3.6 Counter C

Counter function: used to complete the counting function.

The composition of the counter: the coil of the counter, the contact of the counter, and the timing data value register.

The working process of the counter: When the counter coil has a rising edge (from OFF→ON), the counter's count value changes by 1. When the count value reaches the set value, the counter's contact action and the normally open point (NO) closes. , the normally closed point (NC) is disconnected. If the count value is cleared, the input a contact is disconnected and the b contact (NC contact) is closed. Some counters have the characteristics of power-down hold, accumulation, etc., and maintain the value before power-off after power-on.

Counter number: numbered in decimal, identified by C0, C1...C11, etc.

Generally use 16bit	Keep 16bit	Generally use 32bit	Keep 32bit	Keep 32bit
C0~C99 100 point increment [1]	C100~C199 100 point increment [2]	C200~C219 20-point bidirectional counting [1]	C220~C234 15-point bidirectional counting [2]	C235~C255 21-point high-speed counting [2]

[1] Non-blackout holding area. Use the parameter setting to change to the power failure holding area.

[2] Power outage retention area. Use the parameter setting to change to the non-power-off holding area.

Counter No.	Direction switch	Counter No .	Direction switch	Counter No.	Direction switch	Counter No.	Direction switch
C200	M8200	C209	M8209	C218	M8218	C226	M8226
C201	M8201	C210	M8210	C219	M8219	C227	M8227
C202	M8202	C211	M8211	—	—	C228	M8228
C203	M8203	C212	M8212	C220	M8220	C229	M8229
C204	M8204	C213	M8213	C221	M8221	C230	M8230
C205	M8205	C214	M8214	C222	M8222	C231	M8231
C206	M8206	C215	M8215	C223	M8223	C232	M8232
C207	M8207	C216	M8216	C224	M8224	C233	M8233

C208	M8208	C217	M8217	C225	M8225	C234	M8234

For the 32-bit counters C200~C234, the special auxiliary relays M8200~M8234 are used as the up/down counter switching control, as shown in the following table:

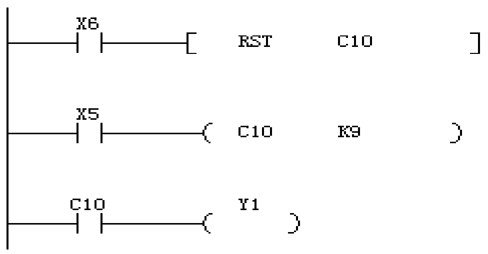
The characteristics of the 16-bit counter and the 32-bit counter are shown in the table below. It can

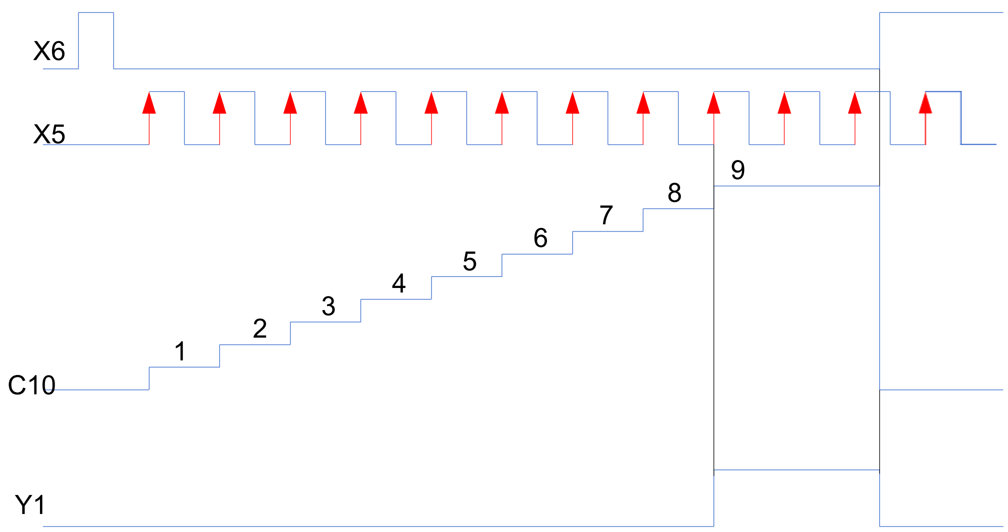
project	16-bit counter	32-bit counter
Counting direction	Count up	Increase or decrease switching usage (see table above)
Set value range	1~32,767	-2,147,483,648~+2,147,483,647
Specified set value type	Constant K or data register	Constant K, also available in 2 D data registers
Current value change	No change after the number	Subsequent change (cycle counter)
Output contact	Keep moving after the number	Keep the action in order, countdown reset
Reset action	When the RST command is executed, the current value of the counter is zero, and the output contact is reset.	
Current value register	16 bits	32 bits

be used separately by the switching of the counting direction and the use condition of the counting range.

3.6.1 16bit counter

For the 16-bit counter, the effective setting values are K1 to K32, 767 (decimal constant); the set values K0 and K1 have the same effect, that is, the output contact operates at the beginning of the first counting. The following example:





The count input X5 drives the C10 coil once, and the current value of the counter increases. When the ninth coil command is executed, the output contact operates. In the future, even if the count input X5 is operated again, the current value of the counter does not change. If reset input X6 is ON, the RST instruction is executed, the current value of the counter is cleared to 0, and the output contact is reset.

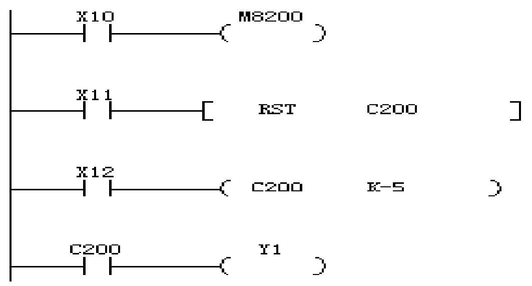
The set value of the counter can be specified by the data register number in addition to the above constant K setting. In the above example, D20 is specified, and if the content of D20 is 9, it is the same as setting K9.

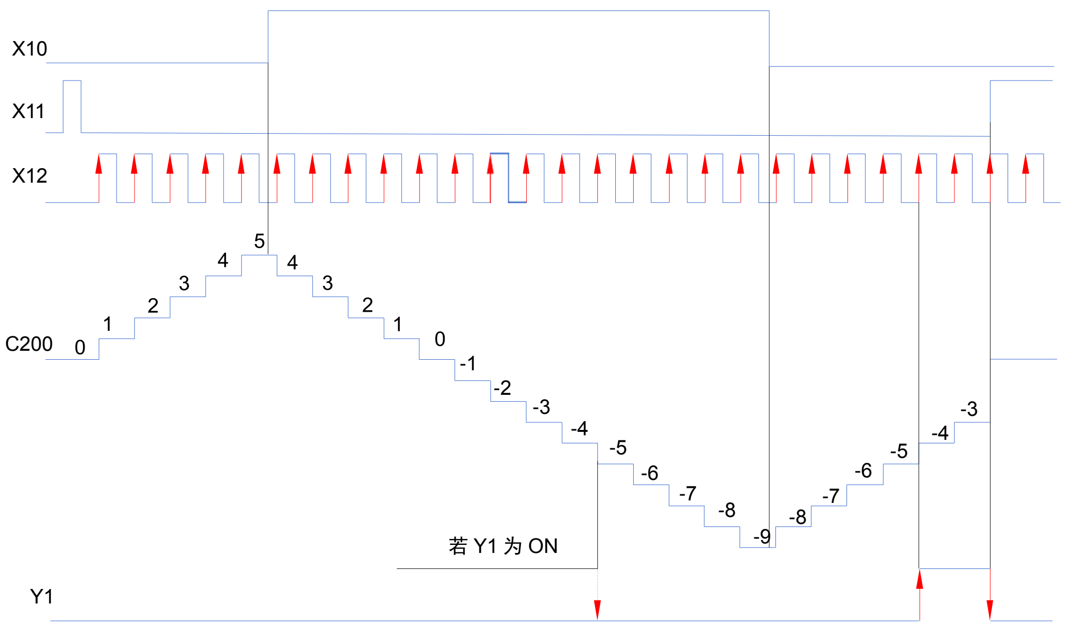
When data of a set value or more is written to the current value register by a command such as MOV, the output coil is turned on at the next input, and the current value register becomes the set value.

For the general counter, if the power of the programmable controller is turned off, the counter value of the counter is cleared, and the counter for power failure hold can store the count value before the power failure, so the counter can be counted up again.

3.6.2 32bit counter

For the 32-bit counter, the set value of the up/down count is valid range -2,147,483,648 to +2,147,483,647 (decimal constant), which can be set by the constant K or the contents of the data register D. Use the special auxiliary relays M8200~M8234 to specify the direction of up/down counting. If C△△△ drive M8△△△ is set to 1, it will count down, and if it is not driven, it will count up.





The increase or decrease of the current value is independent of the action of the output contact, but if it is incremented from 2,147,483,647 and then input one pulse, it becomes -2,147,483,648. Similarly, if you start counting down from -2, 147, 483, 648 and then input a pulse, it becomes 2,147,483,647. (This type of action is called a ring count); if the reset input X11 is ON, the RST instruction is executed, the current value of the counter becomes 0, and the output contact is also reset.

When the counter for power failure hold is used, the current value of the counter, the output contact action, and the reset state are maintained. The 32-bit counter can also be used as a 32-bit data register. However, the 32-bit counter cannot be used as a device in a 16-bit application instruction. When the data of the set value or more is written to the current value data register by the DMOV command or the like, the count can be continued at the time of the subsequent count input, and the contact does not change.

For a 16-bit counter, the highest bit (bit 15) is a sign bit, and the processed data is in the range of 0 to 32767, that is, it can only be a positive number; for a 32-bit counter, the highest bit (bit 31, the highest bit of the high byte) is a sign bit. The processed data range is -2,147,483,648-2,147,483,647.

3.6.3 High number counter

The high-speed counter can count the external input signal, and can realize single-phase single counting, single-phase double counting, and AB phase 1/4 multi-frequency counting.

3.7 Register

The role of the register: for the operation and storage of data. Such as the operation and storage of timers, counters, analog parameters.

The width of the register: 16bit. If a 32-bit instruction is used, the two adjacent registers are automatically composed of 32-bit registers, the lower address is the lower byte, and the higher address is the high byte.

Register type: data register D, indexed data registers V and Z, file register R.

General use	Keep in use	Keep in use	Special use	Indexing	Keep in use	Special use
D0~D199 200 points [1]	D200~D511 312 points [2]	D512~D7999 7488 points [3]	D8000~D8511 512 points	V0~V7 Z0~Z7	R0~R32767 32768 points [3]	SD0~SD1023 1024 points

[1] Non-blackout holding area. Use the parameter setting to change to the power failure holding area.

[2] Power outage retention area. Use the parameter setting to change to the non-power-off holding area.

[3] Power failure holding area cannot be changed by parameters.

3.7.1 Data Register D

The role of the data register: processing various numerical data, by using it, you can perform various controls. It is used as a set value of a timer and a counter, and is used for various calculations of data, and the like. Some special data registers are used for the system working state parameter cache. These registers can be queried to determine the operating parameters. See the appendix for the power outage retention features of the special data registers.

The data register D is 16 bits. When 32-bit data is used, 32 adjacent data is represented by 2 adjacent data registers, the lower 16 bits of data are stored in the lower address (Dn), and the upper 16 bits of data are stored in the upper address. (Dn+1).

When 32-bit data is specified, if the low bit (Dn) is specified, the high bit is automatically occupied by the number following it (Dn+1). The lower bits can be specified by any of the odd or even device numbers. Considering the monitoring function of the peripheral device, it is recommended that the lower bits use the even device number.

Data register of the non-power-off holding area: Once the data is written in the register, it will not change as long as other data is no longer written. However, when power is lost or RUN → STOP, the data of the register will be cleared to 0. (If you drive a special auxiliary relay M8033, you can keep it).

Data register of the power failure holding area: The data of the register will be held during power failure or from RUN to STOP.

When the power-down dedicated data register is used for general purposes, use the RST or ZRST instruction at the beginning of the program to clear the data in the register.

The special purpose data register refers to the data written for a specific purpose, which is used to implement some special functions of the controller, and can be understood as a special unit for data interaction between the user program and the PLC system program.

3.7.2 Index register V, Z

The function of the index register is the same as the normal data register, which is a 16-bit data register for reading and writing numerical data. In the operand of the application instruction, it can also be used in combination with other device numbers or values. However, it should be noted that the device numbers of basic sequence commands such as LD, AND, OUT, or step ladder instructions cannot be combined with the index register.

There are 16 V0 to V7 and Z0 to Z7. When combined into 32 bits, V is in the high position and Z is in the low position.

The V and Z registers can be accessed in 16-bit and 32-bit modes, as shown in the following figure:

16-bit independent register for 16-bit access mode

16-bit 16-bit

V0~V7: 8 points Z0~Z7: 8 points

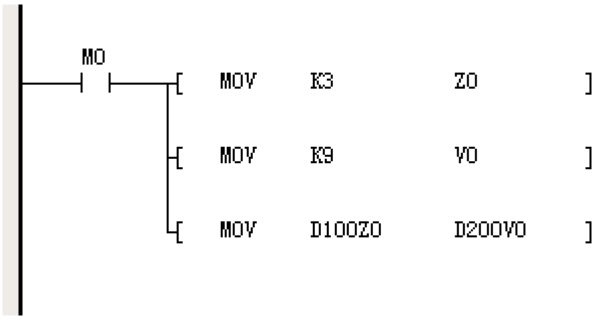
When the 32-bit access mode is used, it is combined into 8 registers as follows

32bits

V0（high bit）	Z0（ low bit ）
V1（ high bit ）	Z1（ low bit ）
V2（ high bit ）	Z2（ low bit ）
V3（ high bit ）	Z3（ low bit ）
V4（ high bit ）	Z4（ low bit ）
V5（ high bit ）	Z5（ low bit ）
V6（ high bit ）	Z6（ low bit ）
V7（ high bit ）	Z7（ low bit ）

When processing a device in a 32-bit application instruction or processing a value exceeding a range of 16 bits, V (high) and Z (low) are simultaneously accessed, and the specified register name must be Z0 to Z7. Indexing cannot be performed even if the high side of V0~V7 is specified.

(1) 16-bit index application example:

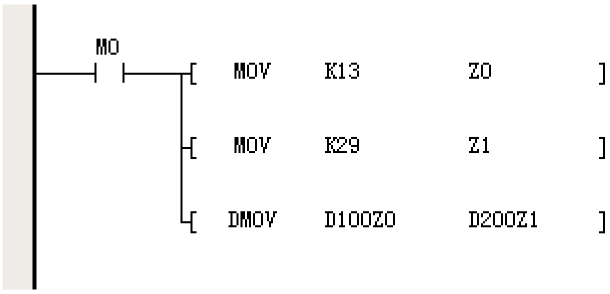


D100Z0=D（100+3）=D103

D200V0=D（200+9）=D209

Equivalent：MOVD103D209

(2) 32-bit index application example:

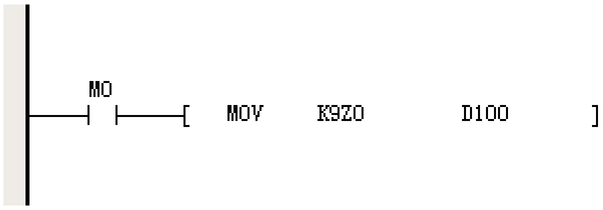


D100Z0=D（100+13）=D113

D200V0=D（200+29）=D229

Equivalent：DMOVD113D229

(3) Special cases of constant indexing:



The completed operation is to add the value of Z0 to K5 and then to D10. The

When the V and Z indirect addressing modes are used in the loop instruction (V, Z changes with the loop variable), the operation of the sliced data area is performed, or the table lookup operation is performed, the programming is simplified, and the instruction efficiency is improved.

3.7.3 File register R

The use of the file register R is the same as that of the data register D.

3.8 Labels and subroutines

The label/jump pointer (P) is used to identify the entry address of the jump program, the subroutine SBR is used for the identification of the start address of the subroutine, the motion control subroutine is labeled with MC, and the interrupt subroutine (I) is used for The start address identifier of the interrupt program, the number of which is assigned in decimal.

Label	Subroutine	Overview
P	CJ instruction	Used in conjunction with the LBL instruction. Labels are used within each block and cannot be jumped outside of the current block. A total of 512 jump labels are allowed for all blocks
L	CJ instruction	Equivalent to P

<table>
<tr><td>SBR</td><td>CALL instruction</td><td colspan="2">Supports up to 512 subroutines; subroutine properties can be set to normal subroutines, encrypted subroutines, subroutines with parameters, encrypted subroutines with parameters; encrypted subroutines, subroutines with the same subroutine capacity Restricted, together occupy the capacity of the system 64K steps.</td></tr>
<tr><td rowspan="4">I</td><td rowspan="4">Interrupt subroutine</td><td>External Interrupt</td><td>X000-X002 input interrupt, number I00□, I10□, I20□, 3 points, (□ indicates: 0 falling edge interrupt, 1 rising edge interrupt). After the interrupt disable flag register is turned ON, the corresponding input interrupt is disabled.</td></tr>
<tr><td>Timed interrupt</td><td>I6□□, I7□□, I8□□, 3 points
(□□=1~99, time base=1ms)</td></tr>
<tr><td>Count completion interrupt</td><td>I010, I020, I030, I040, I050, I060, I070, I080, 8 points
(for DHSCS instructions)</td></tr>
<tr><td>Pulse completion interrupt</td><td>I502~I504, 3 points</td></tr>
<tr><td>MC</td><td>Motion control subroutine</td><td colspan="2">Supports up to 64; number MC0–MC63;
Also supports 1 G-code code subroutine, number MC10000,
Multiple Oxxxxs are supported in the G-code subroutine file, numbered O0000-O9999.
The motion subroutine has the same capacity as other subroutines, and it takes up 64K steps of the system.</td></tr>
</table>

For details on how to use interrupts and subroutine pointers, refer to "Chapter 7 Interrupts " and "Chapter 9 Subroutines".

3.9 Constant

Column programmable controllers use five types of values for different purposes and purposes. Its role and function are as follows:

Types of	Application note in programming
Decimal number, DEC	Timer and counter settings (K constant) Number of auxiliary relay (M), timer (T), counter (C), status relay S, etc. (Device number) specifies the value and instruction action (K constant) in the operand of the application instruction
Hexadecimal number, HEX	Same as the decimal number, used to specify the operand and the specified action (H constant) in the application instruction
Binary, BIN	The timer, counter, or data register is specified numerically in decimal or hexadecimal numbers, but within the programmable controller, these numbers are treated as binary numbers. Moreover, when monitoring on a peripheral device, these devices are automatically converted to decimal numbers as shown (can also be switched to hexadecimal)
Octal, OCT	The device numbers of the input relays and output relays are assigned in octal values. Therefore, the carry of [0-7, 10-17...70-77, 100-107] can be performed. In the octal number, there is no [8,9]

BCD	BCD is a 4-digit binary representation of the decimal number 0-9 values. Everyone's handling is very easy, so it can be used for BCD output digital switch or seven-segment display control.
BIN floating point number	The programmable controller has a high-precision floating-point operation function, and internally uses binary (BIN) floating-point numbers for floating-point operations.
Decimal floating point number	Decimal floating point values are only used for monitoring and are easy to read.

[K] is a symbol indicating a decimal integer. It is mainly used to specify the setting value of the timer or counter or the value in the operand of the application instruction. In the 16-bit instruction, the value of the constant K ranges from -32768 to 32767. In the 32-bit instruction, the value of the constant K ranges from -2,147,483,648 to 2,147,483,647.

[H] is a representation of a hexadecimal number. Primarily used to specify the value of the operand of an application instruction.

In the 16-bit command, the value of the constant H is 0000 to FFFF.

In the 32-bit instruction, the constant K has a value range of 0x0 to 0xFFFFFFFF.

Chapter 4 Instruction

There are many instructions in the motion control system, which can be divided into the following categories: program logic instructions; program flow instructions, data comparison; data operations; data processing; matrix instructions; string instructions; clock instructions; high-speed input, pulse positioning, communication positioning; Motion control; communication; peripherals; electronic cam commands.

4.1 Program logic instructions

Contact instruction	
LD	Load normally open contacts
LDI	Loading normally closed contacts
AND	Series normally open contact
ANI	Series normally closed contact
OR	Parallel normally open contact
ORI	Parallel normally closed contact
LDP	Take the rising edge of the pulse
LDF	Take the pulse falling edge
ANDP	Serial connection with pulse rising edge detection
ANDF	Serial connection with pulse falling edge detection
ORP	Or pulse rising edge detection parallel connection
ORF	Or pulse falling edge detection parallel connection
INV	Invert the result of the operation
BLD	Bit data bit contact
BLDI	Bit data bit anti-contact
BAND	Bit data bits and contacts
BANI	Bit data bits and non-contact
BOR	Bit data bit or contact
BORI	Bit data bit or non-contact
Combined instruction	
ANB	Series circuit block
ORB	Parallel loop block
MPS	Deposit on the stack
MRD	Read stack (can flow pointer unchanged)
MPP	Read stack
MEP	Energy flow edge control, operation result pulsed
MEF	
Output instruction	
OUT	Drive coil
SET	Set action save coil command
RST	Contact or buffer clear
PLS	Pulse rising edge detection coil command

PLF	Pulse falling edge detection coil command
BOUT	Bit data output
BSET	Bit data set
BRST	Bit data reset
ALT	Alternate output
Other processing instructions	
NOP	No action

4.1.1 Contact Instructions

Contact instruction	
LD	Load normally open contacts
LDI	Loading normally closed contacts
LDP	Take the rising edge of the pulse
LDF	Take the pulse falling edge
AND	Series normally open contact
ANI	Series normally closed contact
ANDP	Serial connection with pulse rising edge detection
ANDF	Serial connection with pulse falling edge detection
OR	Parallel normally open contact
ORI	Parallel normally closed contact
ORP	Or pulse rising edge detection parallel connection
ORF	Or pulse falling edge detection parallel connection
INV	Invert the result of the operation
BLD【Note】	Bit data bit contact
BLDI【Note】	Bit data bit anti-contact
BAND【Note】	Bit data bits and contacts
BANI【Note】	Bit data bits and non-contact
BOR【Note】	Bit data bit or contact
BORI【Note】	Bit data bit or non-contact

【Note】 This instruction is used to selectively extract a bit to participate in a word or double word component. The word instruction takes 5 steps and the double word instruction takes 9 steps. The operands are the same. The first operand is the word or double word component that needs to participate in the operation, and the second parameter is to take the one bit to participate in the operation. When the word instruction is used, the second operand can only take 0-15. When the double word instruction, the second operand can only take 0-31.

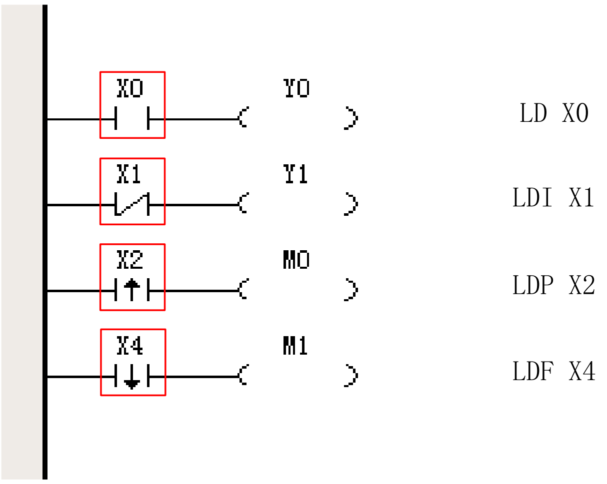
LD/LDI/LDP/LDF

LD	Load normally open contacts	Operand type: S, X, Y, M, T, C Instruction step size: 1step
LDI	Loading normally closed contacts	
LDP	Take the rising edge of the pulse	
LDF	Take the pulse falling edge	

instruction	Operand						
LD LDI LDP LDF	X0~X377	Y0~Y377	M0~M7679 M8000~M8511	S0~S4095	SM0-SM1023	T0~T511	C0~C255
	✓	✓	✓	✓	✓	✓	✓

The LD/LDI/LDP/LDF instructions are used for the contacts at the beginning of the left bus, among them:

- LD The LD/LDI instruction saves the current power flow state of the A contact and the B contact, respectively, and stores the acquired contact state in the accumulation buffer.
- The LDP instruction is used to take the rising edge of the contact signal. If the rising jump of the corresponding signal is detected in this scan, the contact is valid, and the contact becomes invalid at the next scan.
- The LDF command is used to take the falling edge of the contact signal. If the falling transition of the corresponding signal is detected in this scan, the contact is valid, and the contact becomes invalid at the next scan.

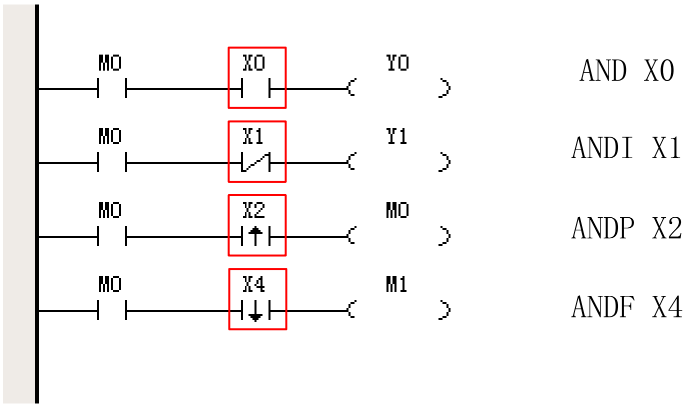


AND/ANI/ANDP/ANDF

AND	Series normally open contact	Operand type: S, X, Y, M, T, C command step: 1step
ANI	Series normally closed contact	
ANDP	Serial connection with pulse rising edge detection	Instruction step size: 3step
ANDF	Serial connection with pulse falling edge detection	

instruction	Operand						
AND ANI ANDP ANDF	X0~X377	Y0~Y377	M0~M7679 M8000~M8511	S0~S4095	SM0-SM1023	T0~T511	C0~C255
	✓	✓	✓	✓	✓	✓	✓

- The AND/ANI/ANDP/ANDF instruction is used for the state operation of the series contact. The operation is to first read the state of the currently specified series contact and then perform an AND operation with the logical operation result before the contact, and the result will be Stored in the cumulative buffer.
- The AND/ANI instruction participates in the AND operation of the state of the A and /B contacts, respectively;
- The ANDP instruction participates in the AND operation on the rising edge transition state of the contact;
- The ANDF instruction is to participate in the AND operation on the falling edge transition state of the contact;

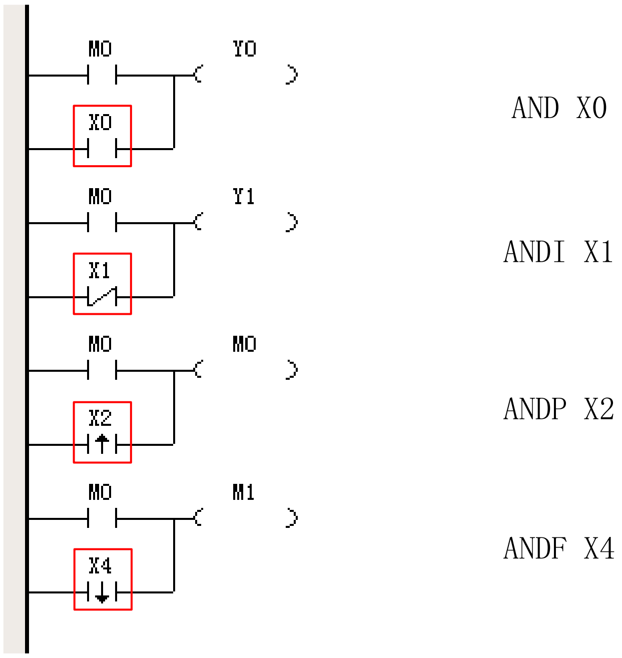


OR/ORI/ORP/ORF

OR	Parallel normally open contact	Operand type: S, X, Y, M, T, C Instruction step size: 1step
ORI	Parallel normally closed contact	
ORP	Or pulse rising edge detection parallel connection	Instruction step size: 3step
ORF	Or pulse falling edge detection parallel connection	

instruction	Operand						
OR ORI ORP ORF	X0~X377	Y0~Y377	M0~M7679 M8000~M8511	S0~S4095	SM0-SM1023	T0~T511	C0~C255
	✓	✓	✓	✓	✓	✓	✓

- The OR/ORI instruction is used for the state operation of the joint point. The operation is to first read the state of the currently specified contact, and then perform an OR operation with the logical operation result before the joint, and store the result in the cumulative cache. Inside the device.
- The OR/ORI instruction participates in the OR operation of the state of the A contact and the /B contact, respectively;
- The ORP instruction participates in the OR operation by hopping the rising edge of the contact;
- The ORF instruction participates in the OR operation on the falling edge transition state of the contact.

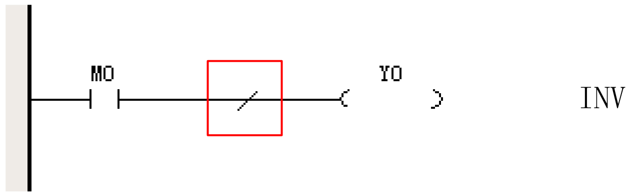


INV Invert the result of the operation

INV	Invert the result of the operation	Instruction step size:1step

instruction	Operand
INV	no

The logical operation result before the INV instruction is inverted and stored in the accumulation buffer. When the flow can be turned ON before the INV instruction, the flow becomes OFF after the INV is passed; otherwise, it turns ON.



BLD Bit data bit contact

1.Instruction form

The state of the specified bit of the source data is turned ON (OFF) to determine the state of the node ON (OFF), the contact directly connected to the left bus.

BLD S n			**Bit data bit contact**	**Instruction execution**	
S	source data	Source data device number		16-bit instruction (5step) BLD continuous execution	32-bit instruction (9step) DBLD continuous execution
n	Loading bit	Load the specified bit, ranging from 0-15 (16-bit instruction) or 0-31 (32-bit instruction)			

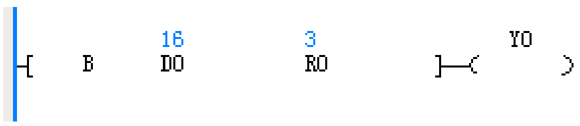
2. Operands

Operand	Bit device							word device														
	System. user							system .user					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

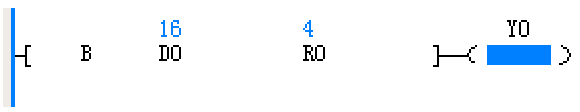
Note: With gray shading device, it means support.

【Example of use】

The case of n=3 is as follows:



The case of n=4 is as follows:



BLDI Bit data bit anti-contact

1. Instruction form

The node whose state is OFF (ON) and directly connected to the left bus is determined according to the state of the source data specified bit ON (OFF).

BLDI S n			Bit data bit anti-contact	Instruction execution	
S	source data	Source data device number		16-bit instruction (5step) BLDI continuous execution	32-bit instruction (9step) DBLDI continuous execution
n	Loading bit	Load the specified bit, ranging from 0-15 (16-bit instruction) or 0-31 (32-bit instruction)			

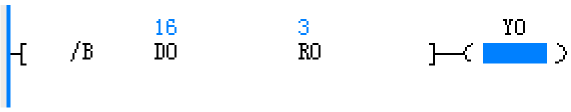
2. Operand

Operand	Bit device							Word device														
	System. user							System. user					Digit designation					Indexing		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

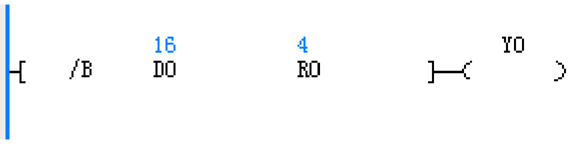
Note: With gray shading device, it means support.

【Example of use】

The case of n=3 is as follows:



n=4 is as follows



BAND Bit data bits and contacts

1.Instruction form

The node whose state is ON (OFF) and which is connected in series with other nodes is determined according to the state ON (OFF) of the source data designation bit.

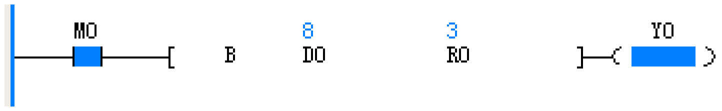
BAND S n			Bit data bits and contacts	Instruction execution	
S	source data	Source data device number		16-bit instruction (5step) BAND continuous execution	32-bit instruction (9step) DBAND continuous execution
n	Loading bit	Load the specified bit, ranging from 0-15 (16-bit instruction) or 0-31 (32-bit instruction)			

2. Operand

Operand	Bit device							Word device														
	system user							system user					Digit designation					Indexing		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

[Example of use]



BANI Bit data bits and non-contact

1.Instruction form

The node whose state is OFF (ON) and which is connected in series with other nodes is determined according to the state of the source data designation bit ON (OFF).

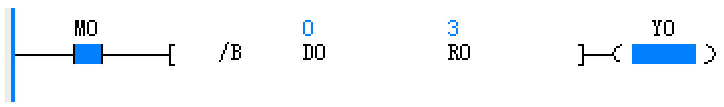
BANI S n		**Bit data bits and non-contact**	**Instruction execution**	
S	source data	Source data device number	16-bit instruction (5step) BANI continuous execution	32-bit instruction (9step) DBANI continuous execution
n	Loading bit	Load the specified bit, ranging from 0-15 (16-bit instruction) or 0-31 (32-bit instruction)		

2. Operand

Operand	Bit device							Word device														
	system user							system user					Digit designation					Indexing		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

[Example of use]



BOR Bit data bit or contact

1. Instruction form

The node whose state is ON (OFF) and which is connected in parallel with other nodes is determined according to the state ON (OFF) of the specified bit of the source data.

BOR S n	Bit data bit or contact	Instruction execution

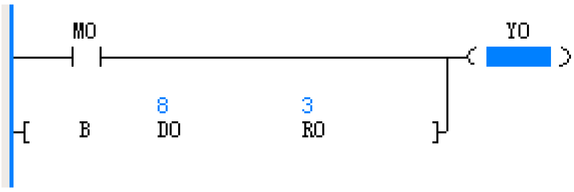
S	source data	Source data device number	16-bit instruction (5step) BOR continuous execution	32-bit instruction (9step) DBOR continuous execution
n	Loading bit	Load the specified bit, ranging from 0-15 (16-bit instruction) or 0-31 (32-bit instruction)		

2. Operand

Operand	Bit device							Word device														
	system user							system user					Digit designation					Indexing		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

[Example of use]



BORI Bit data bit or non-contact

1. Instruction form

The node whose state is OFF (ON) and which is connected in series with other nodes is determined according to the state of the source data designation bit ON (OFF).

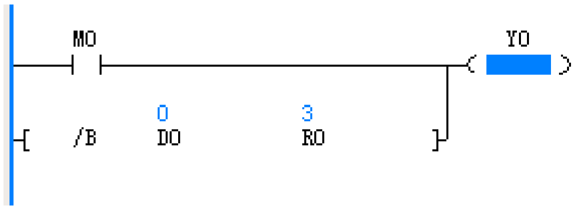
BORI S n		**Bit data bit or non-contact**	**Instruction execution**	
S	source data	Source data device number	16-bit instruction (5step) BORI continuous execution	32-bit instruction (9step) DBORI continuous execution
n	Loading bit	Load the specified bit, ranging from 0-15 (16-bit instruction) or 0-31 (32-bit instruction)		

2. Operand

Operand	Bit device							Word device														
	system user							system user					Digit designation					Indexing		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

[Example]


M0
Y0
0
3
/B
D0
R0

4.1.2 Combined instructions

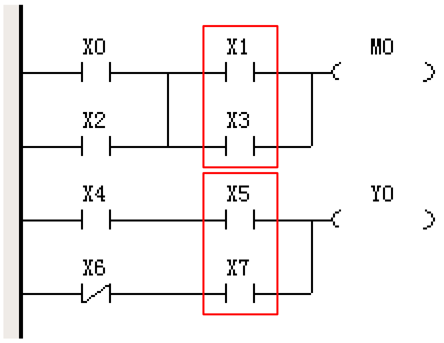
Combined instructions	
ANB	Series circuit block
ORB	Parallel loop block
MPS	Deposit on the stack
MRD	Read stack (energy flow pointer unchanged)
MPP	Read stack
MEP	Energy flow edge control, operation result pulsed
MEF	

ANB Series circuit block, ORB parallel circuit block

ANB	Series circuit block	Instruction step size: 1step
ORB	Parallel loop block	Instruction step size: 1step

Instruction	operand
ANB	none
ORB	Participating in block operations is the computational energy flow of the last two LD (or LDI/LDP/LDF) intervals.

ANB and ORB are operations that "and" and "or" the previously saved logical result with the current accumulated buffer contents.



ANB	ORB
LDX0	LDX4
ORX2	ANDX5
LDX1	LDIX6
ORX3	ANDX7
ANB	ORB
OUTM0	OUTY0

Stack instruction MPS.MRD.MPP

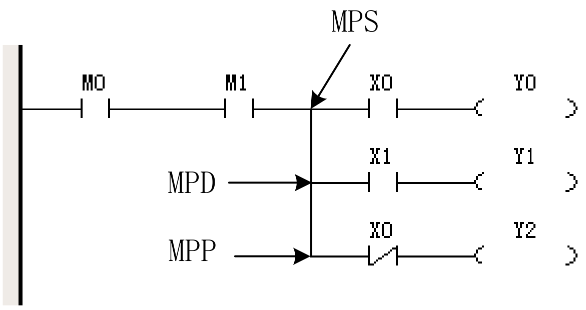
MPS	Deposit on the stack	Instruction step size: 1step
MRD	Read stack (can flow pointer unchanged)	Instruction step size: 1step
MPP	Read stack	Instruction step size: 1step

Instruction	operand
MPS MRD MPP	none

MPS: Stores the contents of the current accumulated buffer on the stack. (Stack pointer plus one).

MRD: The contents of the read stack are stored in the accumulation buffer. (The stack pointer does not move).

MPP: The result of the previous saved logical operation is retrieved from the stack and stored in the accumulation buffer. (The stack pointer is decremented by one).



LDM0
ANDM1
MPS
ANDX0
OUTY0
MRD
ANDX1
OUTY1
MPP
ANIX0
OUTY2

MEP /MEF Pulsed operation result

1. Instruction form

2. An instruction is an instruction that pulsing an operation result without specifying a device number

3. (1) MEP: The calculation result until the MEP command is turned from ON to ON.

4. (2) MEF: The calculation result until the MEF command is turned from ON to OFF.

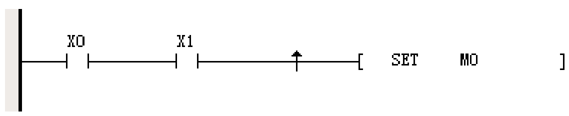
5. (3) Step size: MEP/MEF are both 1step.

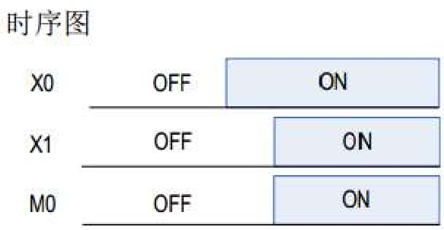
6. 2. Operands

Operand	Bit device	Word device				
	system .user	system user	Digit designation	Indexing	Constant	Real number
MEP	Objectless device					
MEF	Objectless device					

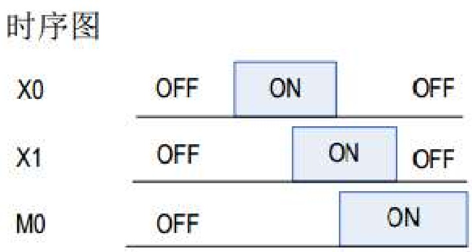
[Example of use]

(1) MEP instruction (the rising edge of the operation result is ON)





(2) MEF instruction (the falling edge of the operation result is ON)



4.1.3 Output Instructions

Output Instructions	
OUT	Drive coil
SET	Set action save coil command
RST	Contact or buffer clear
PLS	Pulse rising edge detection coil command
PLF	Pulse falling edge detection coil command
BOUT	Bit data output
BSET	Bit data set
BRST	Bit data reset
ALT	Alternate output

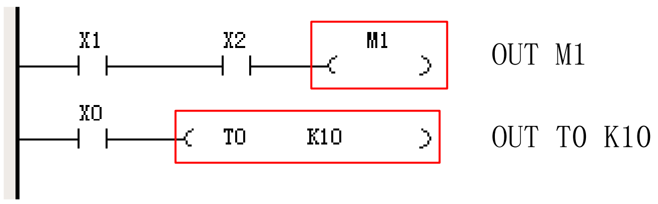
The operands are the same. The first operand is the word or double word component that needs to participate in the operation, and the second parameter is to take the one bit to participate in the operation. When the word instruction is used, the second operand can only take 0-15. When the double word instruction, the second operand can only take 0-31.

OUT/SET/RST/PLS/PLF

<table>
<tr><td>OUT</td><td>Drive coil</td><td rowspan="2">Operand type: S, Y, M instruction step: 1step</td></tr>
<tr><td>SET</td><td>Set action save coil command</td></tr>
<tr><td>RST</td><td>Contact or buffer clear</td><td>Operand type: S, Y, M, T, C, D
Instruction step size: 3step</td></tr>
<tr><td>PLS</td><td>Pulse rising edge detection coil command</td><td></td></tr>
<tr><td>PLF</td><td>Pulse falling edge detection coil command</td><td></td></tr>
</table>

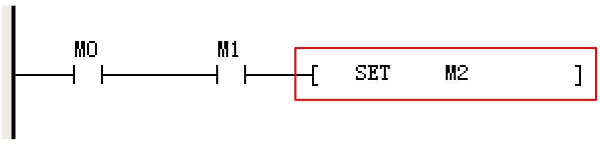
<table>
<tr><td>Instruction</td><td colspan="7">operand</td></tr>
<tr><td rowspan="2">OUT</td><td>X0~X377</td><td>Y0~Y377</td><td>M0~M7679
M8000~M8511</td><td>S0~S4095</td><td>SM0-SM1023</td><td>T0~T511</td><td>C0~C255</td></tr>
<tr><td></td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>

Outputs the result of the logical operation before the OUT instruction to the specified component.



instruction	Operand						
SET	X0~X377	Y0~Y377	M0~M7679 M8000~M8511	S0~S4095	SM0-SM1023	T0~T511	C0~C255
		✓	✓	✓	✓		

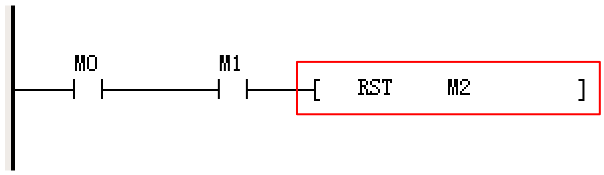
When the SET instruction is driven, its specified component is set to ON, and the set component will remain ON regardless of whether the SET instruction is still driven. This component can be set to OFF using the RST instruction.



SET M2

Instruction	Operand						
RST	X0~X377	Y0~Y377	M0~M7679 M8000~M8511	S0~S4095	SM0-SM1023	T0~T511	C0~C255
		✓	✓	✓	✓	✓	✓
			D0~D8511	R0~R32767	SD0~SD1023		
			✓	✓	✓		

- When the RST instruction is driven, its specified component is set to OFF, and the set component remains OFF regardless of whether the RST instruction is still driven. This component can be turned ON using the SET instruction.
- The RST instruction can also be used to reset the D, V, and Z variables, and clear the values of the specified D, V, and Z components to zero.



RST M2

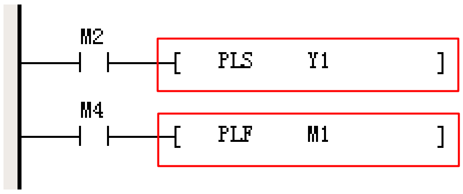
Element	Operation result
S，M，Y	Coil and contact are set to OFF
T，C	The current timing or count value will be set to 0 and the coil and contacts will be set to OFF.
D，V,Z	The value of the component is cleared to 0.

Instruction	Operand						
PLS PLF	X0~X377	Y0~Y377	M0~M7679 M8000~M8511	S0~S4095	SM0-SM1023	T0~T511	C0~C255
		✓	✓		✓		

- When the PLS instruction is driven by the rising edge, its specified component is set to the ON state, which lasts for only one scan cycle.

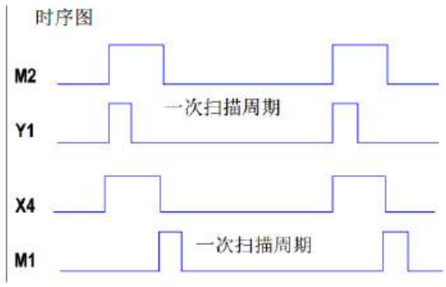
■ When the PLF instruction is driven by the falling edge, its specified component is set to the ON state, which lasts for only 1 scan cycle.

[Example of use]



PLS Y1

PLF M2



BOUT Bit data output

1. Instruction form

Output the result of the logical operation before the BOUT instruction to the specified bit.

<table>
<tr><td colspan="3">BOUT D n</td><td>Bit data output</td><td colspan="2">Instruction execution</td></tr>
<tr><td>D</td><td>source data</td><td>Output data device number</td><td rowspan="2" colspan="2">16-bit instruction (5step) BOUT continuous execution</td><td rowspan="2">32-bit instruction (9step) DBOUT continuous execution</td></tr>
<tr><td>n</td><td>Loading bit</td><td>Output specified bit, range 0-15 (16-bit instruction) or 0-31 (32-bit instruction)</td></tr>
</table>

2. Operands

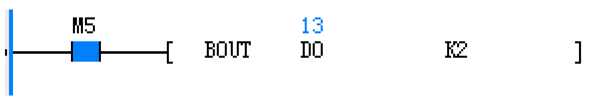
<table>
<tr><td rowspan="2">Operand</td><td colspan="7">Bit device</td><td colspan="15">Word device</td></tr>
<tr><td colspan="7">system .user</td><td colspan="5">system .user</td><td colspan="5">Digit designation</td><td colspan="2">Indexing</td><td colspan="2">Constant</td><td>Real number</td></tr>
<tr><td>D</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
<tr><td>n</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
</table>

Note: With gray shading device, it means support.

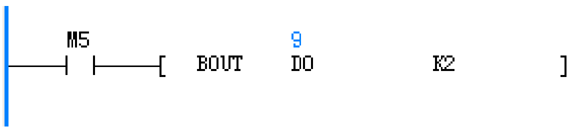
[Example 1]

D0 initial value = 2 #1001 (decimal K9)

The M5=ON condition is as follows, bit 2 of D0 is set, and the result D100=2#1110 (decimal K13)



Then M5=OFF is as follows, bit 2 of D0 is reset, and the result is D0=2#1101 (decimal K13)



BSET Bit data output

1. Instruction form

When the BSET instruction is driven, its specified bit is set to ON, and the set bit remains ON. This bit can be set to OFF using the BRST instruction, regardless of whether the BSET instruction is still driven.

BSET D n			Bit data output	Instruction execution	
D	Actuator	Output data device number		16-bit instruction (5step) BSET continuous execution	32-bit instruction (9step) DBSET continuous execution
n	Output bit	Output specified bit, range 0-15 (16-bit instruction) or 0-31 (32-bit instruction)			

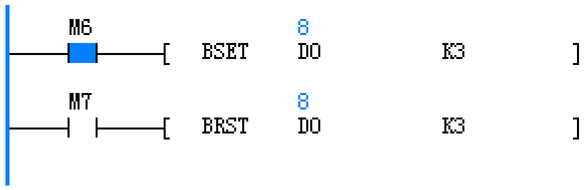
2. Operands

Operand	Bit device							Word device														
	system .user							system .user					Digit designation					Indexing		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

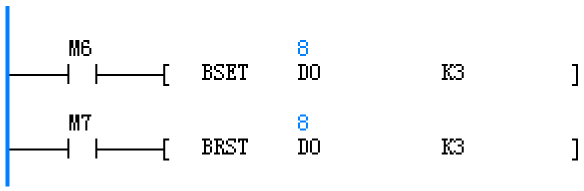
Note: With gray shading device, it means support.

[Example 1]

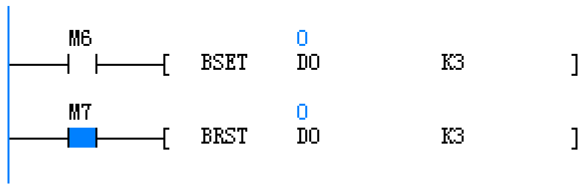
M6=ON



M6=OFF



M7=ON



BRST Bit data output

1. Instruction form

When the BSET instruction is driven, its specified bit is set to OFF.

BRST D n			Bit data reset	Instruction execution	
D	Actuator	Output data device number		16-bit instruction (5step) BRST continuous execution	32-bit instruction(9step) DBRST continuous execution
n	Output bit	Output specified bit, range 0-15 (16-bit instruction) or 0-31 (32-bit instruction)			

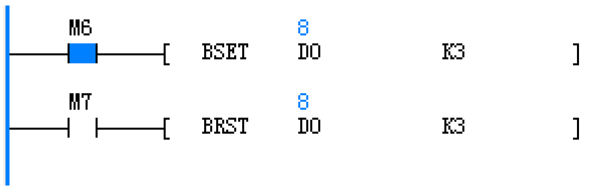
2. Operands

Operands	Bit device							Word device														
	system .user							system .user					Digit designation					Indexing		constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

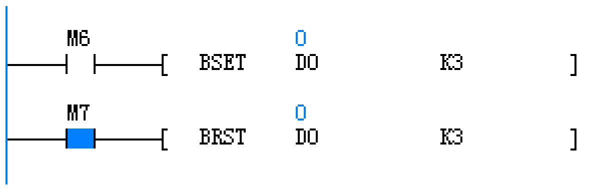
Note: With gray shading device, it means support.

[Example 1].

When M6=ON is as follows:



When M7=ON is as follows:



ALT Alternate output

1. Instruction form

When the driving condition is established, the registration element D performs ON/OFF inversion.

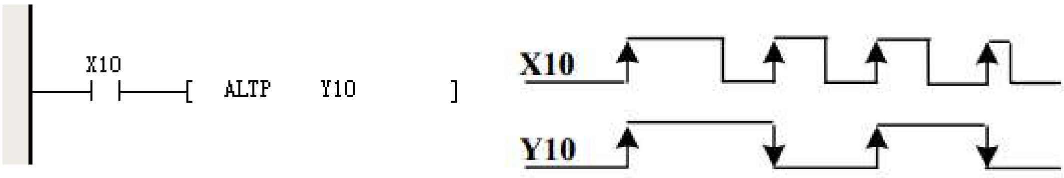
ALT D			Alternate output	Instruction execution
D	Actuator	Bit component		16-bit instruction（3step） ALT continuous execution ALTP Pulse execution

2. Operands

Operands	Bit device							Word device														
	system .user							system .user					Digit designation					Indexing		constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

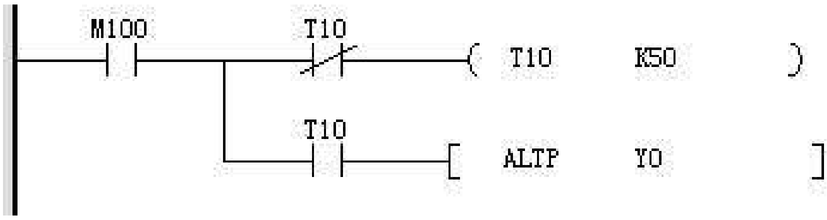
Note: With gray shading device, it means support.

[Example 1].



[Example 2]

If the timer is introduced into the instruction energy stream, the oscillator output can be easily implemented (this function can also be implemented with a special timer STMR instruction).



4.1.4 Other processing instructions

Other processing instructions	
NOP	No action
WDT	Watchdog timer reset

NOP

instruction	Operands
NOP	None

Function description: The instruction NOP does not perform any operation in the program, so the original logic operation result will remain after execution, and there is no actual operation.

4.2 Program Process instruction

Subroutine	
CALL	Subroutine call
SRET	Subroutine return
SSRET	Subroutine with conditional return
IRET	Interrupt return
Interrupt	
EI	Interrupt permission
DI	Interruption

Jump	
CJ	Conditional jump
LBL	Label instruction
CJEND	Condition jumps to the end of the program
cycle	
FOR	Start of cycle
NEXT	End of cycle range

4.2.1 Subroutine

CALL Subroutine call

1. Instruction form

Subroutine call Instruction

CALL S			Subroutine call	Instruction execution
S	Subprogram name	Subroutine call destination pointer label		16-bit instruction（3step） CALL continuous execution CALLP Pulse execution

Function description: When the power flow is valid, the program calls the specified subroutine. After the subroutine is executed, it will return to the next instruction of the CALL (or CALLP) statement and continue to execute the subsequent statement.

(1) The subroutine can be called in multiple places, or can be called by other Subroutines, but the number of nesting layers must not exceed 5 layers.

(2) Do not call itself within a subroutine to prevent an infinite loop or program run timeout.

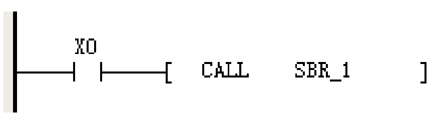
(3) In the subroutine, T192~T199 or T246~T249 can be used as the timer.

The subroutine in the VCAutoDesignsoft software programming environment is written in a separate window. There are no instruction problems such as FEND and SRET, and the subroutine name supports arbitrary Modify (including Chinese).

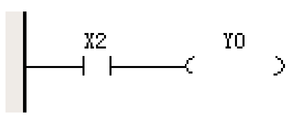
E.g:

The default program name can be changed to a more meaningful name via the subroutine properties dialog

主程序



子程序SBR_1



SRET Subroutine return

1. Instruction form

Subroutine return

SRET	Subroutine return	Instruction execution
No need for contact drive, no single instruction with operands		16-bit instruction（1step） SRET　continuous execution

Function description: In the VCAutoDesignsoft software programming environment, the system does not need to input the SRET instruction, the system will automatically join when downloading.

SSRET Subroutine return

1. Instruction form

Subroutine with conditional return

SSRET	Subroutine return	Instruction execution
No need for contact drive, no single instruction with operands		16-bit instruction（1step） SSRET　continuous execution

Function description: In the VCAutoDesignsoft software programming environment, the system does not need to input the SSRET instruction, the system will automatically join when downloading.

IRET Interrupt program completed

1. Instruction form

Subroutine with conditional return

IRET	Interrupt program completed	Instruction execution
No need for contact drive, no single instruction with operands		16-bit instruction（1step） IRET　continuous execution

Function description: The IRET statement is located at the end of the interrupt subroutine. After executing the instruction, it will return to the statement before the interrupt subroutine is called to continue the program execution. In the VCAutoDesignsoft software programming environment, the interrupt program is written in a separate window, without the user inputting the IRET instruction, the system will automatically join when downloading.

4.2.2 Interrupt

EI/DI Interrupt permission/interrupt disable

1. Instruction form

Interrupt permission /interrupt disable

EI	Interrupt permission	Instruction execution
DI	interrupt disable	
No need for contact drive, no single instruction with operands		16-bit instruction（1step）EI/DI continuous execution

Function description: When the PLC program starts running, the default is the interrupt disable state; after the EI statement is executed, the interrupt function is allowed; when the interrupt is enabled, after the DI statement is executed, the interrupt disable state is entered.

Types and settings of the interrupt:

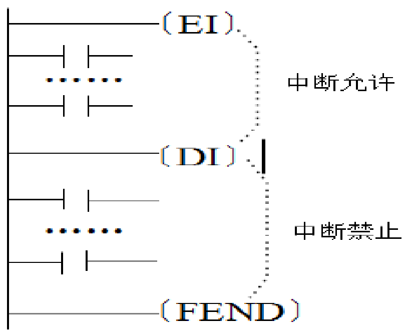
(1) External signal input interrupt: It can define the rising edge or falling edge of the X0~X4 input signal for interrupt. For the X signal that does not require immediate response, the pulse capture function can also be used.

(2) High-speed counter interrupt: compare and set the instruction with DHSCS, and generate an interrupt when the current value of the high-speed counter reaches the set value;

(3) Timer interrupt: an interrupt that occurs at a fixed period of 1 ms to 99 ms;

(4) Pulse completion interrupt: Immediately after the specified number of pulses is sent, the interrupt is executed;

(5) Multi-user interrupt: A high-speed counter can be arbitrarily selected for up to 24 interrupts.



Programming and execution characteristics of interrupts:

An interrupt occurs between the DI-EI instructions (interrupt disable interval) and can also be memorized and executed after the EI instruction. After the interrupt subroutine must be written to the FEND instruction, the end of the subroutine must end with an IRET. In the VCAutoDesignsoft software programming environment, do not write in the main program, the subroutine can omit the IRET.

The pointer number cannot be reused.

When multiple interrupts occur in sequence, the one that occurs first takes precedence. When the simultaneous occurrence occurs completely, the priority is higher. The priority levels from high to low are high-speed counter interrupt, external interrupt, time interrupt, and pulse output completion interrupt.

Other interrupts are disabled during the execution of the interrupt routine.

When controlling the input relay and output relay during interrupt processing, the input/output refresh command (REFF) can be used to achieve high-speed control by reading the latest input status or immediately outputting the operation result.

The number of the input relay used as the interrupt pointer should not be the same as the number of the application command such as [High Speed Counter] [Pulse Density (FNC56)] using the same input range.

For the timers in the subroutine and interrupt routines, use the timer T192-T199 for routines. If a general timer is used, in addition to not being able to perform timing, it is necessary to pay attention when using the 1ms cumulative timer.

If the input interrupt pointer I port 0 is specified, the input filter characteristic of the input relay is automatically turned off. Therefore, it is not necessary to use the REFE (FNC51) instruction and the special data register D8020 (input filter adjustment). In addition, the input filter of the input relay that is not used as an input interrupt pointer can be maintained for 10 ms (initial value).

For details, please see "Interruption Introduction".

4.2.3 Jump

CJ Conditional jump

1. Instruction form

An instruction to execute a program jump when the condition is satisfied.

Note: Operands can also use L, which is equivalent to P.

CJ/CJP P000~P511		Conditional jump	Instruction execution
P	label	Conditional transfer destination pointer label	16-bit instruction（3step） CJ continuous execution CJP Pulse execution

Function Description:

① 1 When the power flow is valid, the program automatically jumps from the address of the CJ (or CJP) instruction to the address specified by the P tag and continues execution. The program instruction of the intermediate address is skipped and is not executed.

② 2 When the power flow is invalid, the program continues to execute, and the CJ (or CJP) instruction is not executed.

③ 3 If there is a counter in the program in the intermediate address area that is crossed and it has been driven, the action is:

Implementation	CJ has a jump	CJ no jump
T192~T199	Normal execution	Normal execution
Other timer	Stop timing	
C235~C255	Normal execution	
Other counter	Stop counting	

The requirements for the P tag are as follows:

The CJ instruction must be used in conjunction with the LBL instruction, and the destination label must be in the current block and cannot be jumped across the block;

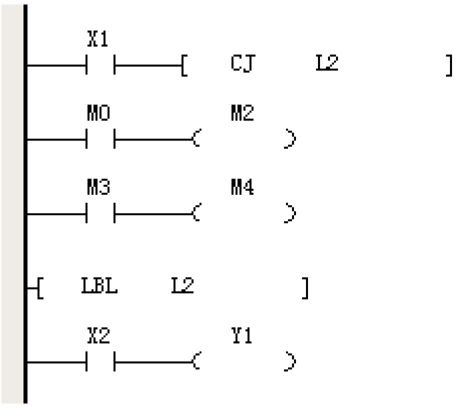
The defined address of the P tag cannot be duplicated in the same block;

When the user wants some part of the program to not need to be executed, or wants to use two coil outputs, to avoid the appearance of double coils. Can use this instruction;

The CJ instruction can repeatedly specify the same pointer P.

Example of instruction:

In the VCAutoDesignsoft software programming environment, the jump instructions are used as follows:



Since the subroutine and the interrupt subroutine are written in a separate window, there is no need to pay attention to matters such as FEND, and the instruction to jump to the end is CJEND in the VCAutoDesignsoft software programming environment.

LBL Label instruction

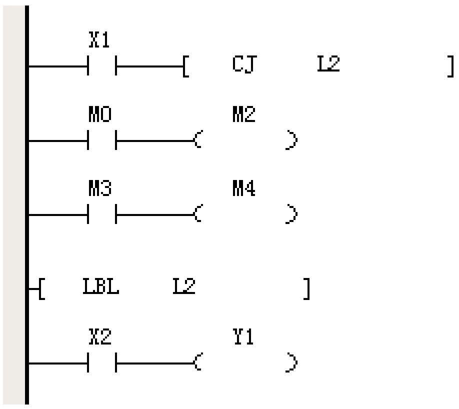
1. Instruction form

The label instruction, used in conjunction with the CJ instruction, is used to mark the target location of the jump.

LBL P000~P511			Conditional jump	Instruction execution
P	label	Target label for conditional transfer		16-bit instruction（3step） BLB continuous execution

Note: Operands can also use L, which is equivalent to P.

[Example of use]



CJEND Condition jumps to the end of the program

1. Instruction form

When the condition is satisfied, the execution program jumps to the end of the program, and the execution of this scan cycle ends.

CJEND	Condition jump to the end of the program	Instruction execution
No need for contact drive, no single instruction with operands		16-bit instruction（3step） CJEND continuous execution

4.2.4 cycle

FOR Start of cycle range

1. Instruction form

Start of cycle range

FOR S1			Start of cycle range	Instruction execution
S1	cycles	Number of loop cycles		16-bit instruction（3step） FOR continuous execution

2. Operands

Operands	Bit device							Word device													
	system .user							system .user					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Functional Description: The FOR instruction is used for the start of a loop and indicates the number of loop executions that must be used in conjunction with the NEXT instruction. Where: S1 is the loop number control variable.

NEXT End of cycle range

1. Instruction form

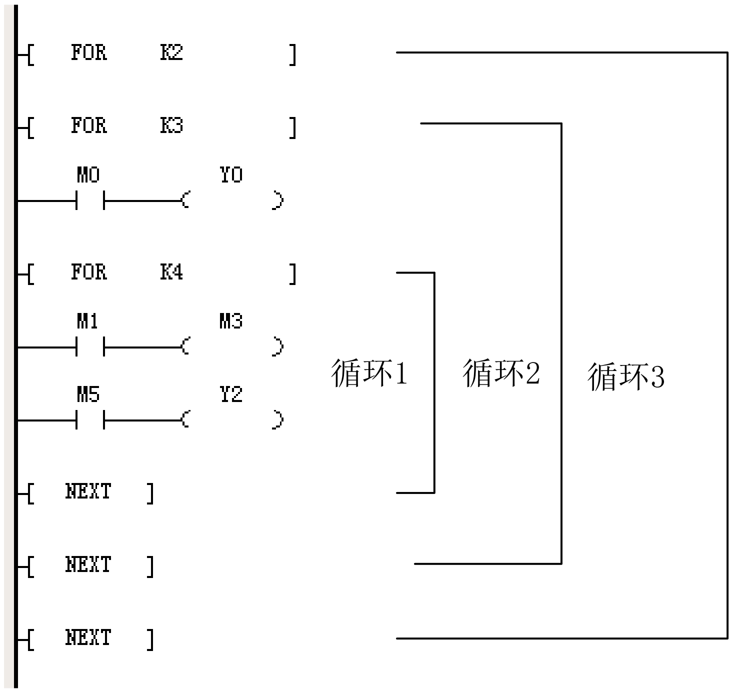
The loop range ends.

NEXT	Start of cycle	Instruction execution
Separate instructions without Operands		16-bit instruction（1step） NEXT continuous execution

Function Description:

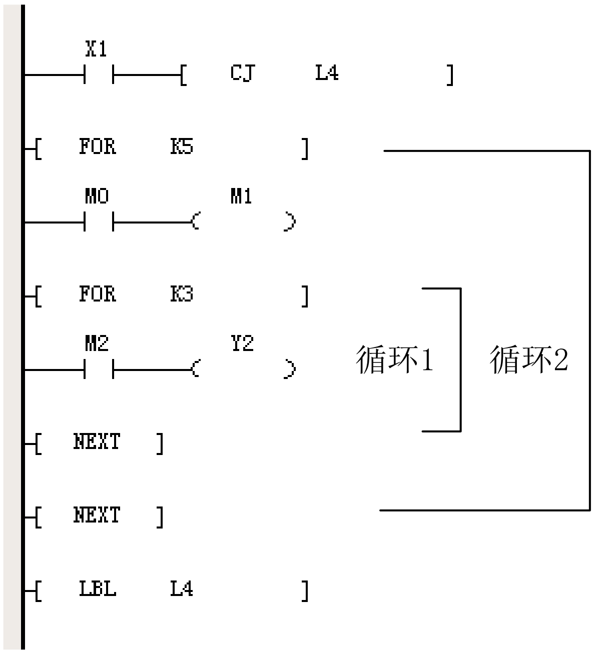
- The NEXT instruction is used to indicate the tail of the loop area. The FOR~NEXT loop specified by the FOR instruction is executed N times and then jumps out of the FOR~NEXT loop to continue execution.
- In the loop interval of the FOR~NEXT instruction, another FOR~NEXT loop can be embedded, but it is stipulated that up to the outer layer of FOR~NEXT can be embedded with up to 6 layers of FOR~NEXT loops. At runtime, the PLC will perform parsing with each FOR~NEXT layer. However, it should be noted that when the number of cycles is too large, the PLC scan cycle will be prolonged, which may cause the overtime watchdog timer to operate and cause an error. This can be improved by using the WDT instruction between FOR~NEXT instructions.
- A FOR instruction corresponds to a NEXT instruction, no more and no less, and a NEXT instruction cannot be written after END/FEND. The FOR instruction precedes the corresponding NEXT instruction.

[Example 1].



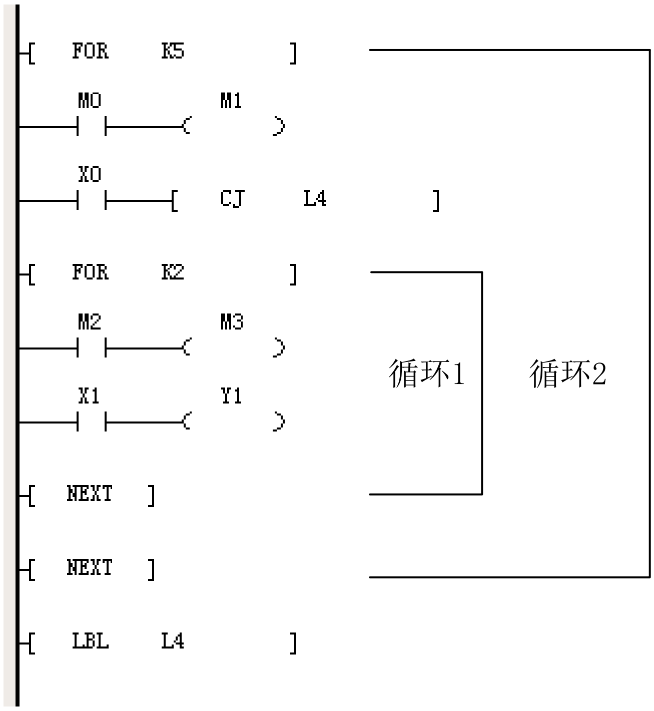
After the loop 3 is executed twice, the program after the NEXT instruction continues to execute, and the loop 3 executes 3 times for each execution of the loop 2, and the loop 2 executes the loop 1 and performs 4 times for each execution, so the loop 1 executes 2*3 in total. *4=24 times, loop 2 performs 2*3=6 times.

[Example 2]



When you want to skip the FOR~NEXT instruction, you can use the CJ jump instruction. When X1 is OFF in the example, execute loop 1 and loop 2. When X0 is ON, the CJ instruction jumps to L4, loop 1 and loop 2. The program between them is not executed.

[Example 3]



When you want to skip the FOR~NEXT instruction nested in the loop or jump out of the loop, you can also use the CJ jump instruction. When X0 is OFF, loop 1 in loop 2 is executed. When X1 is ON, the CJ instruction jumps to L1, and the loop 1FOR~NEXT nested in loop 2 is skipped by the CJ instruction.

4.3 Data comparison instructions

Data comparison instruction: contact comparison, comparison output.

contact comparison	
LD=	LD contact comparison, equal
LD>	LD contact comparison, greater than
LD<	LD contact comparison, less than
LD<>	LD contact comparison, not equal
LD>=	LD contact comparison, greater than or equal to
LD<=	LD contact comparison, less than or equal to
AND=	AND contact comparison, equal
AND>	AND contact comparison, greater than
AND<	AND contact comparison, less than
AND<>	AND contact comparison, not equal
AND>=	AND contact comparison, greater than or equal to
AND<=	AND contact comparison, less than or equal to
OR=	OR contact comparison, equal
OR>	OR contact comparison, greater than
OR<	OR contact comparison, less than
OR<>	OR contact comparison, not equal
OR>=	OR contact comparison, greater than or equal to

OR<=	OR contact comparison, less than or equal to
LD&	LD logical operation, and
LD|	LD logic operation, or
LD^	LD logic operation, XOR
AND&	AND logical operation, and
AND|	AND logic operation, or
AND^	AND logic operation, XOR
OR&	OR logical operation, and
OR|	OR logic operation, or
OR^	OR logic operation, XOR
FLD>	Floating number >: Compare status contact, when S1>S2, turn on
FLD>=	Floating number >=: Compare state contact, when S1≧S2, turn on
FLD<	Floating number <: Compare state contact, when S1＜S2, turn on
FLD<=	Floating number <=: comparison state contact, when S1≦S2, turn on
FLD=	Floating number =: Compare status contacts, when S1=S2, turn on
FLD<>	Floating number <>: Compare status contacts, when S1≠S2, turn on
FAND>	Floating number >: Compare [and] status contacts, when S1>S2, turn on
FAND>=	Floating number >=: Compare [and] status contacts, when S1≧S2, turn on
FAND<	Floating number <: Compare [and] status contacts, when S1＜S2, turn on
FAND<=	Floating number <=: Compare [and] status contacts, when S1≦S2, turn on
FAND=	Floating number =: Compare [and] status contacts, when S1＝S2, turn on
FAND<>	Floating number <>: Compare [and] status contacts, when S1≠S2, turn on
FOR>	Floating number >: Compare [or] status contacts, when S1＞S2, turn on
FOR>=	Floating number >= :Compare [or] status contacts, when S1≧S2, turn on
FOR<	Floating number < :Compare [or] status contacts, when S1＜S2, turn on
FOR<=	Floating number <=:Compare [or] status contacts, when S1≦S2, turn on
FOR=	Floating number =:Compare [or] status contacts, when S1＝S2, turn on
FOR<>	Floating number<>:Compare [or] status contacts, when S1≠S2, turn on
LDZ>	Absolute value>: Compare status contact, when |S1-S2| > |S3|, turn on
LDZ>=	Absolute value>=: Compare state contact, when |S1-S2|≧|S3|, turn on
LDZ<	Absolute value<: Compare state contact, when |S1-S2| < |S3|, turn on
LDZ<=	Absolute value<=: comparison state contact, when |S1-S2|≦|S3|, turn on
LDZ=	Absolute value=: Compare status contacts, when |S1-S2| = |S3|, turn on
LDZ<>	Absolute value<>: Compare status contacts, when |S1-S2|≠|S3|, turn on
ANDZ>	Absolute value>: Compare [and] status contacts, when |S1-S2| > |S3|, turn on
ANDZ>=	Absolute value>=: Compare [and] status contacts, when |S1-S2|≧|S3|, turn on
ANDZ<	Absolute value<: Compare [and] status contacts, when |S1-S2| < |S3|, turn on
ANDZ<=	Absolute value<=: Compare [and] status contacts, when |S1-S2|≦|S3|, turn on
ANDZ=	Absolute value=: Compare [and] status contacts, when |S1-S2| = |S3|, turn on
ANDZ<>	Absolute value<>: Compare [and] status contacts, when |S1-S2|≠|S3|, turn on
ORZ>	Absolute value>: Compare [or] status contacts, when |S1-S2| > |S3|, turn on
ORZ>=	Absolute value>= :Compare [or] status contacts, when|S1-S2|≧|S3|, turn on
ORZ<	Absolute value< :Compare [or] status contacts, when |S1-S2| < |S3|, turn on

<table>
<tr><td>ORZ<=</td><td>Absolute value<=:Compare [or] status contacts, when |S1-S2|≦|S3|, turn on</td></tr>
<tr><td>ORZ=</td><td>Absolute value=:Compare [or] status contacts, when |S1-S2| = |S3|, turn on</td></tr>
<tr><td>ORZ<></td><td>Absolute value <>:Compare [or] status contacts, when |S1-S2|≠|S3|, turn on</td></tr>
<tr><td colspan="2">Output comparison instruction</td></tr>
<tr><td>CMP</td><td>Data comparison</td></tr>
<tr><td>ECMP</td><td>Binary floating point comparison</td></tr>
<tr><td>ZCP</td><td>Regional comparison</td></tr>
<tr><td>EZCP</td><td>Binary floating point interval comparison</td></tr>
</table>

4.3.1 Contact comparison

<table>
<tr><th colspan="2">Contact comparison</th></tr>
<tr><td>LD=</td><td>LD contact comparison, equal</td></tr>
<tr><td>LD></td><td>LD contact comparison, greater than</td></tr>
<tr><td>LD<</td><td>LD contact comparison, less than</td></tr>
<tr><td>LD<></td><td>LD contact comparison, not equal</td></tr>
<tr><td>LD>=</td><td>LD contact comparison, greater than or equal to</td></tr>
<tr><td>LD<=</td><td>LD contact comparison, less than or equal to</td></tr>
<tr><td>AND=</td><td>AND contact comparison, equal</td></tr>
<tr><td>AND></td><td>AND contact comparison, greater than</td></tr>
<tr><td>AND<</td><td>AND contact comparison, less than</td></tr>
<tr><td>AND<></td><td>AND contact comparison, not equal</td></tr>
<tr><td>AND>=</td><td>AND contact comparison, greater than or equal to</td></tr>
<tr><td>AND<=</td><td>AND contact comparison, less than or equal to</td></tr>
<tr><td>OR=</td><td>OR contact comparison, equal</td></tr>
<tr><td>OR></td><td>OR contact comparison, greater than</td></tr>
<tr><td>OR<</td><td>OR contact comparison, less than</td></tr>
<tr><td>OR<></td><td>OR contact comparison, not equal</td></tr>
<tr><td>OR>=</td><td>OR contact comparison, greater than or equal to</td></tr>
<tr><td>OR<=</td><td>OR contact comparison, less than or equal to</td></tr>
<tr><td>LD&</td><td>LD logical operation, and</td></tr>
<tr><td>LD|</td><td>LD logic operation, or</td></tr>
<tr><td>LD^</td><td>LD logic operation, XOR</td></tr>
<tr><td>AND&</td><td>AND logical operation, and</td></tr>
<tr><td>AND|</td><td>AND logic operation, or</td></tr>
<tr><td>AND^</td><td>AND logic operation, XOR</td></tr>
<tr><td>OR&</td><td>OR logical operation, and</td></tr>
<tr><td>OR|</td><td>OR logic operation, or</td></tr>
<tr><td>OR^</td><td>OR logic operation, XOR</td></tr>
<tr><td>FLD></td><td>Floating number >: Compare status contact, when S1>S2, turn on</td></tr>
<tr><td>FLD>=</td><td>Floating number >=: Compare state contact, when S1≧S2, turn on</td></tr>
<tr><td>FLD<</td><td>Floating number <: Compare state contact, when S1＜S2, turn on</td></tr>
<tr><td>FLD<=</td><td>Floating number <=: comparison state contact, when S1≦S2, turn on</td></tr>
</table>

FLD=	Floating number =: Compare status contacts, when S1=S2, turn on
FLD<>	Floating number <>: Compare status contacts, when S1≠S2, turn on
FAND>	Floating number >: Compare [and] status contacts, when S1>S2, turn on
FAND>=	Floating number >=: Compare [and] status contacts, when S1≧S2, turn on
FAND<	Floating number <: Compare [and] status contacts, when S1＜S2, turn on
FAND<=	Floating number <=: Compare [and] status contacts, when S1≦S2, turn on
FAND=	Floating number =: Compare [and] status contacts, when S1＝S2, turn on
FAND<>	Floating number <>: Compare [and] status contacts, when S1≠S2, turn on
FOR>	Floating number >: Compare [or] status contacts, when S1＞S2, turn on
FOR>=	Floating number >= :Compare [or] status contacts, when S1≧S2, turn on
FOR<	Floating number < :Compare [or] status contacts, when S1＜S2, turn on
FOR<=	Floating number <=:Compare [or] status contacts, when S1≦S2, turn on
FOR=	Floating number =:Compare [or] status contacts, when S1＝S2, turn on
FOR<>	Floating number<>:Compare [or] status contacts, when S1≠S2, turn on
LDZ>	Absolute value>: Compare status contact, when |S1-S2| > |S3|, turn on
LDZ>=	Absolute value>=: Compare state contact, when |S1-S2|≧|S3|, turn on
LDZ<	Absolute value<: Compare state contact, when |S1-S2| < |S3|, turn on
LDZ<=	Absolute value<=: comparison state contact, when |S1-S2|≦|S3|, turn on
LDZ=	Absolute value=: Compare status contacts, when |S1-S2| = |S3|, turn on
LDZ<>	Absolute value<>: Compare status contacts, when |S1-S2|≠|S3|, turn on
ANDZ>	Absolute value>: Compare [and] status contacts, when |S1-S2| > |S3|, turn on
ANDZ>=	Absolute value>=: Compare [and] status contacts, when |S1-S2|≧|S3|, turn on
ANDZ<	Absolute value<: Compare [and] status contacts, when |S1-S2| < |S3|, turn on
ANDZ<=	Absolute value<=: Compare [and] status contacts, when |S1-S2|≦|S3|, turn on
ANDZ=	Absolute value=: Compare [and] status contacts, when |S1-S2| = |S3|, turn on
ANDZ<>	Absolute value<>: Compare [and] status contacts, when |S1-S2|≠|S3|, turn on
ORZ>	Absolute value>: Compare [or] status contacts, when |S1-S2| > |S3|, turn on
ORZ>=	Absolute value>= :Compare [or] status contacts, when|S1-S2|≧|S3|, turn on
ORZ<	Absolute value< :Compare [or] status contacts, when |S1-S2| < |S3|, turn on
ORZ<=	Absolute value<=:Compare [or] status contacts, when |S1-S2|≦|S3|, turn on
ORZ=	Absolute value=:Compare [or] status contacts, when |S1-S2| = |S3|, turn on
ORZ<>	Absolute value <>:Compare [or] status contacts, when |S1-S2|≠|S3|, turn on

LD※Contact comparison

1. Instruction form

The two operands are compared, the comparison result is output in a logical state, and the variables participating in the comparison are processed as signed numbers.

<table>
<tr><td colspan="3">LD※ S1 S2</td><td>Contact type data comparison</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S1</td><td>Comparison number 1</td><td colspan="2">Data source or data variable unit 1 to be compared</td><td rowspan="2">16-bit instruction (5step)
LD= Continuous execution</td><td rowspan="2">32-bit instruction (9step)
LDD= Continuous execution</td></tr>
<tr><td>S2</td><td>Comparison number 2</td><td colspan="2">Data source or data variable unit 2 to be compared</td></tr>
</table>

Note: The ※ number is one of =, >, <, <>, >=, <=.

2. Operand

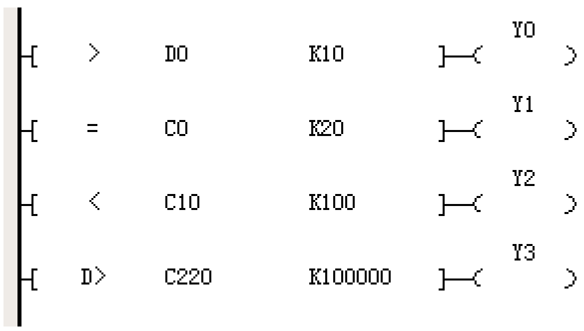
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

LD contact type comparison instruction:

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
LD=	LDD=	S1=S2	S1≠S2
LD>	LDD>	S1>S2	S1<=S2
LD<	LDD<	S1<S2	S1>=S2
LD<>	LDD<>	S1<>S2	S1=S2
LD<=	LDD<=	S1<=S2	S1>S2
LD>=	LDD>=	S1>=S2	S1<S2

【Example】



When the value of D0 is greater than 10, Y0 is ON.

When the value of C0 is equal to 20, Y1 is ON.

When the value of C10 is less than 100, Y2 is ON.

When the value of C220 is greater than 100000, Y3 is ON.

Note: The comparison of 32-bit counters (C200 to C255) must be performed with 32 bits (LDD=, LDD>, LDD<, etc.).

If a 16-bit operation (LD=, LD>, LD<, etc.) is specified, a program error or an operation error will occur.

AND※Data comparison

1. Instruction form

The two operands are compared, the comparison result is output in a logical state, and the variables participating in the comparison are processed as signed numbers.

AND※ S1 S2		Contact type data comparison	Instruction execution	
S1	Comparison number 1	Data source or data variable unit 1 to be compared	16-bit instruction (5step) AND== Continuous execution	32-bit instruction (9step) ANDD= Continuous execution
S2	Comparison number 2	Data source or data variable unit 2 to be compared		

Note: The ※ number is one of =, >, <, <>, >=, <=.

2. Operand

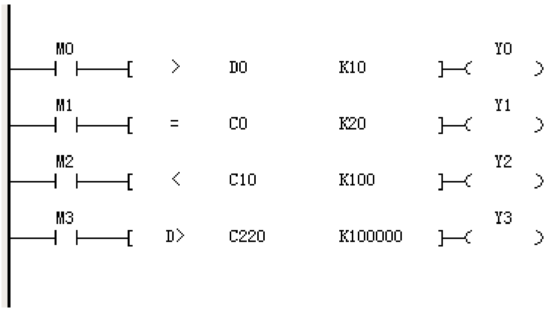
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

AND Contact type comparison instruction：

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
AND=	ANDD=	S1=S2	S1≠S2
AND>	ANDD>	S1>S2	S1<=S2
AND<	ANDD<	S1<S2	S1>=S2
AND<>	ANDD<>	S1<>S2	S1=S2
AND<=	ANDD<=	S1<=S2	S1>S2
AND>=	ANDD>=	S1>=S2	S1<S2

【Example】



When M0 is ON and the value of D0 is greater than 10, Y0 is ON.

When M1 is ON and the value of C0 is equal to 20, Y1 is ON.

When M2 is ON and the value of C10 is less than 100, Y2 is ON.

When M3 is ON and the value of C220 is greater than 100000, Y3 is ON.

Note: The comparison of 32-bit counters (C200 to C255) must be performed with 32 bits (ANDD=, ANDD>, ANDD<, etc.). If a 16-bit operation (AND=, AND>, AND<, etc.) is specified, a program error or an operation error will occur.

OR※Data comparison

1. Instruction form

The two operands are compared, the comparison result is output in a logical state, and the variables participating in the comparison are processed as signed numbers.

OR※ S1 S2			Comparison of parallel contact type data	Instruction execution	
S1	Comparison number 1	Data source or data variable unit 1 to be compared		16-bit instruction (5step)) OR= Continuous execution	32-bit instruction (9step) ORD= Continuous execution
S2	Comparison number 2	Data source or data variable unit 2 to be compared			

Note: The ※ number is one of =, >, <, <>, >=, <=.

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

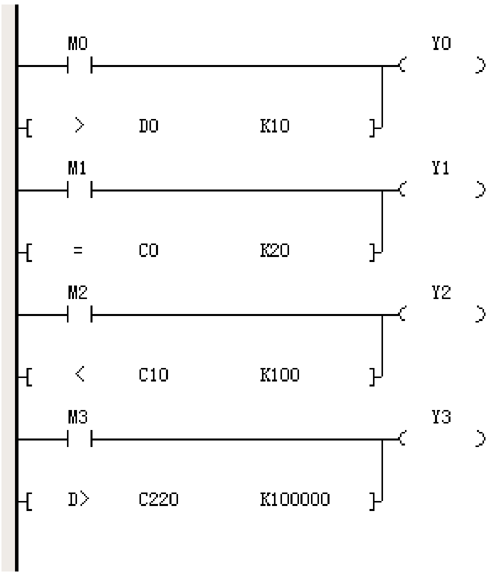
Note: With gray shading device, it means support.

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
OR=	ORD=	S1=S2	S1≠S2
OR>	ORD>	S1>S2	S1<=S2

OR contact type comparison instruction:：

OR<	ORD<	S1<S2	S1>=S2
OR<>	ORD<>	S1<>S2	S1=S2
OR<=	ORD<=	S1<=S2	S1>S2
OR>=	ORD>=	S1>=S2	S1<S2

【Example】



When M0 is ON or the value of D0 is greater than 10, Y0 is ON.

When M1 is ON or the value of C0 is equal to 20, Y1 is ON.

When M2 is ON or the value of C10 is less than 100, Y2 is ON.

When M3 is ON or the value of C220 is greater than 100000, Y3 is ON.

Note: The comparison of 32-bit counters (C200 to C255) must be performed with 32 bits (ORD=, ORD>, ORD<, etc.). If a 16-bit operation (OR=, OR>, OR<, etc.) is specified, a program error or an operation error will occur.

LD※Contact status bit operation

1. Instruction form

Bit logic operation result as the contact conduction state, the node directly connected to the left bus.

<table>
<tr><td colspan="3">LD※ S1 S2</td><td>Contact status bit operation</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S1</td><td>Data 1</td><td colspan="2">Source data 1 device number</td><td rowspan="2">16-bit instruction (5step)) LD※ Continuous execution</td><td rowspan="2">32-bit instruction (9step)) LDD※ Continuous execution</td></tr>
<tr><td>S2</td><td>Data 2</td><td colspan="2">Source data 2 device number</td></tr>
</table>

Note: ※ is one of &, |, ^.

2. Operand

Operand	Bit device							Word device															
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

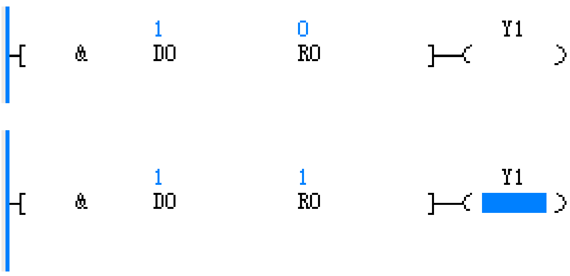
Note: With gray shading device, it means support.

Function description: [S1] and [S2] content logical operation ("and", "or |", "exclusive OR"), the result is not 0, the instruction is turned on; the comparison result is 0 This instruction does not turn on.

operation results, as follows：

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
LD&	LDD&	S1&S2≠0	S1&S2=0
LD|	LDD|	S1|S2≠0	S1|S2=0
LD^	LDD^	S1^S2≠0	S1^S2=0

【Example】



AND※Contact status bit operation

1. Instruction form

Bit logic operation result as the contact conduction state, nodes connected in series with other nodes。

AND※ S1 S2		Contact status bit operation	Instruction execution	
S1	Data 1	Source data 1 device number	16-bit instruction (5step)） AND※ Continuous execution	32-bit instruction (9step)） ANDD※ Continuous execution
S2	Data 2	Source data 2 device number		

Note: ※ is one of &, |, ^.

2. Operand

Operand	Bit device							Word device															
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

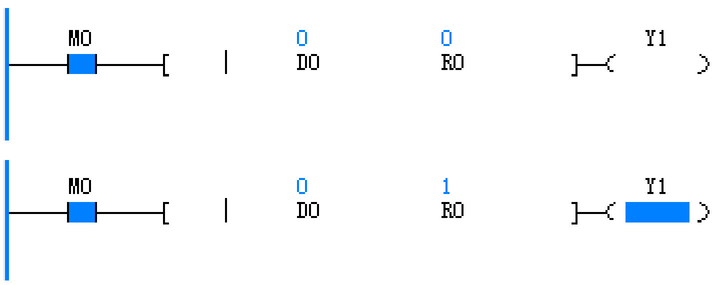
Note: With gray shading device, it means support.

Function description: [S1] and [S2] content logical operation ("and", "or |", "exclusive OR"), the result is not 0, the instruction is turned on; the comparison result is 0 This instruction does not turn on.

operation results, as follows：

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
AND&	ANDD&	S1&S2≠0	S1&S2=0
AND|	ANDD|	S1|S2≠0	S1|S2=0
AND^	ANDD^	S1^S2≠0	S1^S2=0

【Example】



OR※Contact status bit operation

1. Instruction form

Bit logic operation result as the contact conduction state, a node connected in parallel with other nodes.

OR※ S1 S2		Contact status bit operation	Instruction execution	
S1	Data 1	Source data 1 device number	16-bit instruction (5step)) OR※ Continuous execution	32-bit instruction (9step)) ORD※ Continuous execution
S2	Data 2	Source data 2 device number		

Note: ※ is one of &, |, ^

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

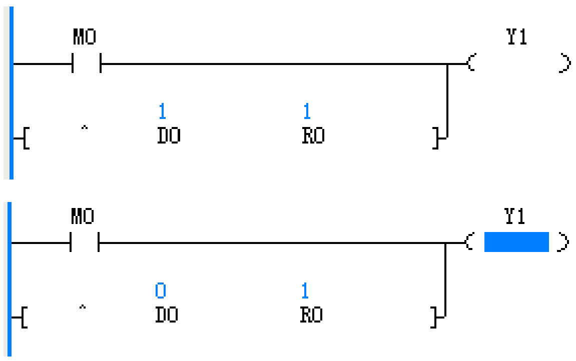
Function description: [S1] and [S2] content logical operation ("and", "or |", "exclusive OR"), the result is not 0, the instruction is turned on; the comparison result is 0 When the instruction does not turn on。

operation results, as follows：

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
OR&	ORD&	S1&S2≠0	S1&S2=0

OR|	ORD|	S1|S2≠0	S1|S2=0
OR^	ORD^	S1^S2≠0	S1^S2=0

【Example】



FLD※Floating number contact comparison

1. Instruction form

Compare the two operand sizes and turn the contact ON or OFF according to the comparison result, the node directly connected to the left bus.

FLD※ S1 S2		Floating number contact comparison	Instruction execution
S1	Data 1	Source data 1 device number	32-bit instruction (9step)） FLDD※ Continuous execution
S2	Data 2	Source data 2 device number	

Note: ※ is one of =, >, <, <>, >=, <=.

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

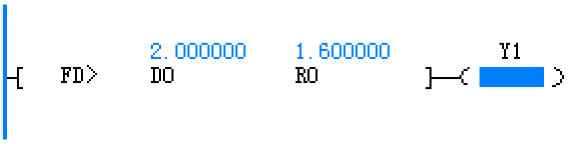
Note: With gray shading device, it means support.

Function Description:

[S1] and [S2] Comparative instructions. When the condition is met, it is turned on, otherwise it is not turned on.

32bit instruction	Continuity condition	Non-Continuity condition
FLDD>	S1>S2	S1<=S2
FLDD>=	S1>=S2	S1<S2
FLDD<	S1<S2	S1>=S2
FLDD<=	S1<=S2	S1>S2
FLDD=	S1=S2	S1<>S2

【Example】



FAND※Floating number [and] contact comparison

1. Instruction form

Compare the two operand sizes and turn the contact ON or OFF according to the comparison result, a node connected in series with other nodes.

FLD※ S1 S2			Floating number[and] contact comparison	Instruction execution
S1	Data 1	Source data 1 device number		32-bit instruction (9step)） FLDD※ Continuous execution
S2	Data 2	Source data 2 device number		

Note: ※ is one of =, >, <, <>, >=, <=

2. Operand

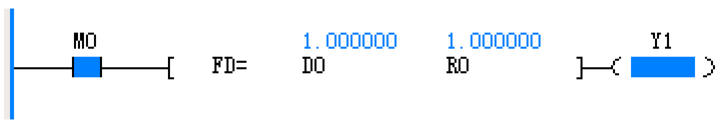
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description: [S1] and [S2] Comparative instructions. Turned on when the conditions are met, otherwise it is not conducting.

32bit instruction	Continuity condition	Non-Continuity condition
FANDD>	S1>S2	S1<=S2
FANDD>=	S1>=S2	S1<S2
FANDD<	S1<S2	S1>=S2
FANDD<=	S1<=S2	S1>S2
FANDD=	S1=S2	S1<>S2
FANDD<>	S1<>S2	S1=S2

【Example】



FOR※Floating number [or] contact comparison

1. Instruction form

Compare the size of the two operands, turn the contact ON or OFF according to the comparison result, and connect the nodes in parallel with other nodes.

FOR※ S1 S2		Floating number[or] contact comparison	Instruction execution
S1	Data 1	Source data 1 device number	32-bit instruction (9step)） FORD※ Continuous execution
S2	Data 2	Source data 2 device number	

Note: ※ is one of =, >, <, <>, >=, <=

2. Operand

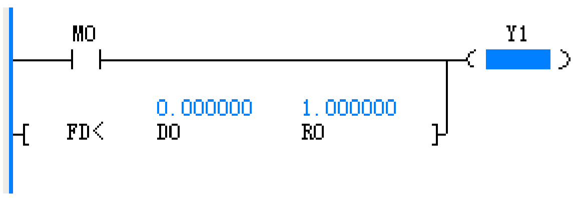
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description: [S1] and [S2] Comparative instructions. Turned on when the conditions are met, otherwise it is not conducting。

32bit instruction	Continuity condition	Non-Continuity condition
FORD>	S1>S2	S1<=S2
FORD>=	S1>=S2	S1<S2
FORD<	S1<S2	S1>=S2
FORD<=	S1<=S2	S1>S2
FORD=	S1=S2	S1<>S2
FORD<>	S1<>S2	S1=S2

【Example】



LDZ※Absolute value comparison contact

1. Instruction form

The absolute value of the result obtained by subtracting S1 from S2 is compared with the absolute value of S3, and the contact is turned ON or OFF according to the comparison result, and the node directly connected to the left bus is connected.

LDZ※ S1 S2 S3		Absolute value comparison contact	Instruction execution	
S1	Subtracted	Subtracted source component	16-bit instruction (5step) LDZ※ Continuous execution	32-bit instruction (9step) LDDZ※ Continuous execution
S2	Subtraction	Subtraction source component		
S3	Comparison value	Comparison value source component		

Note: The ※ number is one of =, >, <, <>, >=, <=.

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

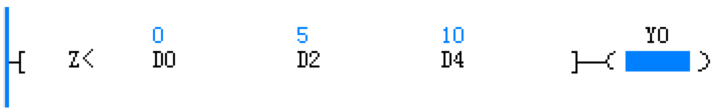
Note: With gray shading device, it means support.

Function Description:

The instruction is that the absolute value of the result of [S1] subtraction from [S2], compared with the absolute value of [S3]. Turns on when the condition is met, otherwise it does not turn on.

<table>
<tr><th>16bit instruction</th><th>32bit instruction</th><th>Continuity condition</th><th>Non-Continuity condition</th></tr>
<tr><td>LDZ></td><td>LDDZ></td><td>|S1-S2|>|S3|</td><td>|S1-S2|<=|S3|</td></tr>
<tr><td>LDZ>=</td><td>LDDZ>=</td><td>|S1-S2|>=|S3|</td><td>|S1-S2|<|S3|</td></tr>
<tr><td>LDZ<</td><td>LDDZ<</td><td>|S1-S2|<|S3|</td><td>|S1-S2|>=|S3|</td></tr>
<tr><td>LDZ<=</td><td>LDDZ<=</td><td>|S1-S2|<=|S3|</td><td>|S1-S2|>|S3|</td></tr>
<tr><td>LDZ=</td><td>LDDZ=</td><td>|S1-S2|=|S3|</td><td>|S1-S2|<>|S3|</td></tr>
<tr><td>LDZ<></td><td>LDDZ<></td><td>|S1-S2|<>|S3|</td><td>|S1-S2|=|S3|</td></tr>
</table>

【Example】



ANDZ※Absolute value comparison [and] contact

1. Instruction form

The absolute value of the result obtained by subtracting S1 from S2 is compared with the absolute value of S3, and the contact is turned ON or OFF according to the comparison result, and the node connected in series with other nodes is connected.

ANDZ※ S1 S2 S3		Absolute value comparison [and] contact	Instruction execution	
S1	Subtracted	Subtracted source component	16-bit instruction (5step)	32-bit instruction (9step)
S2	Subtraction	Subtraction source component		

S3	Comparison value	Comparison value source component	LDZ※ Continuous execution	LDDZ※ Continuous execution

Note: The ※ number is one of =, >, <, <>, >=, <=.

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

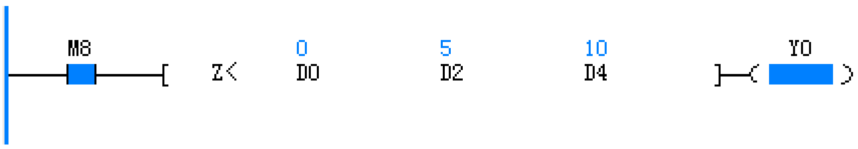
Note: With gray shading device, it means support.

Function Description:

The instruction is that the absolute value of the result of [S1] subtraction from [S2], compared with the absolute value of [S3]. Turns on when the condition is met, otherwise it does not turn on.

16bit instruction	32bit instruction	Continuity condition	Non-Continuity condition
ANDZ>	ANDDZ>	|S1-S2|>|S3|	|S1-S2|<=|S3|
ANDZ>=	ANDDZ>=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ANDZ<	ANDDZ<	|S1-S2|<|S3|	|S1-S2|>=|S3|
ANDZ<=	ANDDZ<=	|S1-S2|<=|S3|	|S1-S2|>|S3|
ANDZ=	ANDDZ=	|S1-S2|=|S3|	|S1-S2|<>|S3|
ANDZ<>	ANDDZ<>	|S1-S2|<>|S3|	|S1-S2|=|S3|

【Example】



ORZ※Absolute value comparison [or] contact

1. Instruction form

The absolute value of the result obtained by subtracting S1 from S2 is compared with the absolute value of S3, and the contact is turned ON or OFF according to the comparison result, and the node connected in parallel with other nodes.。

Note: The ※ number is one of =, >, <, <>, >=, <=.

ORZ※ S1 S2 S3		Absolute value comparison [or] contact	Instruction execution	
S1	Subtracted	Subtracted source component	16-bit instruction (5step) LDZ※ Continuous execution	32-bit instruction (9step) LDDZ※ Continuous execution
S2	Subtraction	Subtraction source component		
S3	Comparison value	Comparison value source component		

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

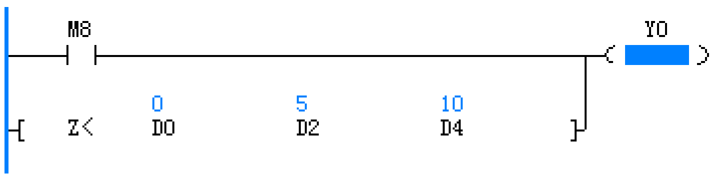
Note: With gray shading device, it means support.

Function Description:

The instruction is that the absolute value of the result of [S1] subtraction from [S2], compared with the absolute value of [S3]. Turns on when the condition is met, otherwise it does not turn on.

<table>
<tr><th>16bit instruction</th><th>32bit instruction</th><th>Continuity condition</th><th>Non-Continuity condition</th></tr>
<tr><td>ORZ></td><td>ORDZ></td><td>|S1-S2|>|S3|</td><td>|S1-S2|<=|S3|</td></tr>
<tr><td>ORZ>=</td><td>ORDZ>=</td><td>|S1-S2|>=|S3|</td><td>|S1-S2|<|S3|</td></tr>
<tr><td>ORZ<</td><td>ORDZ<</td><td>|S1-S2|<|S3|</td><td>|S1-S2|>=|S3|</td></tr>
<tr><td>ORZ<=</td><td>ORDZ<=</td><td>|S1-S2|<=|S3|</td><td>|S1-S2|>|S3|</td></tr>
<tr><td>ORZ=</td><td>ORDZ=</td><td>|S1-S2|=|S3|</td><td>|S1-S2|<>|S3|</td></tr>
<tr><td>ORZ<></td><td>ORDZ<></td><td>|S1-S2|<>|S3|</td><td>|S1-S2|=|S3|</td></tr>
</table>

【Example】



4.3.2 Output comparison

Output comparison instruction	
CMP	Data comparison
ECMP	Binary floating number comparison
ZCP	Regional comparison
EZCP	Compare binary floating number range

CMP Data comparison

1. Instruction form

When the driving condition is satisfied, the sizes of S1 and S2 are compared, and according to the comparison result (S1>S2, S1=S2, S1<S2), one of the address bits D, D+1, D+2 is set to be ON.

<table>
<tr><td colspan="3">CMP S1 S2 D</td><td>Compare the two numbers</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S1</td><td>Comparison value 1</td><td colspan="2">Comparison value 1 data or data storage word device address</td><td rowspan="2">16-bit instruction (7step) CMP</td><td rowspan="2">32-bit instruction (13step) DCMP</td></tr>
<tr><td>S2</td><td>Comparison</td><td colspan="2">Comparison value 2 data or data storage</td></tr>
</table>

	value 2	word device address	Continuous execution CMPP pulse execution	Continuous execution DCMPP pulse execution
D	Comparing results	Comparison result ON/OFF bit first address, occupying 3 consecutive bits		

2. Operand

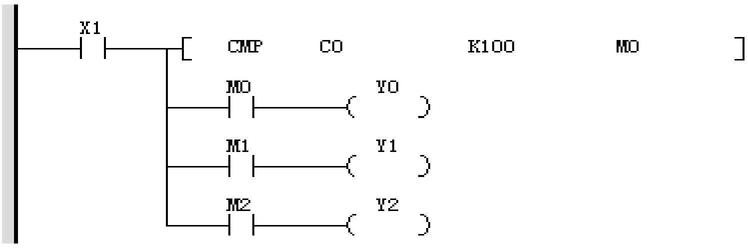
Operand	Bit device							Word device															
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

This instruction completes the comparison of the sizes of the two manipulated variables, and outputs the comparison result to the specified bit variable. The operands perform algebraic comparison operations on the signed number. D will occupy 3 consecutive address bit variables.

【Example】



When X1 is ON, C0 is greater than 100, M0 is ON, and Y0 has an output.

When X1 is ON, C0 is equal to 100, M1 is ON, and Y1 has an output.

When X1 is ON, C0 is less than 100, M2 is ON, and Y2 has an output.

When the X1 turns on to off, the CMP command is not executed. M0~M2 still maintain the state before X0=OFF. To clear the comparison result of M0~M2, M0~M2 can be cleared by RST or ZRST.

If you need to get the results of ≥, ≤, ≠, you can get M0~M2 in series and parallel.。

ECMP Binary floating number comparison

1. Instruction form

The comparison of the two floating number variables is performed, and the result of the comparison is output to the three variables starting from D.

ECMP S1 S2 D			Binary floating number comparison	Instruction execution
S1	Comparison value 1	Comparison value 1 data or data storage word device address		32-bit instruction (13step) DECMP Continuous execution
S2	Comparison value 2	Comparison value 2 data or data storage word device address		
D	Comparing results	Comparison result storage unit, occupying 3 consecutive bits		

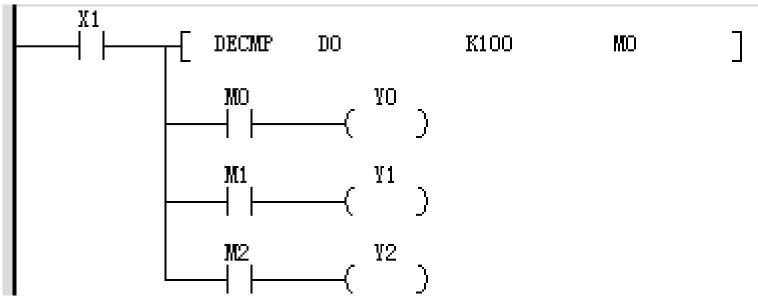
2. Operand

Operand	Bit device							Word device															
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function description: This instruction compares two floating number variables, and outputs the comparison result to the specified bit variable. The operands perform algebraic comparison operations on the signed number. D will occupy 3 bit variables of consecutive addresses.

【 Example】



When X1 is ON, (D1, D0) is greater than 100, M0 is ON, and Y0 has an output.

When X1 is ON, (D1, D0) is equal to 100, M1 is ON, and Y1 has an output.

When X1 is ON, (D1, D0) is less than 100, M2 is ON, and Y2 has an output.

When X1 turns from ON to OFF, the DECMP instruction is not executed, and M0~M2 still maintain the state before X0=OFF. To clear the comparison result of M0~M2, M0~M2 can be cleared by RST or ZRST.

If you need to get the results of ≥, ≤, ≠, you can get M0~M2 in series and parallel.

If S1 or S2 is a K or H constant, the system will automatically convert to a floating point number to participate in the operation.

ZCP Region comparison

1. Instruction form

When the driving condition is established, according to the interval where S is located (S<S1, S1≤S≤S2, S>S2), one of the final address elements D, D+1, D+2 is set to be ON.

ZCP S1 S2 S D		region comparison	Instruction execution	
S1	Regional comparison lower limit	Data or data storage word device address	16-bit instruction (9step) ZCP Continuous execution ZCPP pulse execution	32-bit instruction (17step) DZCP Continuous execution DZCPP pulse execution
S2	Regional comparison limit	Data or data storage word device address		
S	Comparison variable	Data or data storage word device address		
D	Comparing results	Comparison result ON/OFF bit first address, occupying 3 consecutive bits		

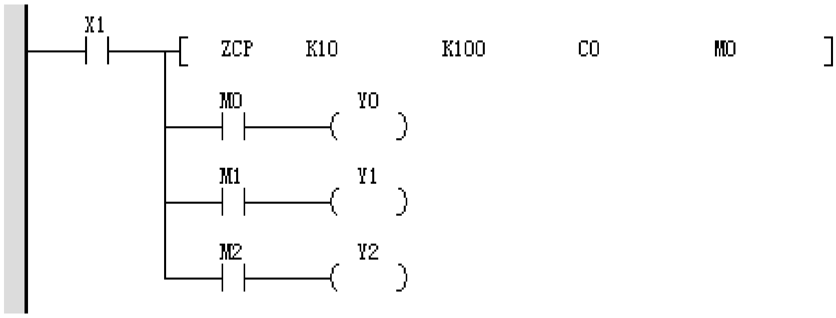
2. Operand

Operand	Bit devic							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function description: This instruction completes the comparison of the size of two operation variables, and outputs the comparison result to the specified bit variable. The operands perform algebraic comparison operations according to the signed number. D will occupy 3 consecutive address bit variables

【Example】



When X1 is ON, C0 is less than 10, M0 is ON, and Y0 has an output.

When X1 is ON, C0 is greater than or equal to 10 and less than or equal to 100, M1 is ON, and Y1 has an output.

When X1 is ON, C0 is greater than 100, M2 is ON, and Y2 has an output.

When X1 turns from ON to OFF, the ZCP command is not executed. M0~M2 still maintain the state before X0=OFF. To clear the comparison result of M0~M2, M0~M2 can be cleared by RST or ZRST.

EZCP Binary floating number region comparison

1. Instruction form

Perform a region comparison of binary floating number variables, and output the result of the comparison to the three variables starting from D.

EZCP S1 S2 S D			Binary floating number region comparison	Instruction execution
S1	Regional comparison lower limit	The lower limit of binary floating number variable region		32-bit instruction (17step) DEZCP Continuous execution DEZCPP pulse execution
S2	Regional comparison upper limit	The upper limit of binary floating number variable region		
S	Comparison variable	Binary floating number variable regional variable		

D	Comparing results	Comparison result ON/OFF bit first address, occupying 3 consecutive bits	

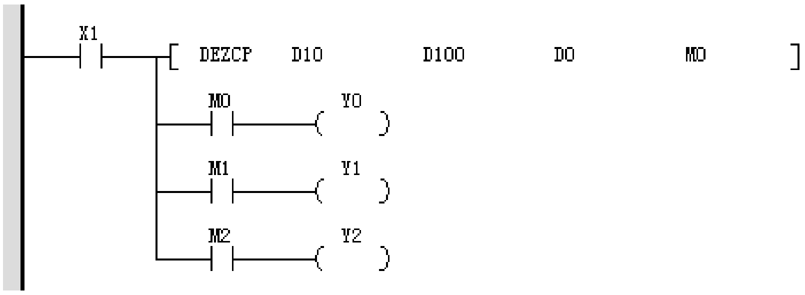
2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: with gray shading device, indicating support。

Function description: This instruction performs interval comparison of binary floating-point variables, and outputs the comparison result to the specified bit variable. The operands perform algebraic comparison operations according to the signed number. D will occupy 3 consecutive address bit variables

【Example】



When X1 is ON, (D1, D0) is less than (D11, D10), M0 is ON, and Y0 has an output.

When X1 is ON, (D1, D0) is greater than or equal to (D11, D10) and less than or equal to (D101, D100), M1 is ON, and Y1 has an output.

When X1 is ON, (D1, D0) is greater than (D101, D100), M2 is ON, and Y2 has an output.

When X1 turns OFF from ON, the DEZCP command is not executed, and M0~M2 still maintain the state before X0=OFF. To clear the comparison result of M0~M2, M0~M2 can be cleared by RST or ZRST.

4.4 Data operation

Data operation: four arithmetic operations; data logical operations; trigonometric functions; exponential operations.

Arithmetic	
ADD	Binary data addition
SUB	Binary data subtraction
MUL	Binary data multiplication
DIV	Binary data division
EADD	Binary floating number addition
ESUB	Binary floating number subtraction
EMUL	Binary floating number multiplication
EDIV	Binary floating number division
INC	Binary data plus one
DEC	Binary data minus one
Data logic operation	
WAND	Binary data logic and
WOR	Binary data logic or
WXOR	Binary data logical XOR
NEG	Binary data complement
ENEG	Binary floating number symbol negation
Trigonometric function	
SIN	Floating number SIN operation instruction
COS	Floating number COS operation instruction
TAN	Floating number TAN operation instruction
ASIN	Binary floating number ARCSIN operation
ACOS	Binary floating number ARCCOS operation
ATAN	Binary floating number ARCTAN operation
RAD	Binary floating number angle → radians conversion
DEG	Binary floating number radians → angle conversion
SINH	Binary floating number SINH operation
COSH	Binary floating number COSH operation

TANH	Binary floating number TANH operation
Exponential operation	
EXP	Binary floating number Exponential operation
LOGE	Binary floating number Natural logarithm operation
LOG	Binary floating number Logarithmic operation with a low of 10
ESQR	Binary floating number Square operation
SQR	Binary data square operation

4.4.1 Arithmetic

Arithmetic	
ADD	Binary data addition
SUB	Binary data subtraction
MUL	Binary data multiplication
DIV	Binary data division
EADD	Binary floating number addition
ESUB	Binary floating number subtraction
EMUL	Binary floating number multiplication
EDIV	Binary floating number division
INC	Binary data plus one
DEC	Binary data minus one

ADD Binary data addition

1. Instruction form

Binary addition instruction

ADD S1 S2 D		Binary data addition	Instruction execution	
S1	Summand	Data or data storage word device address	16-bit finger (7step) ADD Continuous execution ADDP pulse execution	32- bit instruction (13step) DADD Continuous execution DADDP pulse execution
S2	Addend	Data or data storage word device address		
D	sum	data storage word device address		

2. Operand

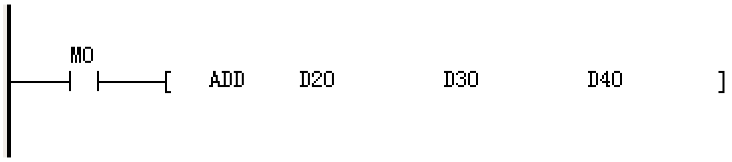
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Requires contact drive. There are 3 operation variables. The values of S1 and S2 are added by BIN algebra and stored in D. The variables involved in the operation are processed according to the signed number. The highest bit is the sign bit and 0 is the positive number., 1 is a negative number.
- If the result of the calculation is 0, the 0 flag (M8020) will be set;
- If the calculation result exceeds 32767 (16bit operation) or 2147483647 (32bit operation), the carry flag (M8022) will be set;
- If the calculation result is less than -32768 (16bit operation) or -2147483648 (32bit operation), the borrow flag (M8021) will be set;
- When performing 32-bit operation, the variable address in the instruction is the lower 16-bit address, and the adjacent high-numbered address unit is 16 bits higher. It prevents duplication or false coverage during programming.

【Example】



When M0 is ON, the content of D20 plus the content of D30 is placed in D40.

SUB Binary data subtraction

1. Instruction form

Binary subtraction instruction

SUB S1 S2 D			Binary data subtraction	Instruction execution	
S1	Minuend	Data or data storage word device address		16-bit finger (7step) SUB Continuous execution SUBP pulse execution	32- bit instruction (13step) DSUB Continuous execution DSUBP pulse execution
S2	Subtraction	Data or data storage word device address			
D	difference	data storage word device address			

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

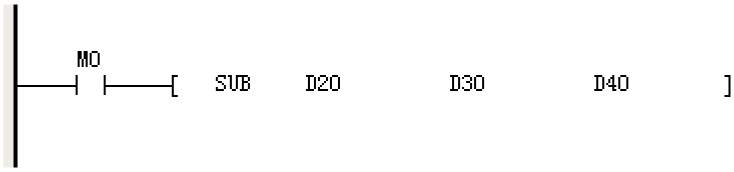
Function Description:

- Requires contact drive. There are 3 operation variables. The values of S1 and S2 are subtracted by BIN algebra and stored in D. The variables involved in the operation are processed according to the signed number. The highest bit is the sign bit and 0 is the positive number. , 1 is a negative number.
- If the result of the calculation is 0, the 0 flag (M8020) will be set;
- If the calculation result exceeds 32767 (16bit operation) or 2147483647 (32bit operation), the

carry flag (M8022) will be set;

- If the calculation result is less than -32768 (16bit operation) or -2147483648 (32bit operation), the borrow flag (M8021) will be set.
- When performing 32-bit operation, the variable address in the instruction is the lower 16-bit address, and the adjacent high-numbered address unit is 16 bits higher. It prevents duplication or false coverage during programming.

【Example】



When M0 is ON, the content of D20 minus the content of D30 is stored in D40.

MUL Binary data multiplication

1. Instruction form

Binary multiply instruction

MUL S1 S2 D		Binary data multiplication	Instruction execution	
S1	Multiplicand	Data or data storage word device address	16-bit finger (7step) MUL Continuous execution MULP pulse execution	32- bit instruction (13step) DMUL Continuous execution DMULP pulse execution
S2	multiplier	Data or data storage word device address		
D	product	Data memory word device address, 16-bit instruction product is 32-bit data, 32-bit instruction product is 64-bit data		

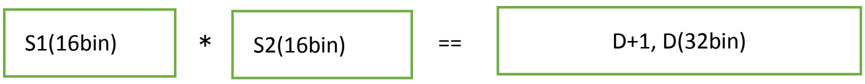
2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

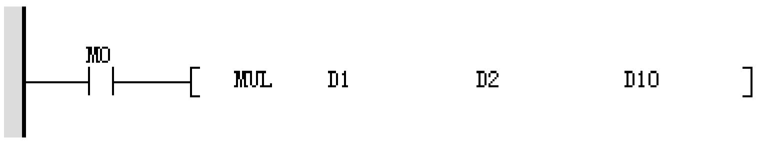
Function Description:

- Requires contact drive. There are 3 operation variables. The values of S1 and S2 are multiplied by BIN algebra and stored in D. The variables involved in the operation are processed according to the signed number. The highest bit is the sign bit and 0 is the positive number., 1 is a negative number.
- When performing 32-bit operation, the variable address bit in the instruction is the lower 16-bit address, and the adjacent high-numbered address unit is 16 bits higher. It prevents duplication or false coverage during programming; the result of the calculation can only be 32 bits. For calculations beyond the 32-bit range, the most It is good to use the floating-point arithmetic instruction EMUL for calculation.



S1(32bin) * S1(32bin) == D+3, D+2,D+1,D(64bin)

【Example】



When M0 is ON, the number in D1 is multiplied by D2 and stored in (D11, D10).

When the result is less than 16bin, it is stored in D10.

When the result is greater than 16bin, it is stored in D11, D10.

DIV Binary data division

1. Instruction form

Binary divide instruction

DIV S1 S2 D		Binary data division	Instruction execution	
S1	Dividend	Data or data storage word device address	16-bit instruction (7step) DIV Continuous execution DIVP pulse execution	32-bit instruction (13step) DDIV Continuous execution DDIVP pulse execution
S2	divisor	Data or data storage word device address		
D	Quotient, remainder	data storage word device address, The quotient is stored in D and the remainder is stored in D+1.		

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

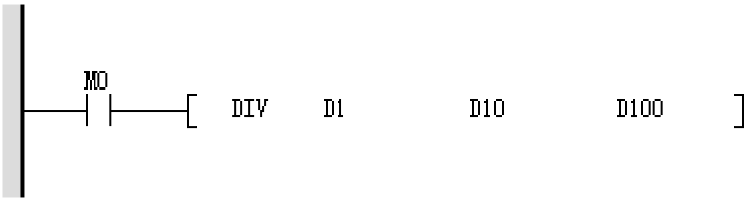
- Requires contact drive. There are 3 operation variables. The values of S1 and S2 are divided into BIN algebra and stored in D. The variables involved in the operation are processed according to the signed number. The highest bit is the sign bit and 0 is the positive number., 1 is a negative number.
- When performing 32-bit operation, the S1 and S2 variable addresses in the instruction are low 16-bit addresses, and the adjacent high-numbered address units are 16 bits high. Prevent duplication or false coverage during programming. The calculated quotient is stored in the unit indicated by D and D+1. The remainder is stored in the D+2, D+3 address unit.
- If the divisor S2 is 0, a calculation error will occur.
- If the bit element (KnY/KnM/KnS) is specified as D, the remainder cannot be obtained.

■ If the dividend is negative, the remainder is negative.

S1(16bin) / S2(16bin) == Quotient D(16bin) Remainder D+1(16bin)

S1(32bin) / S1(32bin) == Quotient Remainder D+3, D+2(32bin)

【Example】



When M0 is ON, the number in D1 is divided by the number in D10, the quotient is stored in D100, and the remainder is stored in D101.

EADD Binary floating number addition

1. Instruction form

Binary floating number addition

EADD S1 S2 D			Binary floating number addition	Instruction execution
S1	Summand	Binary floating number summand		32- bit instruction (13step) DEADD Continuous execution DEADDP pulse execution
S2	Addend	Binary floating number addend		
D	sum	Addition sum storage unit		

2. Operand

Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

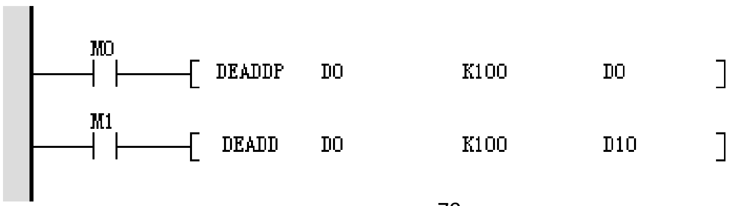
■ If the S1 or S2 source operand is a constant K or H, the constant is automatically converted to a binary floating point value for addition;

■ If the result of the calculation is 0, the 0 flag (M8020) will be set;

■ If the absolute value calculation result is greater than the maximum representable floating-point value, the carry flag (M8022 is) bit is set.

■ If the absolute value calculation result is smaller than the minimum representable floating-point value, then the borrow flag (M8021) is set.

【Example】



When M0 is ON, the value of D1 and D0 is added to 100 and stored in D1, D0. The constant K100 is automatically adjusted to a binary floating point number before the operation.

The storage unit of the sum can be the same unit as the addend or the addend. In this case, use the pulse execution type instruction DEADDP. Otherwise, if the Continuous execution instruction is used, the calculation will be executed once every time the program is scanned.

When M1 is ON, the value of D1 and D0 is added to 100 and stored in D11 and D10.

ESUB Binary floating number subtraction

1. Instruction form

Binary floating number subtraction

ESUB S1 S2 D		Binary floating number subtraction	Instruction execution
S1	Minuend	Binary floating number Minuend	32-bit instruction (13step) DESUB Continuous execution DESUBP pulse execution
S2	Subtracted	Binary floating number Subtracted	
D	Difference	Difference storage unit	

2. Operand

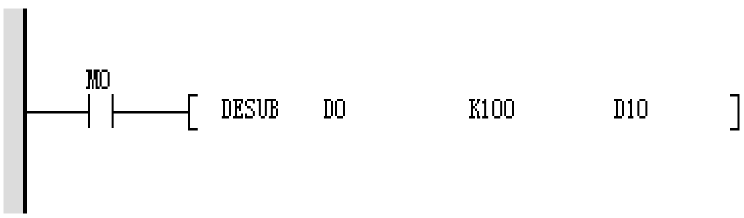
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- If the S1 or S2 source operand is a constant K or H, the constant is automatically converted to a binary floating point value for addition;
- If the result of the calculation is 0, the 0 flag (M8020) will be set;
- If the absolute value calculation result is greater than the maximum representable floating-point value, the carry flag (M8022 is) bit is set.
- If the absolute value calculation result is smaller than the minimum representable floating-point value, the flag (M8021) will be set.

【Example】



When M0 is ON, the contents of D1 and D0 are stored in D11 and D10 after subtracting 100.

EMUL Binary floating number multiplication

1. Instruction form

Binary floating number multiplication

EMUL S1 S2 D		Binary floating number multiplication	Instruction execution
S1	Multiplicand	Binary floating number Multiplicand	32- bit instruction (13step) DEMUL Continuous execution DEMULP pulse execution
S2	multiplier	Binary floating number multiplier	
D	product	Product Storage unit	

2. Operand

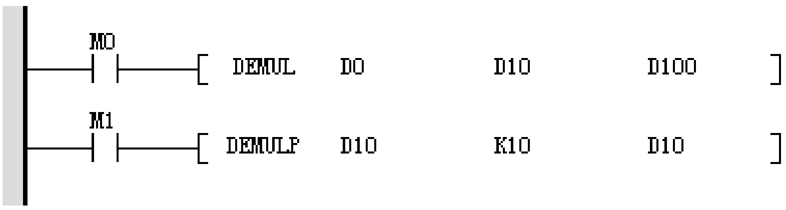
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- If the S1 or S2 source operand is a constant K or H, it will automatically convert the constant into a binary floating point value for multiplication;
- If the result of the calculation is zero, the 0 flag (M8020) will be set.
- If the absolute value calculation result is greater than the maximum representable floating-point value, the carry flag (M8022 is) bit is set.
- If the absolute value calculation result is smaller than the minimum representable floating-point value, then the borrow flag (M8021) is set.

【Example】



When M0 is ON, the number in D1, D0 is multiplied by D11, and the number in D10 is stored in D101, D100.

When M1 is ON, the number in D11, D10 is multiplied by 10 and stored in D11, D10.

The constant 10 is automatically adjusted to a binary floating point number before the operation.

The storage unit of the product can be the same unit as the multiplier or the multiplicand. In this case, use the pulse execution type instruction DEMULP. Otherwise, if the instruction is executed continuously, the calculation will be executed once every time the program is scanned.

EDIV Binary floating number division

1. Instruction form

Binary floating number division

EDIV S1 S2 D			Binary floating number division	Instruction execution
S1	Dividend	Binary floating number Dividend		32- bit instruction (13step) DEDIV Continuous execution DEDIVP pulse execution
S2	Divisor	Binary floating number Divisor		
D	Quotient	Binary floating number division store storage unit start address		

2. Operand

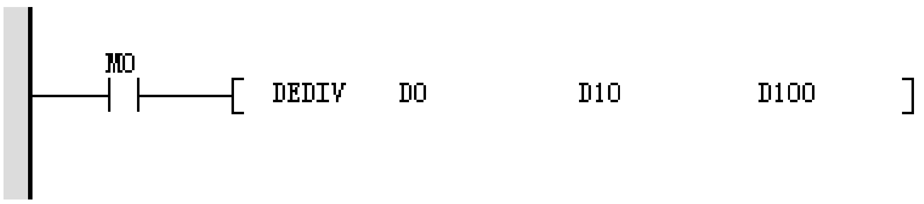
Operand	Bit device							Word device														
	System • User							System • User					Digit designationn					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- If the S1 or S2 source operand is a constant K or H, the constant is automatically converted to a binary floating point value for division;
- If the result of the calculation is zero, the 0 flag (M8020) will be set.
- If the absolute value calculation result is greater than the maximum representable floating-point value, the carry flag (M8022 is) bit is set.
- If the absolute value calculation result is smaller than the minimum representable floating-point value, then the borrow flag (M8021) is set.
- The divisor must not be 0. Otherwise, the calculation error will occur, and M8067 and M8068 will turn ON.

【Example】



When M0 is ON, the binary floating number (D1, D0) is divided by the binary floating point number (D11, D10), and the binary floating number quotient is stored in (D101, D100).

INC Binary data plus one

1. Instruction form

Binary plus one instruction

INC D			Binary plus one instruction	Instruction execution	
D	Cumulative result	data storage Word device address		16- bit instruction (3step) INC Continuous execution INCP Pulse execution	32- bit instruction (5step) DINC Continuous execution DINCP Pulse execution

2. Operand

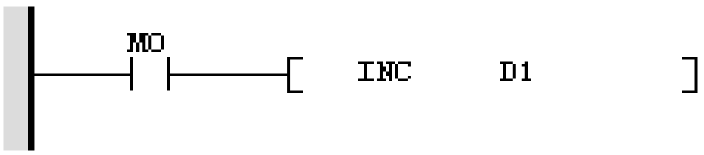
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Each time the instruction, a value of D is increased.
- 16-bit operation, 32767 plus 1 becomes -32768;
- For 32-bit operations, 2147483647 plus 1 becomes -2147483648.
- This instruction does not refresh the 0 flag, carry, and borrow flag.
- When 32bit operation performed, the instruction address of the variable D 16bit address bit is low, the high number of adjacent cells is high 16bit address, or a duplicate preventing erroneous programming covered.

【Example】



When MO is ON once, the value of D1 is increased by 1.

DEC Binary data minus one

1. Instruction form

Binary minus one instruction

DEC D			Binary data minus one	Instruction execution	
D	Reducing result	data storage Word device address		16- bit instruction (3step) DEC Continuous execution DECP Pulse execution	32- bit instruction (5step) DDEC Continuous execution DDECP Pulse execution

2. Operand

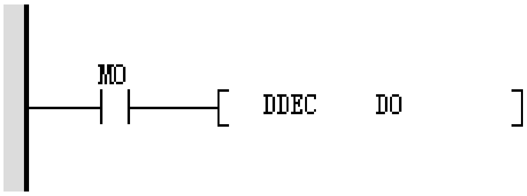
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Each time the instruction, the value D is decremented by 1.
- 16-bit operation, -32768 minus 1 to 32767;
- When 32-bit is operation, -2147483648 are further reduced to 1 and 2147483647.
- This instruction does not refresh the 0 flag, carry, and borrow flag.
- When 32bit operation performed, the instruction address of the variable D 16bit address bit is low, the high number of adjacent cells is high 16bit address, or a duplicate preventing erroneous programming covered.

【Example】



When MO is ON once, the value of D1, D0 is decremented by 1.

4.4.2 Data logic operation

Data logic　operation	
WAND	Binary data logic and
WOR	Binary data logic or
WXOR	Binary data logical XOR
NEG	Binary data complement
ENEG	Binary floating point symbol negation

WAND Binary data logic and

1. Instruction form

When the driving condition is satisfied, S1 and S2 are logically ANDed by bit, and the result is stored in D.

WAND　S1　S2　D			Binary data logic and	Instruction execution	
S1	Data 1	Data or　data storage　participating in the operation Word device　address		16　bit instruction （7step）	32　bit instruction （13step）
S2	Data 2	Data or　data storage　participating in the operation Word device　address		WAND Continuous execution	DAND Continuous execution
D	operation result	operation result　data storage　Word device　address		WANDP Pulse execution	DANDP Pulse execution

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

In this Instruction execution, each bit of the BIN value in S1 and S2 is referred to as a "logical AND" operation, and the result is stored in the D variable. The rule of the logical AND operation is 0 for any result.

【Example】

When M0 is ON, D0 and D10 are performed in bin units and the result is stored in D100.

	D0(16bin)	D10（16bin）	D100（16bin）
Bit Logic and operation	0	0	0
	0	1	0
	1	0	0
	1	1	1

WOR Binary data logic or

1. Instruction form

When the driving condition is satisfied, S1 and S2 bitwise logical OR operation, and stores the result in D.

WOR S1 S2 D			Binary data logic or	Instruction execution	
S1	Data 1	OR operation data or data storage Word device address		16 bit instruction（7step）	32 bit instruction（13step）
S2	Data 2	OR operation data or data storage Word device address		WOR Continuous execution	DOR Continuous execution
D	operation result	operation result data storage Word device address		WORP Pulse execution	DORP Pulse execution

2. Operand

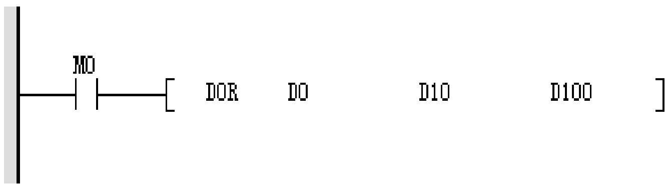
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

In this Instruction execution, each bit of the BIN value in S1 and S2 is referred to as a "logical OR” operation, and the result is stored in the D variable. The rule of the logical 'OR’ operation is either 1 and the result is 1.

【Example】



When M0 is ON, D1, D0 and D11, D10 are performed in units of bin or operation result is stored in D101, D100.

	D1,D0(32bin)	D11,D10（32bin）	D101,D100（32bin）
Bit Logical OR operation	0	0	0
	0	1	1
	1	0	1
	1	1	1

WXOR Binary data logic XOR

1. Instruction form

When the driving condition is satisfied, S1 and S2 are logically XORed in bits, and the result is stored in D.

WXOR S1 S2 D			Binary data logic XOR	Instruction execution	
S1	Data 1	XOR operation data or data storage Word device address		16 bit instruction（7step）WXOR Continuous execution WXORP Pulse execution	32 bit instruction（13step）DXOR Continuous execution DXORP Pulse execution
S2	Data 2	XOR operation data or data storage Word device address			
D	operation result	operation result data storage Word device address			

2. Operand

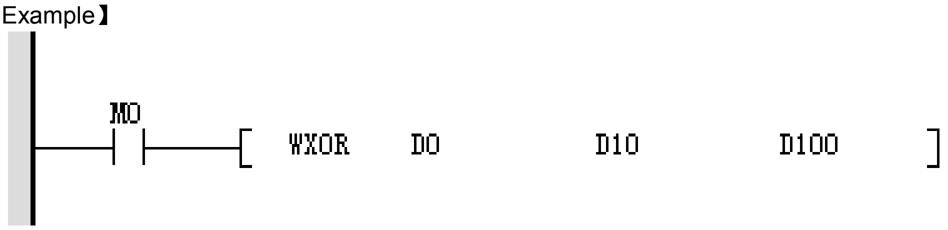
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

In this Instruction execution, each bit of the BIN value in S1 and S2 is referred to as a "logical XOR" operation, and the result is stored in the D variable. The rule of the logical 'XOR' (XOR) operation is 0 when the two numbers are the same, and is not 1.

【Example】



When M0 is ON, D0 and D10 are stored in D100 in XOR operation result.

	D0(16bin)	D10（16bin）	D100（16bin）
Bit Logical XOR operation	0	0	0
	0	1	1
	1	0	1
	1	1	0

NEG Binary dataComplement

1. Instruction form

When the driving condition is satisfied, the D bit by bit inverted, plus 1, and writes the result D

NEG D			Binary data Complement	Instruction execution	
D	operation result	data storage Word device address		16 bit instruction（3step） NEG Continuous execution NEGP Pulse execution	32 bit instruction（5step） DNEG Continuous execution DNEGP Pulse execution

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

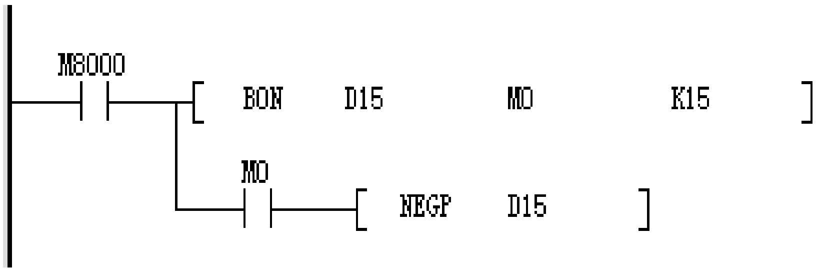
Note: With gray shading device, it means support.

Function Description:

- Requires contact drive with 1 manipulated variable. Invert the value of D bit by bit, add 1 to it, and save it back to D. This instruction generally uses the Pulse execution type instruction.
- Use the NEG instruction to make negative numbers absolute.

【Example】

NEG instruction makes negative numbers absolute：



When the 15th bit (bin0-bin15) of D15 is 1 (indicating that D15 is negative), M0 is ON. When M0 is ON, the absolute value of negative D15 can be obtained for D15 Complement.

ENEG Binary floating number Negate

1. Instruction form

Binary floating number (Real number) symbol Negate instruction。

ENEG S/D			Binary floating number symbol Negate	Instruction execution
S/D	Operand	The start number of the device holding the Binary floating number data for which symbol flipping is to be performed		32 bit instruction（5step） DENEG Continuous execution DENEGP Pulse execution

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

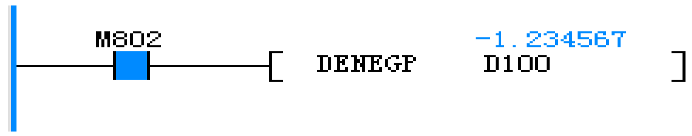
Function Description:

The sign of the binary floating point number of [D+1, D] is Negate, and the result is stored in [D+1, D]. Generally, use pulse type instructions.

【Example】

Before Instruction Execution

After Instruction execute



4.4.3 Trigonometric function

Trigonometric function	
SIN	Floating numberSIN operation instruction
COS	Floating numberCOS operation instruction
TAN	Floating numberTAN operation instruction
ASIN	Binary floating numberARCSIN operation
ACOS	Binary floating numberARCCOS operation
ATAN	Binary floating numberARCTAN operation
RAD	Binary floating number Angle → radians conversion
DEG	Binary floating number Radian → angle conversion
SINH	Binary floating numberSINH operation
COSH	Binary floating numberCOSH operation
TANH	Binary floating numberTANH operation

SIN Floating number SIN operation

1. Instruction form

Find the SIN (sinusoidal) value of the specified angle (RAD, radians), and the variable is in binary floating point storage format.

SIN S D		Floating number SIN operation	Instruction execution
S	data source	The angle variable to be sinusoidal, RAD unit, expressed as Binary floating number. The value range is 0 ≤ α ≤ 2π	32 bit instruction（9step） DSIN Continuous execution DSINP Pulse execution
D	operation result	The storage unit of the SIN calculation result after the transformation, in the Binary floating number format.	

2. Operand

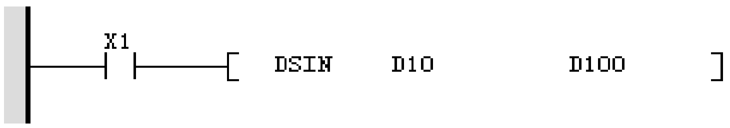
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is a SIN (sinusoidal) value for the specified angle (RAD, radians) and the variable is in binary floating point storage format.
- S is the angular variable of the sine to be obtained, RAD unit, expressed as Binary floating number.
- D is the storage unit of the converted SIN calculation result, Binary floating number format.
- RAD (radian) value = angle × π / 180 °.

【Example】



When X1 is ON, the radians (D11, D10) are obtained as SIN values and stored in (D101, D100).

The source data and SIN results calculated here are in Binary floating number format.

COS Floating number COS operation

1. Instruction form

Find the COS (cosine) value of the specified angle (RAD, radians), and the variable is in binary floating point storage format.

COS S D		Floating numberCOS operation	Instruction execution
S	data source	The angular variable of the cosine value to be sought, RAD unit, expressed as Binary floating number. The value range is 0 ≤ α ≤ 2π	32 bit instruction（9step） DCOS Continuous execution DCOSP Pulse execution
D	operation result	The storage unit of the transformed COS calculation result, Binary floating number format.	

2. Operand

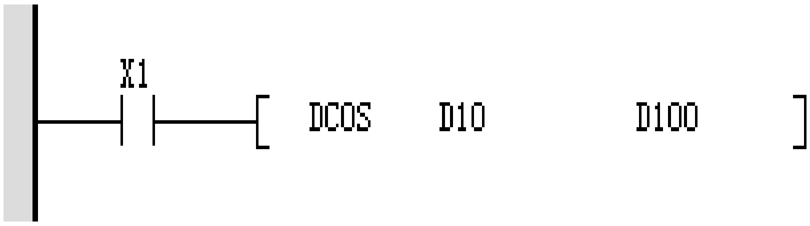
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is the COS (cosine) value of the specified angle (RAD, radians), and the variable is in binary floating point storage format.
- S is the angle variable of the cosine value to be sought, RAD unit, expressed as Binary floating number.
- D is the storage unit of the transformed COS calculation result, Binary floating number format.
- RAD (radian) value = angle × π / 180

【Example】



When X1 is ON, the radiance (D11, D10) is obtained as the COS value and stored in (D101, D100). The source data and COS result calculated here are in Binary floating number format.

TAN Floating number TAN operation

1. Instruction form

Find the TAN (tangent) value of the specified angle (RAD, radians), and the variable is in binary floating point storage format.

TAN S D		Floating numberTAN operation	Instruction execution
S	data source	The angular variable of the tangent to be obtained, RAD unit, expressed as Binary floating number. The value range is 0 ≤ α ≤ 2π	32 bit instruction（9step） DTAN Continuous execution DTANP Pulse execution
D	operation result	The storage unit of the converted TAN calculation result, Binary floating number format.	

2. Operand

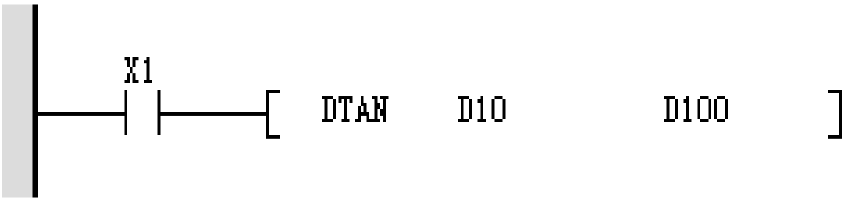
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is the TAN (tangent) value for the specified angle (RAD, radians) and the variable is in binary floating point storage format.
- S is the angular variable of the tangent to be determined, RAD unit, expressed as Binary floating number.
- D is the storage unit of the converted TAN calculation result, Binary floating number format.
- RAD (radian) value = angle × π / 180 °.

【Example】



When X1 is ON, the radians (D11, D10) are calculated as TAN values and stored in (D101, D100). The source data and TAN results calculated here are in Binary floating number format.

ASIN Binary floating number ARCSIN operation

1. Instruction form

Find the corresponding radians from the SIN value

ASIN S D			Binary floating numberARCSIN operation	Instruction execution
S	data source	Store the Binary floating number variable of the ARCSIN (anti-sinusoid) to be sought		32 bit instruction（9step） DASIN Continuous execution DASINP Pulse execution
D	operation result	Storage unit for calculating the result (-л/2 to л/2)		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

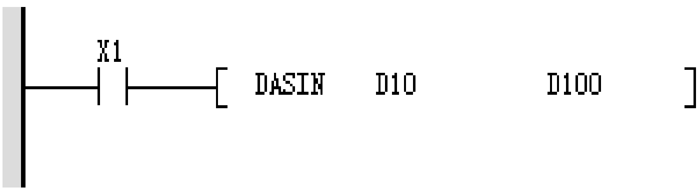
Note: With gray shading device, it means support.

Function Description:

This command is an operation that finds the corresponding radians from the SIN value.

Note: An operation error will occur when the value in S is not in the range of -1.0 to 1.0. The error code is K6706, K6706 will be saved in D8067, and error flag M8067 will be turned ON.

【Example】



When X1 is ON, the arc (D11, D10) is subjected to the SIN-1 budget and stored in (D101, D100).

ACOS Binary floating number ARCCOS operation

1. Instruction form

Find the corresponding radians from the COS value

ACOS S D		Binary floating number ARCCOS operation	Instruction execution
S	data source	Store the Binary floating number variable of the ARCCOS (anti-cosine) to be sought	32 bit instruction（9step） DACOS Continuous execution DACOSP Pulse execution
D	operation result	Calculate the result (0~л) of the storage unit	

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

This command is an operation that finds the corresponding radians from the COS value.

Note: An operation error will occur when the value in S is not in the range of -1.0 to 1.0. The error code is K6706, K6706 will be saved in D8067, error flag M8067 will be turned ON.

【Example】

When X1 is ON, the radians (D11, D10) are budgeted for COS-1 and stored in (D101, D100).

ATAN Binary floating number ARCTAN operation

1. Instruction form

Find the corresponding radians from the TAN value.

ATAN S D			Binary floating number ARCTAN operation	Instruction execution
S	data source	Store the Binary floating number variable of the ARCTAN (arctangent)		32 bit instruction（9step） DATAN Continuous execution DATANP Pulse execution
D	operation result	Calculated result (-л/2~л/2) memory cell		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

This command is an operation that finds the corresponding radians from the TAN value.

【Example】

When X1 is ON, the radians (D11, D10) are subjected to the TAN-1 budget and stored in (D101, D100).

RAD Binary floating number Angle → radians conversion

1. Instruction form

The operation of converting the Binary floating number angle into radians.

Its calculation formula is (radian unit = angle unit × π / 180)

RAD S D			Binary floating number Angle → radians conversion	Instruction execution
S	data source	Binary floating number angle variable for storing the arc to be determined		32 bit instruction（9step） DRAD Continuous execution DRADP Pulse execution
D	operation result	Storage unit for calculation results		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

This command is an operation to convert the angle of the Binary floating number into radians.

【Example】

When X1 is ON, the Binary floating number value in (D11, D10) is stored in angle (D101, D100) after the operation is angled to radians.

DEG Binary floating numberRadian → angle conversion

1. Instruction form

The operation of converting the Binary floating number radians into angles.

Its calculation formula is (angle unit = radians unit × π / 180)

DEG S D			Binary floating number Radian → angle conversion	Instruction execution
S	data source	Binary floating number radians variable storing the angle to be sought		32 bit instruction（9step） DDEG Continuous execution DDEGP Pulse execution
D	operation result	Storage unit for calculation results		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

The instruction is to perform a Binary floating number radians to an angle.

【Example】

When X1 is ON, the Binary floating number values in (D11, D10) are saved in radians to angles and saved to (D101, D100).

SINH Floating number SINH operation

1. Instruction form

Perform the Binary floating number to take the SINH value.

Its calculation formula is sinh(s)=(es-e-s)/2.

SINH S D			Floating number SINH operation	Instruction execution
S	data source	Binary floating number variable storing the SINH value to be sought		32 bit instruction (9step) DSINH Continuous execution DSINHP Pulse execution
D	operation result	The storage unit of the calculation result (when the operation result exceeds the floating number range, an error is reported 6706)		

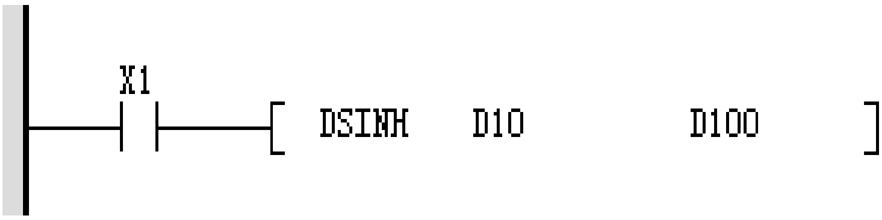
2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description: The instruction is to perform the Binary floating number to take the SINH value.

【Example】



When X1 is ON, the Binary floating number value in (D11, D10) is obtained as the SINH value and stored in (D101, D100).

COSH Floating number COSH operation

1. Instruction form

Perform a Binary floating number to take the COSH value.

Its calculation formula is cosh(s)=(es+e-s)/2

COSH S D		Floating number COSH operation	Instruction execution
S	data source	Binary floating number variable storing the COSH value to be sought	32 bit instruction (9step) DCOSH Continuous execution DCOSHP Pulse execution
D	operation result	Storage unit for calculation results	

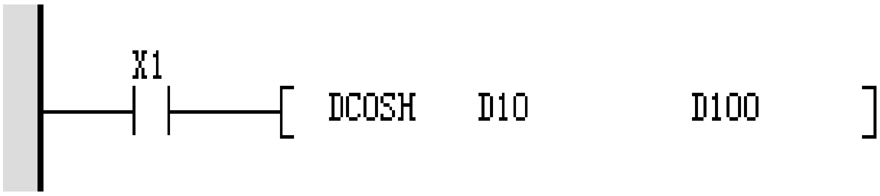
2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description: The instruction is to perform the Binary floating number to take the COSH value.

【Example】



When X1 is ON, the Binary floating number value in (D11, D10) is obtained by taking the COSH value and storing it in (D101, D100).

TANH Floating number TANH operation

1. Instruction form

Perform a Binary floating number to take the TANH value.

Its calculation formula is $\tanh(s)=(e^{s}-e^{-s})/(e^{s}+e^{-s})$.

TANH S D			Floating numberTANH operation	Instruction execution
S	data source	Binary floating number variable storing the TANH value to be sought		32 bit instruction （9step） DTANH Continuous execution DTANHP Pulse execution
D	operation result	Storage unit for calculation results		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description: The instruction is to take the Binary floating number and take the TANH value.

【Example】

When X1 is ON, the Binary floating number value in (D11, D10) is obtained as a TANH value and stored in (D101, D100).

4.4.4 Form operation

Trigonometric function	
WSUM	Calculate the total value of the data
MEAN	Average calculation
LIMIT	Upper and lower limit control
BZAND	Dead zone control
ZONE	Regional control

WSUM Calculate the total value of the data

1. Instruction form

This command can calculate the total value of consecutive 16-bit or 32-bit data.

WSUM S D n			Calculate the total value of the data	Instruction execution	
S	data source	The device start number of the data for which the total value is to be calculated		16 bit instruction (7step)	32 bit instruction (13step)
D	result	Device start number for saving total values		WSUM Continuous execution	DWSUM Continuous execution
n	Number of data	Number of data (n>0)		WSUMP Pulse execution	DWSUMP Pulse execution

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

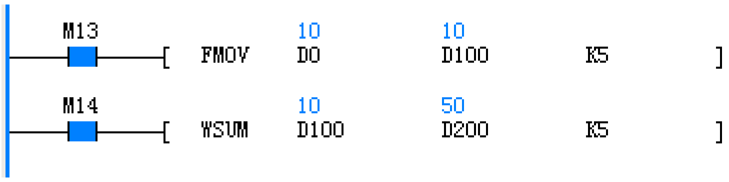
Function Description:

- 16 bit instruction: The sum of the n-point 16-bit data starting from [S], and the result is stored in [D+1, D] as 32-bit data.
- 32 bit instruction: The sum of the n-point 32-bit data starting from [S+1, S], and the result is stored in [D+3, D+2, D+1, D] in the form of 64-bit data.。

Some of the following conditions will report a failure. The error flag M8067 turns ON and the error code is saved in D8067.

- If the n-point device starting with [S] is beyond its legal range, report error 6705.
- When storing data, if [D] is beyond its legal range, report error 6705.
- Operandn's effective range: When n≤0, it is reported as error 6706.

【Example】



D100~D104 are assigned a value of 10, and M14 is turned ON, and the five D elements starting from D100 address are summed, and the result is stored in (D201, D200).

MEANAverage calculation（In development）

1. Instruction form

When the driving condition is established, the average of the K data starting with S is obtained, and the result is stored in D.

MEAN S D n		Average calculation instruction	Instruction execution	
S	Data first address	Average value of data storage Word device	16 bit instruction（7step）	32 bit instruction（13step）
D	average value	Average data storage Word device address	MEAN Continuous execution	DMEAN Continuous execution
n	Data length	Immediate, K=1～64	MEANP Pulse execution	DMEANP Pulse execution

2. Operand

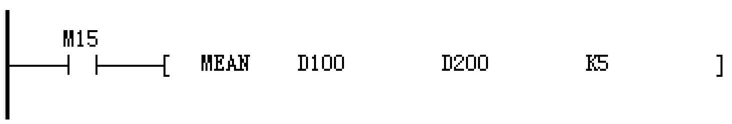
Operand	Bit device							Word device															
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Find the average of the N variables starting with S (first sum, divide by n) and store in D.
- If there is a remainder in the calculation, the remainder will be discarded;
- When the value of n is not in the range of 1 to 64, an error is calculated.

【Example】



For example：D100=9，D101=10，D102=11，D103=12，D104=15；Then D200=11, and the remainder is discarded.

（D100+D101+D102+D103+D104）/4=D20

LIMIT Upper and lower limit control

1. Instruction form

Set the upper/lower limit value of the input value, and then output the command

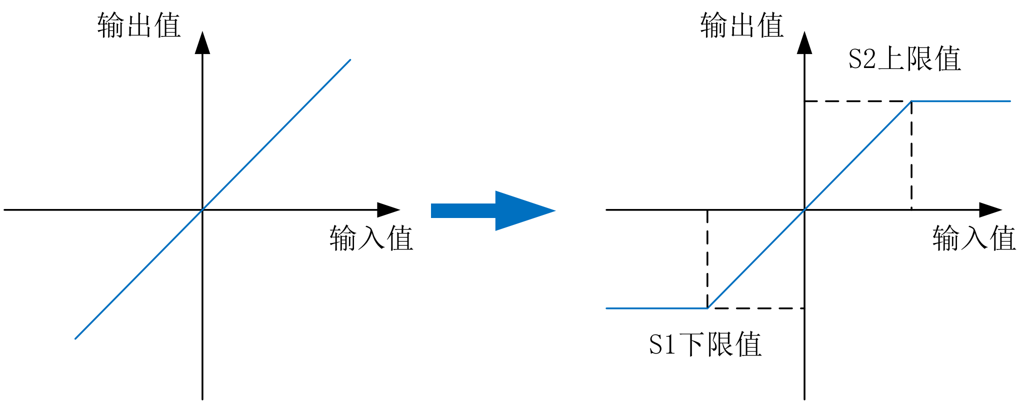
LIMIT S1 S2 S3 D		Upper and lower limit control	Instruction execution	
S1	lower limit	Lower limit value (minimum output limit value)	16 bit instruction （9step） LIMIT Continuous execution LIMITP Pulse execution	32 bit instruction （17step） DLIMIT Continuous execution DLIMITP Pulse execution
S2	Upper limit	Upper limit limit value (maximum output limit value)		
S3	input value	Input value that needs to be controlled by upper and lower limit		
D	output value	Save the device start number of the output value that has passed the upper and lower limit control		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:



- 16 bit instruction
  - By setting the upper and lower limits in [S1] and [S2], the output value [D] is output according to the input value [S3] within a certain range. When [S1]>[S3], [S1]→[D].
  - When [S2]<>[S3], [S2]→[D].
  - When [S1] ≤ [S3] ≤ [S2], [S3] → [D].
  - If only the upper limit is controlled, set the 16-bit signed minimum value, ie -32768, in the lower limit [S1].

- If only the lower limit is controlled, set the 16-bit signed maximum value in the upper limit [S2], that is, 32767.

■ 32 bit instruction

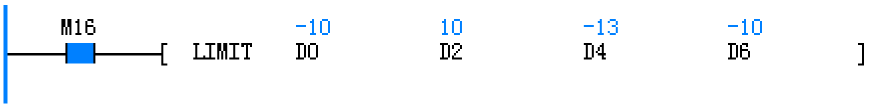
- By setting the upper and lower limits in [S1+1, S1], [S2+1, S2], the output value [D+1, D] is based on the input value [S3+1, S3] within a certain range.
- When [S1+1,S1]>[S3+1,S3], [S1+1,S1]→[D+1,D].
- When [S2+1, S2] < [S3+1, S3], [S2+1, S2] → [D+1, D].
- When [S1+1, S1] ≤ [S3+1, S3] ≤ [S2+1, S2], [S3+1, S3] → [D+1, D]

■ If only the upper limit is controlled, set the 32-bit signed minimum value, ie -2,147,483,648, in the lower limit [S1+1, S1]. If only the lower limit is controlled, set the 32-bit signed maximum value in the upper limit [S2+1, S2], ie 2,147,483,647

■ Some of the following conditions will report a malfunction. The error flag M8067 turns ON and the error code is saved in D8067.

16 bit instruction and 32 bit instruction. When the lower limit value > upper limit value, error 6706 is reported.

【Example】



BZAND Dead zone control (in development)

1. Instruction form

An instruction to control the output value by judging whether the input value is within the upper and lower limits of the specified dead zone

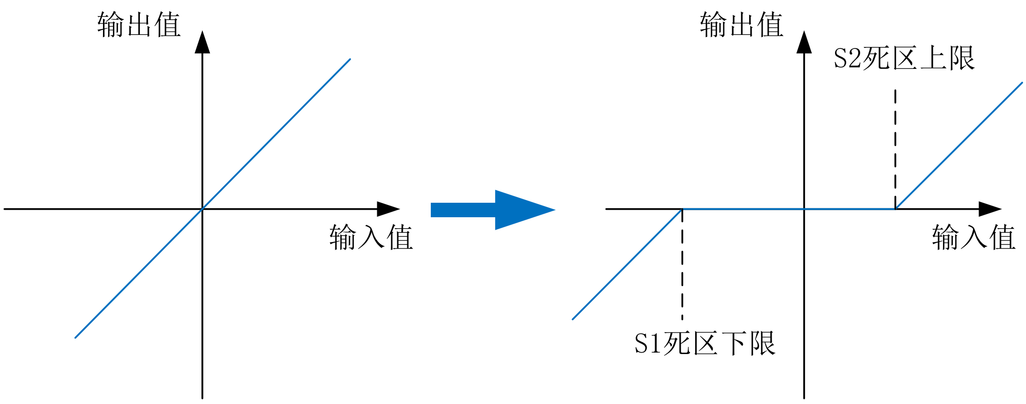
<table>
<tr><td colspan="3">BZAND S1 S2 S3 D</td><td>Dead zone control</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S1</td><td>lower limit</td><td colspan="2">Lower limit of dead zone (no output area)</td><td rowspan="4">16 bit instruction （9step）
BZAND Continuous execution
BZANDP Pulse execution</td><td rowspan="4">32 bit instruction （17step）
DBZAND Continuous execution
DBZANDP Pulse execution</td></tr>
<tr><td>S2</td><td>Upper limit</td><td colspan="2">Upper limit of the dead zone (no output area)</td></tr>
<tr><td>S3</td><td>input value</td><td colspan="2">Input value to be controlled by dead zone</td></tr>
<tr><td>D</td><td>output value</td><td colspan="2">Save the device number of the output value controlled by the deadband</td></tr>
</table>

2. Operand

<table>
<tr><td rowspan="2">Operand</td><td colspan="7">Bit device</td><td colspan="15">Word device</td></tr>
<tr><td colspan="7">System·User</td><td colspan="5">System·User</td><td colspan="5">Digit designation</td><td colspan="2">Indexing</td><td colspan="2">constant</td><td>Real number</td></tr>
<tr><td>S1</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
<tr><td>S2</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
<tr><td>S3</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
<tr><td>D</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
</table>

Note: With gray shading device, it means support.

Function Description:



■ 16 bit instruction

- By setting the deadband range in [S1] and [S2], the input value [S3] is output to the [D] device outside the deadband range. The output value is controlled as shown below.
- When [S1]>[S3], [S3]-[S1]→[D].
- When [S2]<[S3], [S3]-[S2]→[D].
- When [S1] ≤ [S3] ≤ [S2], 0 → [D].

■ 32-bit instruction

- By setting the deadband range in [S1+1, S1], [S2+1, S2], the input value [S3+1, S3] is output to [D+1, D] soft outside the deadband range. In the component. When [S1+1, S1]>

[S3+1, S3], [S3+1, S3]- [S1+1, S1] →[D+1,D];

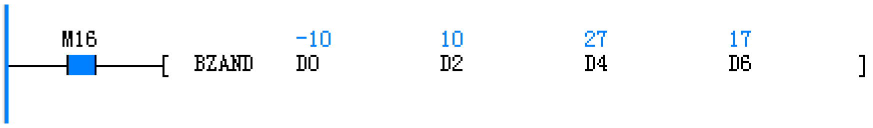
- When [S2+1, S2] < [S3+1, S3], [S3+1, S3]- [S2+1, S2] →[D+1,D]; when [S1+1, S1] ≤ [S3+1, S3] ≤ [S2+1, S2], 0 → [D+1, D].

■ During the use of the instruction, the overflow of the data conforms to the loop processing, that is, the maximum value is increased by 1 to become the minimum value, and the minimum value is decreased by 1 to become the maximum value.

■ Some of the following conditions will report a malfunction. The error flag M8067 turns ON and the error code is saved in D8067.

- 16 bit instruction and 32 bit instruction. When lower limit>Upper limit, error 6706 is reported.

【Example】



ZONE Regional control (in development)

1. Instruction form

An instruction to control the output value with a specified offset value depending on whether the input value is positive or negative.

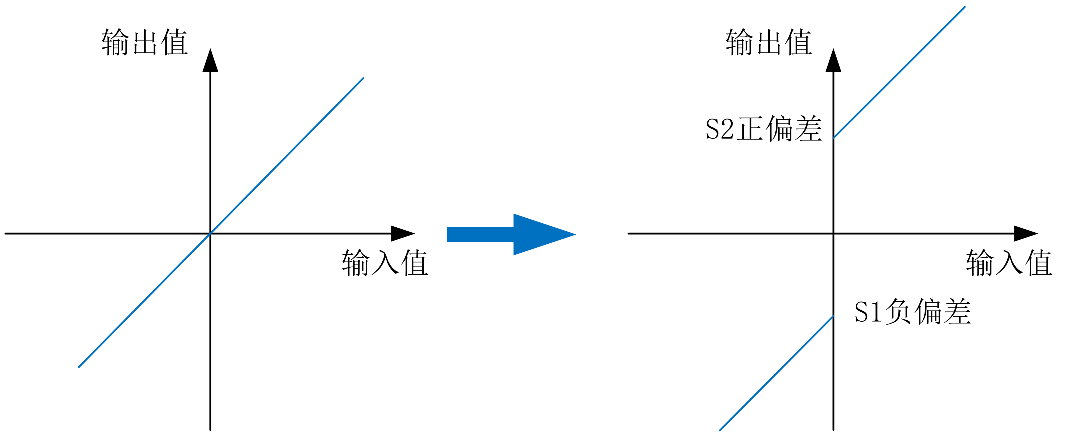
ZONE S1 S2 S3 D		Regional control	Instruction execution	
S1	Negative deviation	Negative deviation value added to the input value (can be positive, negative, 0)	16 bit instruction (9step) ZONE Continuous execution ZONEP Pulse execution	32 bit instruction (17step) DZONE Continuous execution DZONEP Pulse execution
S2	Positive deviation	Positive deviation value added to the input value (can be positive, negative, 0)		
S3	input value	Input value to pass the zone control		
D	output value	Save the device start number of the output value that has passed the zone control		

2. Operand

Operand	Bit device							Word device															
	System·User							System·User					Digit designation					Indexing		constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:



- 16 bit instruction
  - According to the symbol of input value[S3], add [S2] or [S1], and execute result to save to [D] component. When [S3] < 0, [S3] + [S1] → [D].
  - When [S3]>0, [S3]+[S2]→[D].
  - When [S3]=0, 0→[D]。
- 32 bit instruction
  - According to the symbol of input value [S3+1, S3], add [S2+1, S2] or [S1+1, S1], and save the result to [D+1, D] component.

【Example】

4.4.5 Index operation

Index operation	
EXP	Binary floating number index operation
LOGE	Binary floating number Natural logarithm operation
LOG	Binary floating number with a logarithm of 10
ESQR	Binary floating number Open square operation
SQR	Binary data Open square operation

EXP Binary floating number index operation

1. Instruction form

Perform index operation of Binary floating number data based on e (2.71828)

EXP S D			Binary floating number index operation	Instruction execution
S	data source	Binary floating number variable of Binary floating number index		32 bit instruction (9step) DEXP Continuous execution DEXPP Pulse execution
D	operation result	Calculate the storage unit of result after index operation		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This command is an index operation for Binary floating numberdata based on e(2.71828).
- S is the Binary floating number variable of Binary floating numberindex.
- D is the storage unit of result after calculating the index operation.

Note: An operation error will occur when the operation result is not [2-126 ≤ operation result < 2128].

The error code is K6706, K6706 is saved in D8067, and error flag M8067 is turned ON.

【Example】

When X1 is ON, the Binary floating number value in the pair e (D11, D10) is indexed and stored in (D101, D100).

LOGE Binary floating number natural logarithm operation

1. Instruction form

Perform the natural logarithm operation of Binary floating numberdata based on e(2.71828).

LOGE S D		Binary floating number natural logarithm operation	Instruction execution
S	data source	Binary floating number of the natural logarithm of Binary floating number	32 bit instruction (9step) DLOGE Continuous execution DLOGEP Pulse execution
D	operation result	Calculate the storage unit of result after natural logarithm operation	

2. Operand

Operand	Bit device							Word device															
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is a natural logarithm operation of Binary floating numberdata based on e(2.71828).
- Note: The value in S can only be a positive number. When the content value of S is 0 or a negative number, an operation error will occur. The error code is K6706, K6706 will be saved in D8067, error flag M8067 will be turned ON.

【Example】

When X1 is ON, the Binary floating number value in the pair e (D11, D10) is subjected to a natural logarithm operation and stored in (D101, D100).

LOG Binary floating number with a base 10 logarithm operation

1. Instruction form

Perform a common logarithm operation of the base 10 Binary floating numberdata operation

LOG S D		Binary floating number10 as the base log operation	Instruction execution
S	data source	Binary floating number variable for common logarithm of Binary floating number	32 bit instruction (9step) DLOG Continuous execution DLOGP Pulse execution
D	operation result	After the common logarithm operation, calculate the storage unit of result	

2. Operand

Operand	Bit device	Word device

	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

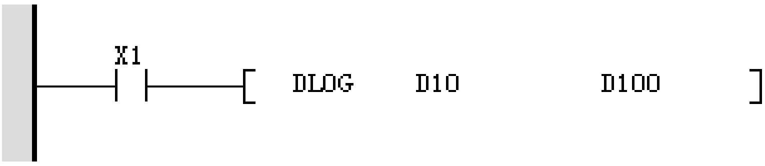
Note: With gray shading device, it means support.

Function Description:

This instruction is a common logarithm operation of the base 10 Binary floating number data.

Note: The value in S can only be a positive number. An operation error will occur when the content value of S is 0 or a negative number. The error code is K6706, K6706 will be saved in D8067, error flag M8067 will be turned ON

【Example】



When X1 is ON, the value of Binary floating number in the pair of e (D11, D10) is stored in natural logarithm operation (D101, D100).

ESQR Binary floating number Open square operation

1. Instruction form

The Open square operation of the binary floating number, that is, the Square root of the Binary floating number.

ESQR S D		Binary floating number Open square operation	Instruction execution
S	data source	Waiting for the Square root floating number variable of Square root	32 bit instruction (9step) DESQR Continuous execution DESQRP Pulse execution
D	operation result	Binary floating number square root storage unit	

2. Operand

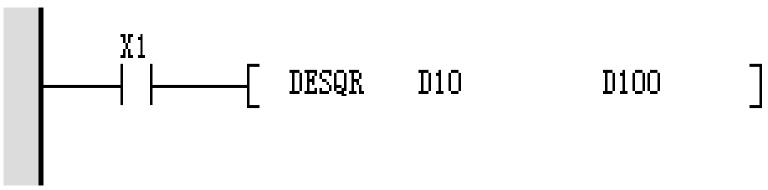
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is the square root operation of the binary floating numbers, that is, the square root of the Binary floating number.
- If OperandS is constantK or H, it will automatically convert the constant into a Binary floating number value for the opening operation;
- 0 If the result of the calculation is zero, the 0 flag (M8020) will be set.
- S only has a positive number, if it is a negative number, the calculation is wrong, M8067, M8068 will turn ON.

【Example】



When X1 is ON, the square of the Binary floating number is squared.result $\sqrt{(D11, D10)}$ ，Stored in (D101, D100)。

SQR Binary data Binary data Open square operation

1. Instruction form

Binary data Open square operation

SQR S D			Binary data Open square operation	Instruction execution	
S	data source	Open square operation Data or data storage Word device address		16 bit instruction（5step） SQR Continuous execution SQRP Pulse execution	32 bit instruction（9step） DSQR Continuous execution DSQRP Pulse execution
D	operation result	Storage unit of operation result			

2Operand

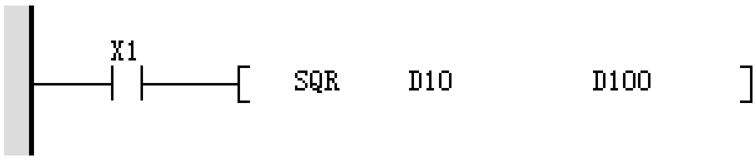
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- S is squared according to the BIN value, and result is stored in D.
- Only S can be specified as a positive number. If S is negative, the operation error flag M8067 will turn ON and the instruction will not be executed.
- Operation resultD takes only an integer. The decimal point is rounded off, and the borrowing flag M8021 turns ON when the decimal point is rounded off.
- When the operation result is 0, the zero mark M8020 turns ON

【Example】



When X1 is ON, the Binary floating number is stored in the D100.

If D10 is 9, calculate resultD100 to be 3.

If D10 is 10, the calculation records that D100 is 3, and the fractional part is rounded off.

4.5 Data processing

Data conversion	
INT	Binary floating number→BIN integer transform
FLT	Binary data→ Binary floating number conversion
BCD	Binary data→BCD data
BIN	BCDdata→ Binary data
WTOB	Byte unit of data Separation
BTOW	Byte unit of data combination
UNI	4-bit combination of 16-bit data
DIS	4-bit Separation of 16-bit data
ASCI	HEX→ASCIIconversion
HEX	ASCII→HEXconversion
Data transfer	
MOV	Assignment transfer
EMOV	Binary floating numberTransfer
SMOV	Shift Transfer
BMOV	Data batch transfer
FMOV	Data one-to-many transfer
CML	Data NegateTransfer
Table operation	
ZRST	Reset all data
Data shift	
ROR	Loop right shift
ROL	Loop left shift
RCR	Loop right shift with carry
RCL	Loop left shift with carry
Other data processing	
SWAP	Upper and lower byte swap
BON	ON bit judgment
SUM	ON is the total
RND	Generate random data
XCH	Data exchange

4.5.1 Data conversion

Data conversion	
INT	Binary floating number→BIN integer transform
FLT	Binary data→ Binary floating number conversion
BCD	Binary data→BCD data
BIN	BCD data → Binary data
WTOB	Byte unit of data Separation
BTOW	Byte unit of data combination
UNI	4-bit combination of 16-bit data
DIS	4-bit Separation of 16-bit data
ASCI	HEX→ASCII conversion
HEX	ASCII→HEX conversion

INT Binary floating number→BIN integer transform

1. Instruction form

Perform a rounding operation of the binary floating number, discard the fractional part, and store the binary result in D.

INTSD		Binary floating number→BIN integer transform	Instruction execution	
S	data source	Binary floating number variable to be rounded	16 bit instruction （5step） INT Continuous execution INTP Pulse execution	32 bit instruction （9step） DINT Continuous execution DINTP Pulse execution
D	operation result	Storage unit of transformed BIN integer result		

2. Operand

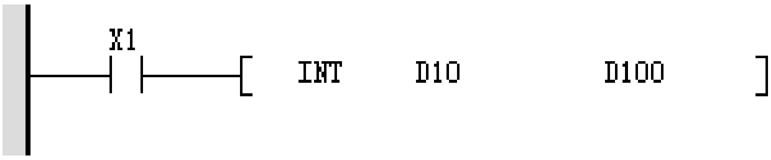
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is a rounding operation of the binary floating number, discarding the fractional part and storing the binary result in D.
- When S = 0, the M8020 will be set.
- When ｜S｜ ≤ 1, the borrow flag (M8021) is set.
- If the operation result is outside the following range (overflow), the carry flag (M8022) will be set.。
- 16 bit instruction：-32768~32767。
- 32 bit instruction：-2147483648~2147483647。

【Example】



When X1 is ON, the floating number (D11, D10) is rounded and stored in D100.

FLT Convert binary integer to Binary floating number

1. Instruction form

Convert binary integer to Binary floating number

FLT S D		Convert binary integer to Binary floating number	Instruction execution	
S	Integer	The binary number to be converted or data storage Word device address	16 bit instruction (5step) FLT Continuous execution FLTP Pulse execution	32 bit instruction (9step) DFLT Continuous execution DFLTP Pulse execution
D	Floating number	Floating number data storage Word device address		

2. Operand

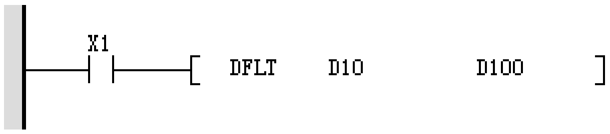
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Convert the integer S to a floating number and store the result in D and D+1 units.
- Constant K and H are automatically converted in each floating point operation instruction, so they cannot be used in this FLT instruction. The inverse transform instruction for this instruction is INT (converts 2 into a floating number value to a BIN integer).

【Example】



When X1 is ON, the 32-bit number (D11, D10) (32-bit BIN integer) is converted to Binary floating number and stored in (D101, D100).

BCD Binary data converts BCD data (in development)

1. Instruction form

Binary data converts BCD data

BCD S D		Binary data converts BCD data	Instruction execution	
S	data source	Binary code data or data storage Word device address	16 bit instruction (5step) BCD Continuous execution BCDP Pulse execution	32 bit instruction (9step) DBCD Continuous execution DBCDP Pulse execution
D	operation result	BCD code data storage Word device address		

2. Operand

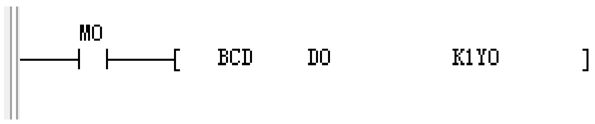
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Need to drive out, there are 2 operating variables, the value of S (BIN) is BCD transformed and stored in D. This instruction is often used to process the data format before the data is displayed.
- When using a 16-bit command, the range is 0-9999. If the file conversion result exceeds 9999, an error occurs. When using a 32-bit command, the range is 0-99999999. When the conversion result exceeds 99999999, an error occurs. M8067, M8068 will be set, D8067 will record the error code.

【Example】



After converting the BIN value of D0 to the BCD value, the single digit of the result is stored in K1Y0 (four bit elements from Y0 to Y3).

If D0=H00F4 (hexadecimal)=K244 (decimal), the result of the transformation is Y0~Y3=0100 (BIN).

If D0=H0046 (hexadecimal)=K70 (decimal), the result of the conversion is Y0~Y3=0000(BIN).

BIN BCD data → Binary data

1. Instruction form

BCD data converse Binary data.

EBIN S D		Binary floating number→decimal floating number transform	Instruction execution	
S	data source	BCD code data or data storage Word device address	16 bit instruction (5step) BIN Continuous execution BINP Pulse execution	32 bit instruction (9step) DBIN Continuous execution DBINP Pulse execution
D	operation result	Binary code data storage Word device address		

2. Operand

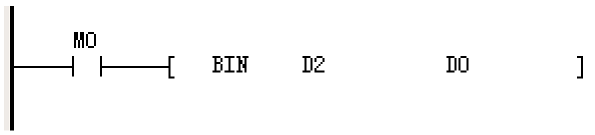
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Requires contact drive. There are two operation variables. The value of S(BCD) is BIN transformed and stored in D. This instruction is often used to process external port read data (such as code disk settings) into a BIN format that can be used directly for operation.
- S (BCD) effective range, 16bit: 0~9999; 32bit: 0~99999999
- S data content is not BCD value (Hex indicates that any digit is not in the range of 0~9), an operation error will occur, M8067, M8068 will be set, D8067 will record the error code.

【Example】



When M0 is ON, the BCD value of D2 is converted into D0 for BIN conversion.

WTOBByte unit data separation

1. Instruction form

Instruction is that separate consecutive 16-bit data in bytes (8-bit) units.

WTOB S D n		Byte unit data separation	Instruction execution
S	source data	Save the device start number of the data to be separated in byte units	16 bit instruction（7step） WTOB Continuous execution WTOBP Pulse execution
D	result	Save the device start number of the result that has been separated by byte units	
n	Separation number	The number of bytes of data to be separated (n ≥ 0, and is not processed when n = 0)	

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

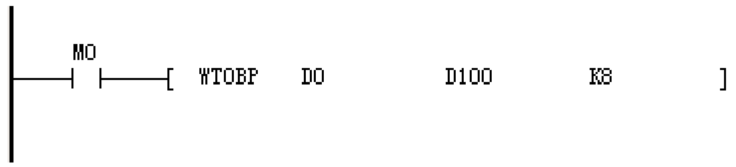
The 16-bit data stored in the device starting with [S] is stored in bytes in the lower 8 bits of the n-point device starting with [D], and the upper 8 bits store 00H.

Some of the following conditions will report a failure. The error flag M8067 turns ON and the error code is saved in D8067. When the device starting with [S], [D] exceeds the range of its device, the error No. 6705 is reported.

S+0	High byte	Low byte
S+1	High byte	Low byte
……	……	……
S+n/2	High byte	Low byte

D+0	00H	Low byte
D+1	00H	High byte
D+2	00H	Low byte
D+3	00H	High byte
……	……	……
D+n-2	00H	00H
D+n-1	00H	High byte

【Example】



source data

D0	0X1122
D1	0X3344
D2	0X5566
D3	0X7788

Separate result

D100	0X22
D101	0X11
D102	0X44
D103	0X33
D104	0X66
D105	0X55
D106	0X88
D107	0X77

BTOW Byte unit of data combination

1. Instruction form

The instruction is that combine the lower 8 bits of consecutive 16-bit data.

BTOW S D n		Byte unit of data combination	Instruction execution
S	source data	Save the device start number of the data to be combined in byte units	16 bit instruction（7step） BTOW Continuous execution BTOWP Pulse execution
D	result	Save the device start number of the result that has been combined in byte units	
n	Separation number	The number of bytes of data to be combined (n ≥ 0, and is not processed when n = 0)	

2. Operand

Operand	Bit device							Word device															
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	**D**	**R**	**T**	**C**	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modify**	K	H	E
D	X	Y	M	T	C	S	SM	**D**	**R**	**T**	**C**	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	**Modify**	K	H	E
n	X	Y	M	T	C	S	SM	**D**	**R**	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	**K**	**H**	E

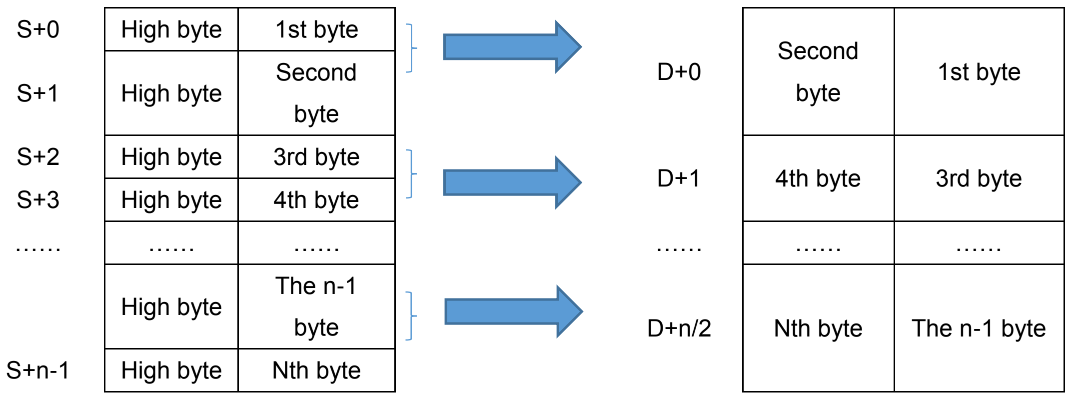
Note: With gray shading device, it means support.

Function Description:

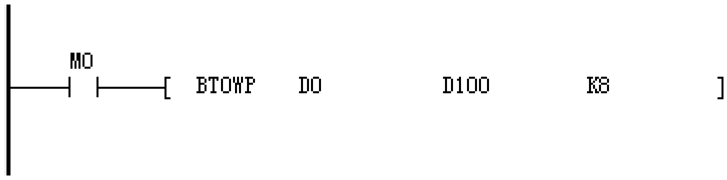
The 16-bit data obtained by combining the lower 8 bits of the n 16-bit data starting from [S] is saved to the device starting with [D], and the upper 8 bits of [S] are ignored.

Some of the following conditions will report a failure. The error flag M8067 turns ON and the error code is saved in D8067.

When the device starting with [S], [D] is beyond the range of its device, error No. 6706 is reported.



【Example】



source data

D0	0X1122
D1	0X3344
D2	0X5566
D3	0X7788
D4	0X99AA
D5	0XBBCC
D6	0XDDEE
D7	0XFF00

Separation result

D100	0X4422
D101	0X8866
D102	0XCCAA
D103	0X00EE

UNI 4-bit combination of 16-bit data (in development)

1. Instruction form

Instruction is that combines the lower 4 bits of consecutive 16-bit data.

UNI S D n		4-bit combination of 16-bit data	Instruction execution
S	source data	Save the device start number of the data to be combined	16 bit instruction（7step） UNI Continuous execution UNIP Pulse execution
D	result	Save the device number of the combined data	
n	Combination number	Number of combinations (0 to 4, no processing when n=0)	

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

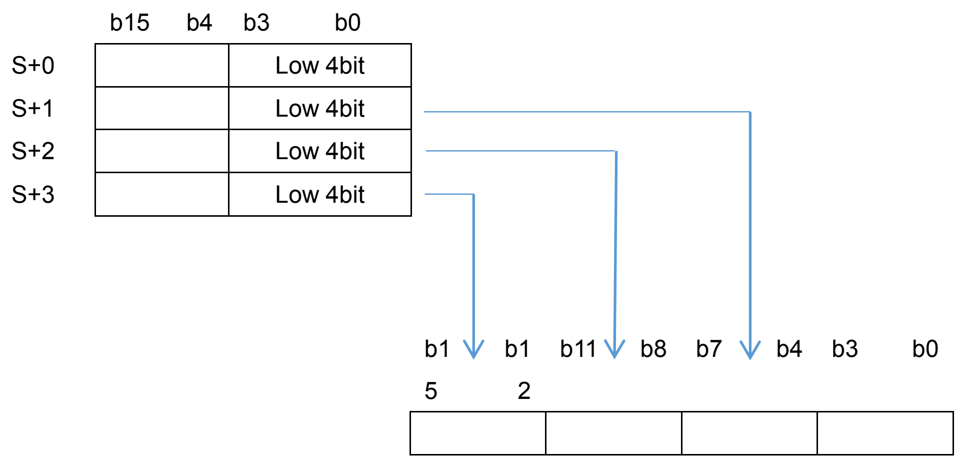
Note: With gray shading device, it means support.

Function Description:

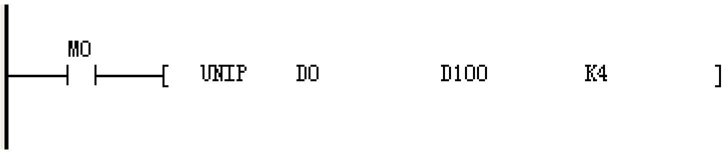
- Combine the lower 4 bits of the n 16-bit data starting from S into 16-bit data and store it in D.
- n ranges from 1 to 4. When n=0, the instruction is not executed. When n is 1~3, the remaining high 0 is filled.

An operation error will occur in some cases, the error flag M8067 turns ON, and the error code is saved in D8067.

- When the device set in S is out of range, the error code K6705 is reported;
- When the n range is set, the error code K6706 is reported.



【Example】



source data

D0	0X1122
D1	0X3344
D2	0X5566
D3	0X7788

Combine result

D100	0X8642

DIS separate 16-bit data in units of 4 bits (in development)

1. Instruction form

Instruction is that separate 16-bit data in units of 4 bit。

DIS S D n			separate 16-bit data in units of 4 bits	Instruction execution
S	source data	Save the device start number of data to be separated		16 bit instruction（7step） DIS Continuous execution DISP Pulse execution
D	result	Save device number of detached data		
n	Separation number	Separation number (0 to 4, no processing when n=0)		

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

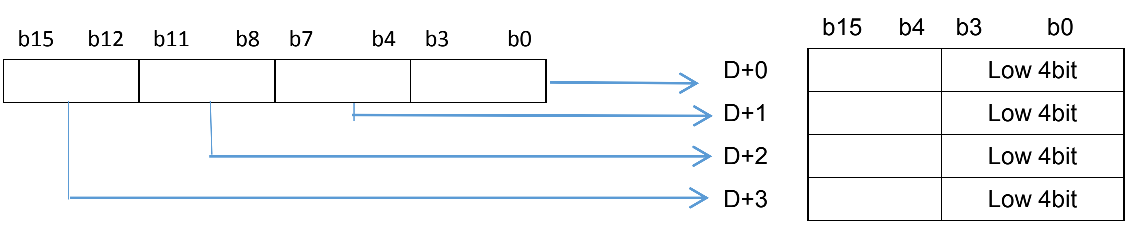
Note: With gray shading device, it means support.

Function Description: 低

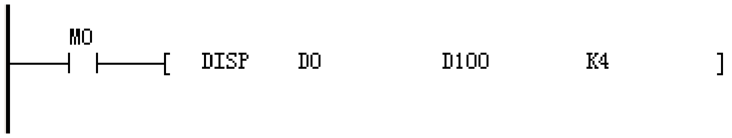
- 16 The 16-bit data of S is separated by 4 bits and stored in the Low 4bit device of D, and the upper 12 bits are padded with 0.
- n ranges from 1 to 4. When n=0, the processing of the instruction is not executed.

An operation error will occur in some cases, the error flag M8067 turns ON, and the error code is saved in D8067.

- When the device set in D is out of range, error code K6705 is reported.
- When the n range is set, the error code K6706 is reported.



【Example】



source data	
D0	0X1122

Separation result

D100	0X4
D101	0X3
D102	0X2
D103	0X1

ASCI HEX → ASCII conversion (in development)

1. Instruction form

After converting the value of S into ASCII code, store it in the variable where D is the starting address.

ASCI S D n			HEX→ASCII conversion	Instruction execution
S	data source	The variable to be converted, address or constant value		16 bit instruction（7step） ASCI Continuous execution ASCIP Pulse execution
D	operation result	The starting address of the converted ASCII code		
n	Number of converted characters	The number of characters to be converted, n ranges from 1 to 256		

2. Operand

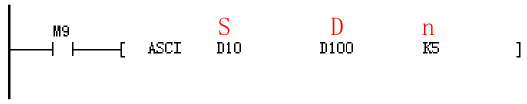
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This command converts the value of S into ASCII code and stores it in the variable where D is the starting address.
- ASCI value conversion follows the ASCII and HEX binary value comparison table, such as: ASCII '0' corresponds to HEX 'H30'; ASCII 'F' corresponds to HEX 'H46', etc.

【Example】



D10	4	3	2	1
D11	8	7	6	5
D12	C	B	A	9
D13	0	F	E	D

S → D10; n → 5

M8161=OFF16bit mode M8161=ON8bit mode

1								1
2							1	2
3						1	2	3
4	No change				1	2	3	4
5				1	2	3	4	5
6			1	2	3	4	5	6
7		1	2	3	4	5	6	7
n	H	L	H	L	H	L	H	L
	D103		D102		D101		D100	

1								1
2							1	2
3						1	2	3
4	No change				1	2	3	4
5				1	2	3	4	5
6			1	2	3	4	5	6
7		1	2	3	4	5	6	7
n	H	L	H	L	H	L	H	L
	D107	D106	D105	D104	D103	D102	D101	D100

- Among them, the M8161 flag determines the width mode of the destination variable to be calculated. When M8161=OFF, it is 16bit mode, that is, the high byte and the low byte of the variable are stored separately; when M8161=ON, it is the 8-bit mode, only the variable Low byte stores result, so the length of the actual variable area is increased.

b15 D10=1234H b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
1				2				3				4			

b15 D11=8765H b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
1				2				3				4			

When M8161=OFF, n=4, the composition of the bit is M8161=OFF, n=5, the composition of the bit

(D10~D11) conversion (D10~D11) conversion

b15 D100 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
[2]=32H								[1]=31H							

b15 D101 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
[4]=34H								[3]=33H							

b15 D102 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0

b15 D100 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
[1]=31H								[5]=35H							

b15 D101 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
[3]=33H								[2]=32H							

b15 D102 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[4]=34H							

When M8161=ON, n=4, the composition of the bit is M8161=ON, n=5, the composition of the bit

(D10~D11) conversion (D10~D11) conversion

M8161=ON, n=4

b15 D100 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[1]=31H							

b15 D101 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[2]=32H							

b15 D102 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[3]=33H							

b15 D103 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[4]=34H							

b15 D104 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0

M8161=ON, n=5

b15 D100 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[5]=35H							

b15 D101 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[1]=31H							

b15 D102 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[2]=32H							

b15 D103 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[3]=33H							

b15 D104 b0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
								[4]=34H							

Note: Instructions such as RS/HEX/ASCI/CCD share the M8161 mode flag.

HEX ASCII→HEX conversion (in development)

1. Instruction form

After the initial value S is converted into a variable HEX code and stores a starting address of the variable D, the conversion of the number of characters, the storage mode may be set

HEX S D n		ASCII→HEX conversion	Instruction execution
S	data source	The value of the variable to be converted, address or constant, if it is a register variable, the conversion is separated by 32-bit variable width (4 ASCII characters).	16 bit instruction（7step） HEX Continuous execution HEXP Pulse execution
D	operation result	The storage start address of the HEX code after conversion, the occupied variable space is related to S2	
n	Number of characters converted	Number of characters converted	

2. Operand

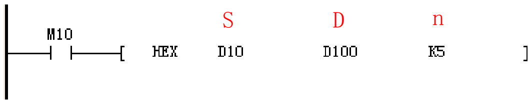
Operand	Bit device							Word device															
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This command converts the value of the S start variable into a HEX code and stores it in the variable where D is the start address. The number of characters converted and the storage mode can be set.
- S is the variable address or constant value to be converted. If it is a register variable, the conversion is separated by the 32bi variable width (ie 4 ASCII characters).
- D is the storage start address of the converted HEX code, and the occupied variable space is related to n.
- n is the number of characters converted (range: 1~256)

【Example】



For example, the data starting from D100 is as follows:

M8161=OFF16bit mode

S	H	L
D10	31H（1）	30H（0）
D11	33H（3）	32H（2）
D12	35H（5）	34H（4）
D13	37H（7）	36H（6）
D14	39H（9）	38H（8）
D15	42H（B）	41H（A）
D16	44H（D）	43H（C）
D17	31H（1）	30H（0）
D18	33H（3）	32H（2）

1			0H
2			01H
3			012H
4	No change		0123H
5		…0H	1234H
6		..01H	2345H
7		.012H	3456H
8		0123H	4567H
9	…0H	1234H	5678H
n	D102	D101	D100

M8161=OFF8bit mode

S	H	L
D10	31H（1）	30H（0）
D11	33H（3）	32H（2）
D12	35H（5）	34H（4）
D13	37H（7）	36H（6）
D14	39H（9）	38H（8）
D15	42H（B）	41H（A）
D16	44H（D）	43H（C）
D17	31H（1）	30H（0）
D18	33H（3）	32H（2）

1			0H
2			02H
3			024H
4	No change		0246H
5		…0H	2468H
6		..02H	468AH
7		.024H	68ACH
8		0246H	8AC0H
9	…0H	2468H	AC02H
n	D102	D101	D100

Among them, the M8161 flag determines the variable width mode. When M8161=OFF, it is 16-bit mode, that is, the variable High byte and Low byte participate in the operation; when M8161=ON, it is the 8-bit mode, only the variable Low byte participates operation, the content of the High byte is discarded, so the length of the actual variable area S is actually increased.

When M8161=OFF, n=4, the composition of the bit is M8161=OFF, n=5, the composition of the bit
(D10~D11) conversion (D10~D11) conversion

b15							D100								b0
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
0				1				2				3			
b15							**D101**								**b0**
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0

b15							D100								b0
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
1				2				3				4			
b15							**D101**								**b0**
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
												0			

When M8161=OFF, n=4, the composition of the bit is M8161=OFF, n=5, the composition of the bit

(D10~D11) conversion (D10~D11) conversion

b15							D100								b0
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
0				2				4				6			
b15							**D101**								**b0**
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0

b15							D100								b0
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
2				4				6				8			
b15							**D101**								**b0**
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
												0			

Note:

Commands such as RS/HEX/ASCI/CCD share the M8161 mode flag, and pay attention when programming;

The source data of the S data area must be an ASCII code character, otherwise the conversion error occurs;

If the output data is in BCD format, after HEX conversion, BCD-BIN conversion is required, which is the correct value.

4.5.2 Data Transfer

data Transfer	
MOV	Assignment Transfer
EMOV	Binary floating number transfer
BMOV	Data batch transfer
FMOV	One-to-many data transfer
CML	Negate data transfer

MOV Assignment transfer

1. Instruction form

Copy the data from source S to the final destination D

MOV S D		Data transfer	Instruction execution	
S	data source	Data to be transferred or data storage Word device address	16 bit instruction (5step) MOV Continuous execution MOVP Pulse execution	32 bit instruction (9step) DMOV Continuous execution DMOVP Pulse execution
D	Data assignment destination	Destination data storage Word device address		

2. Operand

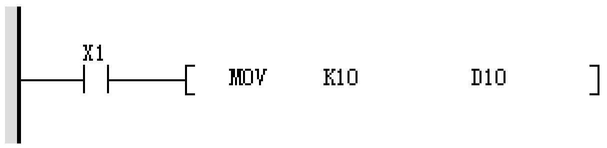
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- Requires contact drive, there are 2 manipulated variables, copy the value of S to D.
- When it is a 32-bit instruction (DMOV), both S and D use the variable unit of the adjacent high address to participate in the operation. For example, the operation results of [DMOVD1D10] is: D1→D10; D2→D11.

【Example】



When X1 is ON, constant10 is assigned to D10.

EMOV Binary floating number transfer

1. Instruction form

The transfer of Binary floating number data is performed. Requires contact drive. When the Instruction is executed, copy the binary floating number data value of S to D.

EMOV S D		Binary floating number transfer	Instruction execution
S	data source	Binary floating number data transmission source	32 bit instruction（9step） DEMOV Continuous execution DEMOVP Pulse execution
D	Transfer destination	The storage unit for storing Binary floating number data	

2. Operand

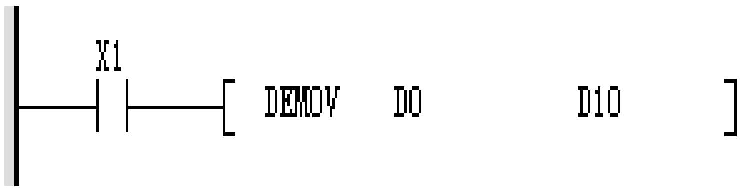
Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: With gray shading device, it means support.

Function Description:

- This instruction is that transfer Binary floating number data. A contact drive is required. When the Instruction is executed, the binary number of the
- S number is copied to D. among them: S is the transmission source of Binary floating number data;
- D is the storage unit for storing Binary floating number data

【Example】



When X1 is ON, the Binary floating number in (D1, D0) is saved to (D11, D10).

SMOV Shift transfer（In development）

1. Instruction form

Moving the digital data of the m2 digit starting from the m1 digit in S to the m2 digit starting from the n digit in the terminal D

SMOV S m1 m2 D n		Binary floating number transfer	Instruction execution
S	data source	Digital transmission Data storage Word device address	16 bit instruction（11step） SMOV Continuous execution SMOVP Pulse execution
m1	Transfer start bit	The position of the start bit to be moved in S	
m2	Number of transmission bits	The number of bits to move in S	
D	Destination device	Destination of transfer data storage Word device address	
n	Destination start bit	Move to the position of the starting digit in D	

2. Operand

Operand	Bit device							Word device														
	System·User							System·User					Digit designation					Indexing		constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

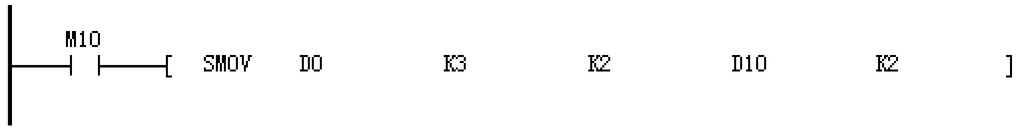
Note: With gray shading device, it means support.

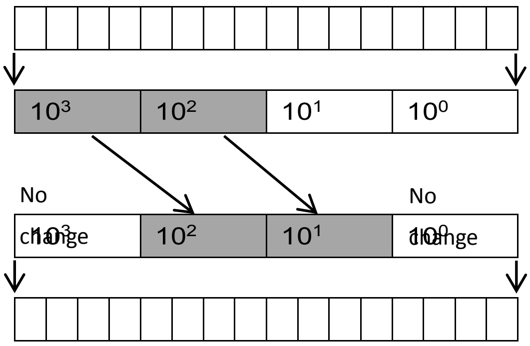
Function Description:

Contact drive is required, up to 5 operating variables:：

- S is the data source variable to be copied. When M8168 is OFF, it is BCD mode (decimal bit), and the range of MOperand is 0000~9999, which cannot be negative. When M8168 is ON, it is BIN mode, and SOperand can be negative.
- m1 is the starting bit number of the data source, (1~4) range;
- m2 is the number of bits transmitted by the data source, (1~m1) range;
- D is the destination variable transmitted by the data source;
- n is the starting bit of the destination variable transmitted by the data source, (m2 to 4).
- The data bit transfer process is related to the status of the special flag M8168. When M8168 is OFF, it is BCD mode (decimal bit). When M8168 is ON, it is BIN mode. In BIN mode, 4 bits are used as a unit for transmission. (hexadecimal digits).

【Example】





When M8168=OFF, the calculation process is as follows：

D0 （BIN hex 16bit）

↓ （Automatic conversion）

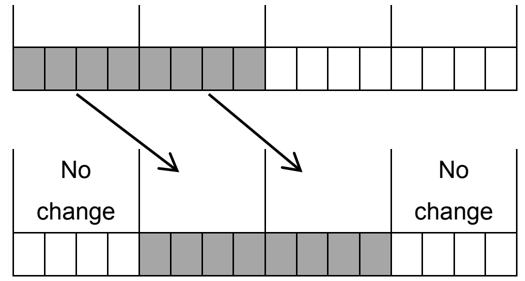
D0 （BCD format 4 digits）

↓ （Bit movement）

D10 （BCD format 4 digits）

↓ （Automatic conversion）

D10（BIN format 16bit）



When M8168=ON, the calculation process is as follows：

D0（HEX format 4 digits）

↓ （Bit movement）

D10（HEX format 4 digits）

Assuming D0=K1234, D10=K5678, when M8168 is OFF (BCD mode), turn M10 ON, then the value of D10 becomes K5623;

When M8168 is ON (BIN mode), D0=H04 D2=K1234, D10=H162E=K5678, turn M10 ON, then D10=H12D2=K4818

BMOV Data batch transfer

1. Instruction form

When the driving condition is satisfied, the data of the n registers with the S first address are transmitted one by one to the n registers with the D as the first address.

BMOV S D n			Data batch transfer	Instruction execution
S	Data source first address	Data transfer for batch transfer Word soft element first address		16Bit instruction（7step） BMOV Continuous execution BMOVP Pulse execution
D	The first transfer destination address	Delivery destination data storage Word soft element first address		
N	Data length	Batch transfer Word soft element points		

2. Operands

Operands	Bit soft component	Word soft element				
	System · user	System · user	Digit assignment	Indexed address	Constant	Real number

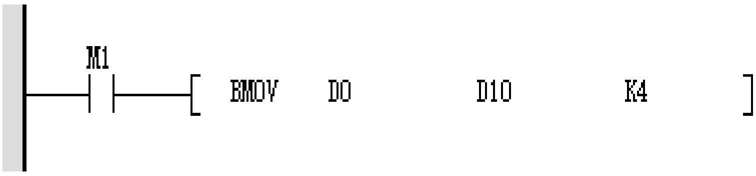
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- Contact drive is required. There are 3 operation variables. Copy n variable values of the start address specified by s to n cells of the start address specified by D. The value range of n is from 1 to 512.
- When the special variable m8024 = 1, the direction of batch transmission is opposite, that is, n variable values of the start address specified by D are copied to n cells of the start address specified by S.
- When operators are bit components, the s and d bits must be equal

【Example】



When M1 is on, the value of (D3, D2, D1, D0) is transmitted to (D13, D12, D11, D10).

Fmov data one to many transfer

1. Instruction form

When the driving condition is established, the data in S is transmitted to n registers with D as the first address.

FMOV S D n			Data one to many transfer	Instruction execution	
S	Data source	One to many data transfer or data storage word soft element		16Bit instruction（7step）FMOV Continuous execution FMOVP Pulse execution	32Bit instruction（13step）DFMOV Continuous execution DFMOVP Pulse execution
D	Data transmission destination address	The first address of the word soft element in the data storage of the transfer destination			
N	Target number	The number of word soft elements for multicast			

2. Operands

Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- Contact driven, with 3 operation variables, is required to copy the data from s to N units with start address specified by D. The value range of n is 1-512.
- Fmov is a 16-bit multicast instruction and dfmov is a 32-bit multicast instruction.

【Example】

When M1 is on, 100 is transferred to four registers D13, D12, D11 and D10.

CML data fetching and reverse transmission

1. Instruction form

The source address s is bitwise reversed and transmitted to the end address D.

<table>
<tr><td colspan="3">CML S D</td><td>Data retrieval and reverse transmission</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S</td><td>Reverse data source</td><td colspan="2">The address of word soft element for data retrieval or data storage</td><td rowspan="2">16Bit instruction（5step）
CML Continuous execution
CMLP Pulse execution</td><td rowspan="2">32 位指（9step）
DCML Continuous execution
DCMLP Pulse execution</td></tr>
<tr><td>D</td><td>Transfer destination</td><td colspan="2">The address of the word soft element of the data storage transferred after data reversal</td></tr>
</table>

2. Operands

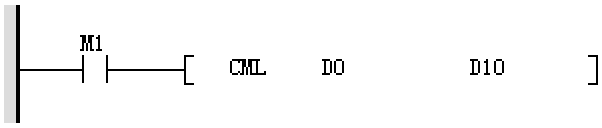
Operands	Bit soft component							Word soft element															
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- It needs to be driven by contacts and has 2 operation variables. The bin value of S is inverted one by one and then copied to d.
- When the digit number of D is less than 16bit, reverse s and transfer it to D variable according to low order alignment;
- When it is a 32-bit instruction (dcml), both s and d use the variable units of the adjacent high address to participate in the operation. For example, the operation result of statement: (dcmld1d5) is / D1 → D5; / D2 → D6.

【Example】



When M1 is on, d0 is reversed and stored in D10.

4.5.3 Table operation

Table operation	
ZRST	Reset all data

ZRSTReset all data

1. Instruction form

Batch reset command

ZRST D1 D2		Reset all data	Instruction execution
D1	Batch reset first address	First address of soft components for batch reset	16Bit instruction（5step） ZRST Continuous execution ZRSTP Pulse execution
D2	Batch reset first address	First address of soft components for batch reset	

2. Operands

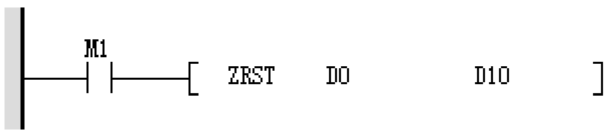
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function description

- Clear all variables from D1 to D2. D1 and D2 can specify word variables or Y, m and S-bit variables. Requirements:
  - D1 and D2 must be soft components of the same type.
  - The number D1 shall not be greater than D2. If the two are the same, only the specified soft element shall be reset.
  - This instruction is 16bit, but D1 and D2 can specify 32bit counter. At this time, they should be both 32bit type or 16bit type.

【Example】



When M1 is on, all registers from d0 to D10 are reset.

Supplementary note: bit element Y, m, s and word elements T, C and D can also be reset individually with RST instruction; word element T, C, D and bit registers kni, knm and kns can also be cleared in multiple points with fmov.

4.5.4 Data shift

Data shift	
ROR	Rotate right
ROL	Rotate left
RCR	Rotate right with carry bit
RCL	Rotate left with carry bit

ROR Rotate right

1. Instruction form

When the driving condition is established, the data in D moves n bits to the right, excluding the carry flag bit m8022, and the low data moved out of D circulates into the high position of D.

ROR D n			Rotate right	Instruction execution	
D	Device to be cycled	Data store word soft element address		16Bit instruction （5step）	32Bit instruction （9step）
N	Single move digits	Effective range: 1 ≤ n ≤ 16 (16 bits), 1 ≤ n ≤ 32 (32 bits)		ROR Continuous execution RORP Pulse execution	DROR Continuous execution DRORP Pulse execution

2. Operands

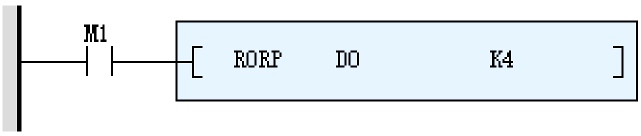
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- Move the loop to the right by N bits.
- This instruction generally uses the pulse execution type instruction. When it is a 32-bit instruction, the register variable occupies a total of 2 units of the subsequent adjacent address.
- Only K4 (16bit) and K8 (32bit) are valid when kni, KM and KS are specified in D.

【Example】



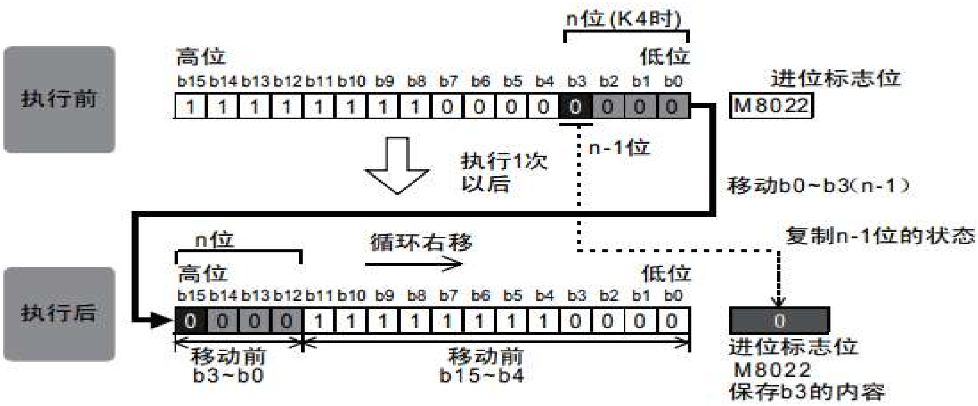
When M1 is on, the number cycle of d0 shifts four bits to the right;

The last bit remains in the carry flag bit m8022.

RC

m80

t



D	Device to be cycled	Data store word soft element address	16Bit instruction（5step） ROL Continuous execution ROLP Pulse execution	32Bit instruction（9step） DROL Continuous execution DROLP Pulse execution
N	Single move digits	Effective range: 1 ≤ n ≤ 16 (16 bits), 1 ≤ n ≤ 32 (32 bits)		

2. Operands

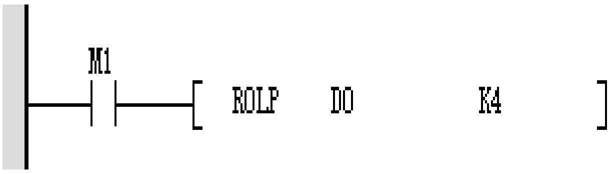
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

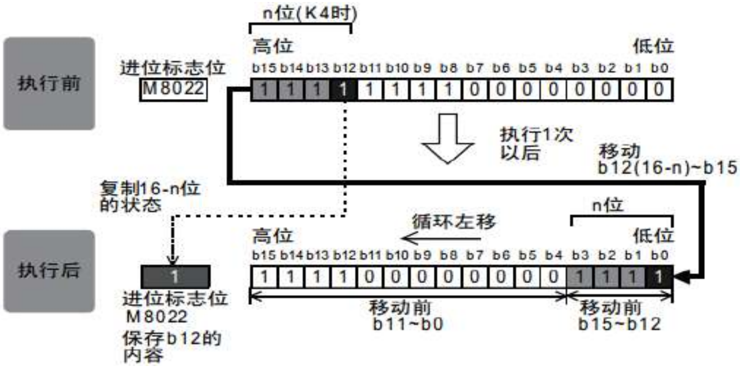
- Move the content of D to the left by N bits.
- This instruction generally uses pulse execution type instruction. When it is a 32-bit instruction, the register variable occupies a total of 2 units of the subsequent adjacent address.
- Only K4 (16bit) and K8 (32bit) are valid when kni, KM and KS are specified in D.

【Example】



When M1 is on, the number cycle of d0 shifts four bits to the left;

The last bit remains in the carry flag bit m8022.



RCR Rotate right with carry bit

1. Instruction form

When the driving condition is established, the data carry (m8022) in D moves n bits to the right, and the moved high carry (m8022) circulates into the high position of D.

RCR D n	Rotate right with carry	Instruction execution

D	Device to be cycled	Data store word soft element address	16Bit instruction （5step） RCR Continuous execution RCRP Pulse execution	32Bit instruction （9step） DRCR Continuous execution DRCRP Pulse execution
N	Single move digits	Effective range: 1 ≤ n ≤ 16 (16 bits), 1 ≤ n ≤ 32 (32 bits)		

2. Operands

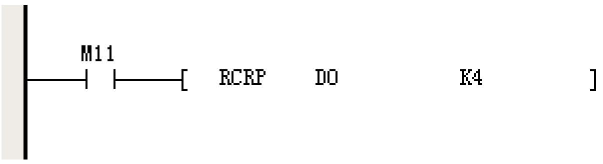
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

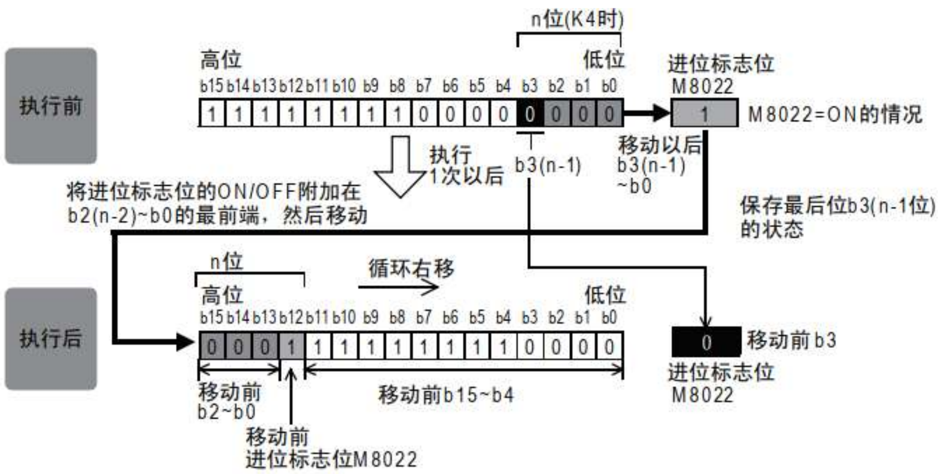
- Move the contents of D and carry flag m8022 to the right by N-bits.
- This instruction generally uses the pulse execution type instruction.
- When it is a 32-bit instruction, the register variable occupies a total of 2 units of the subsequent adjacent address.
- Only K4 (16bit) and K8 (32bit) are valid when kni, KM and KS are specified in D.

【Example】



When M11 is on, the number band carry cycle of d0 shifts four bits to the right;

The last bit remains in the carry flag bit m8022.



RCL Loop left shift with carry Rotate left with carry bit

1. Instruction form

When the driving condition is established, the data carry (m8022) in D moves n bits to the left, and the moved high carry (m8022) circulates into the low position of D.

RCL D n			Rotate left with carry bit	Instruction execution	
D	Device to be cycled	Data store word soft element address		16Bit instruction （5step）	32Bit instruction （9step）
N	Single move digits	Effective range: 1 ≤ n ≤ 16 (16 bits), 1 ≤ n ≤ 32 (32 bits)		RCL Continuous execution RCLP Pulse execution	DRCL Continuous execution DRCLP Pulse execution

2. Operands

Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

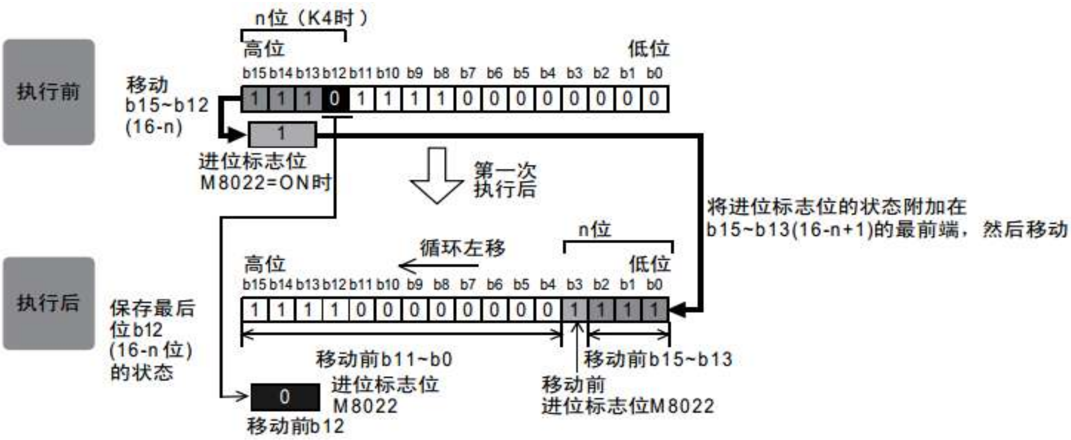
Function Description:

- Move the content of D and carry mark m8022 to the left by N bits.
- This instruction generally uses the pulse execution type instruction. When it is a 32-bit instruction, the register variable occupies a total of 2 units of the subsequent adjacent address.
- Only K4 (16bit) and K8 (32bit) are valid when kni, KM and KS are specified in D.

【Example】

When M1 is on, the number band carry cycle of d0 shifts four bits to the left;

The last bit remains in the carry flag bit m8022.



4.5.5 Other data processing

Other data processing	
SWAP	Upper and lower byte exchange
BON	ON bit judgement
SUM	ON is the total number.
RND	Generate random data
XCH	Data exchange

SWAP Upper and lower byte exchange

1. Instruction form

Exchanges the high and low byte values of the specified variable s with each other.

<table>
<tr><td colspan="3">SWAP S</td><td>Upper and lower byte exchange</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S</td><td>Operands</td><td colspan="2">Data storage unit to perform upper / lower byte interchange</td><td>16Bit instruction（3step）
SWAP Continuous execution
SWAPP Pulse execution</td><td>32Bit instruction（5step）
DSWAP Continuous execution
DSWAPP Pulse execution</td></tr>
</table>

2. Operands

<table>
<tr><td rowspan="2">Operands</td><td colspan="7">Bit soft component</td><td colspan="15">Word soft element</td></tr>
<tr><td colspan="7">System · user</td><td colspan="5">System · user</td><td colspan="5">Digit assignment</td><td colspan="2">Indexed address</td><td colspan="2">Constant</td><td>Real number</td></tr>
<tr><td>S</td><td>X</td><td>Y</td><td>M</td><td>T</td><td>C</td><td>S</td><td>SM</td><td>D</td><td>R</td><td>T</td><td>C</td><td>SD</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnSM</td><td>V,Z</td><td>Modify</td><td>K</td><td>H</td><td>E</td></tr>
</table>

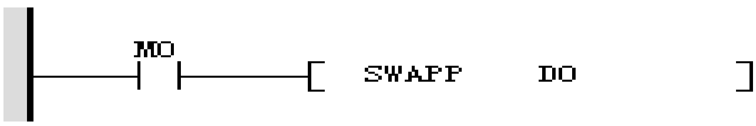
Note: soft components with gray shading indicate that they can be supported.

Function Description:

- This instruction exchanges the high and low byte values of the specified variable s with each other.
- In 16bit instruction, the values of high 8 bits and low 8 bits are exchanged.
- In 32bit instruction, the values of high-8 bits and low-8 bits of the two registers exchange with each other.

  Note that this instruction generally uses the pulse execution type instruction, otherwise, if the continuous execution instruction is used, the program will exchange every time it scans.

【Example】



D0 H（8-15bin） ⇆ L（0-7bin）

When M0 changes from OFF to ON, the contents of the high byte H and the low byte L in D0 are exchanged.

Bon on bit judgment

1. Instruction form

When the driving condition is established, the k-th bit state of binary data in the source address s controls the D-state.

BON S D n		ON bit judgement	Instruction execution	
S	Source data	Data or data storage word soft element address	16Bit instruction (7step) BON Continuous execution BONP Pulse execution	32Bit instruction (13step) DBON Continuous execution DBONP Pulse execution
D	Controlled bit	Controlled bit element		
N	Finger positioning	Bit 1 ≤ n ≤ 15 (16 bits) specified in source address s, 1 ≤ n ≤ 31 (32 bits)		

2. Operands

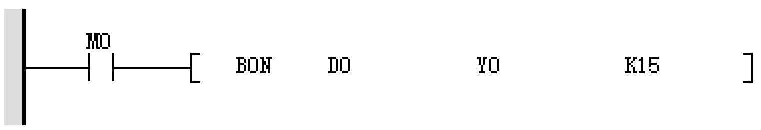
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

Judge the n-th position of S and store the result in D.

【Example】



When M0 is on, the 15th bit (15bin is the symbol bit) of d0 is 1, Y0 is set, and Y0 is reset when bin15 is 0. When M0 changes from on to off, Y0 remains in the previous state.

Sum on bits

1. Instruction form

When the driving condition is established, the number of "1" in binary data represented in source address s is counted, and the statistical results are stored in D.

SUM S D		Total number of ON bits	Instruction execution	
S	Statistical number	Data or data storage word soft element address	16Bit instruction (5step) SUM Continuous execution SUMP Pulse execution	32Bit instruction (9step) DSUM Continuous execution DSUMP Pulse execution
D	Statistical results	Data storage software address		

2. Operands

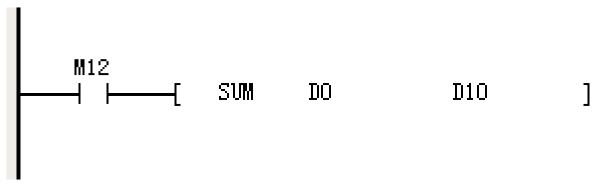
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- Calculate the number of bits of 1 in the binary value of S and store it in D.
- In the case of dsum and dsump instructions, the number of 1 in 32 bits of (s + 1, s) is written to D, and all d + 1 are 0.
- If all the bits in s are zero, then the zero flag bit m8020 will be set to on.

【Example】



D0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	1	1
D1	7															

RND Generate random data

1. Instruction form

An instruction that produces a random number.

RND D			Generate random data	Instruction execution
D	Destination address	Soft components for saving random numbers		16Bit instruction（3step） RND Continuous execution RNDP Pulse execution

2. Operands

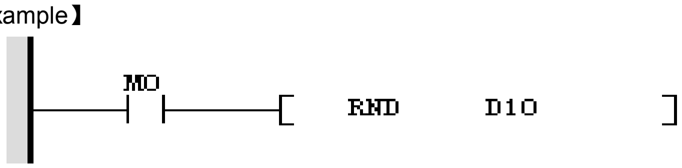
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- This command generates pseudo-random numbers from 0 to 32767 and stores them as random numbers in [D].
- Please write a non-zero value (- 2147483648 ～ 2147483647) to (d8311, d8310) as the initial value only once during stop → run.

【Example】



The resulting random number is stored in D10.

XCH data exchange (under development)

1. Instruction form

When the driving conditions are met, the data in S and D are exchanged.

XCH S D			Date exchange	Instruction execution	
S	Data 1	Data storage for data exchange word soft element 1		16Bit instruction (5step) XCH Continuous execution XCHP Pulse execution	32Bit instruction (9step) DXCH Continuous execution DXCHP Pulse execution
D	Data 2	Data storage for data exchange word soft Element 2			

2. Operands

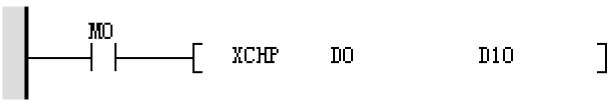
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

Contact drive is required with 2 operation variables to exchange the values of S and D with each other.

【Example】



When M0 changes from off to on, the data in d0 and D10 are exchanged with each other.

When the special variable m8160 = 1, and D and s are at the same address, the completed operation will be the exchange of high-8 bits and low-8 bits, the same with 32-bit instructions, and the completed operation will be the exchange of high-8 bits and low-8 bits. Equivalent to the operation of the swap instruction. Generally, it is implemented with the swap instruction.

4.6 Clock Instruction

Clock read and write	
TRD	Clock data reading
TWR	Clock data write

TRD Clock data reading

1. Instruction form

Read the year / month / day / hour / minute / second / week of the PLC built-in real-time clock, and save the 7 data in the specified register.

TRD D			Clock data reading	Instruction execution
D	Time storage first address	The starting storage unit of time takes up 7 consecutive variable units, and the address stores data from year, month, day, hour, minute, second, week, etc. in turn		16Bit instruction（3step） TRD Continuous execution TRDP Pulse execution

2. Operands

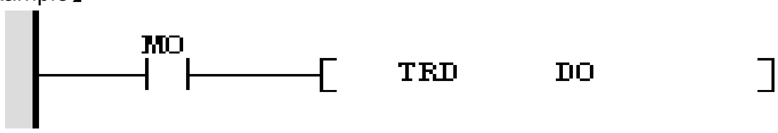
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- This instruction is to read the year / month / day / hour / minute / second / week of the PLC built-in real-time clock, and save the 7 data in the specified register.
- Where D is the starting storage unit to save the read time, occupying a total of 7 consecutive variable units, and storing the data from second, minute, hour, day, month, year, week and so on from small to large.

【Example】



When M0 is on, the operation of the command is as follows:

Project	System variables		D after operation
Year (2000~2099)	D8018	→	D0
Month (1~12)	D8017	→	D1
Day (1~31)	D8016	→	D2
Hour (0~23)	D8015	→	D3
Minute (0~59)	D8014	→	D4
Second (0~59)	D8013	→	D5
Week [0(day)~6]	D8019	→	D6

Note: in general, the clock of PLC should be used. First read out the clock with TDR instruction and put it into D register. Do not directly use the values of d8013 ~ d8019.

TWR clock data write

1. Instruction form

This instruction is to write 7 data of the specified clock data s (including year / month / day / hour / minute / second / week) into the real-time clock data built in PLC.

TWR D			Clock data write	Instruction execution
D	Time write data first address	In order to save the starting storage unit of read time, 7 consecutive variable units are occupied, and the address is stored from small to large: year, month, day, hour, minute, second, week and other data		16Bit instruction（3step） TWR Continuous execution TWRP Pulse execution

2. Operands

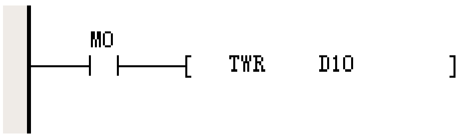
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

This instruction is to write 7 data of the specified clock data s (including year / month / day / hour / minute / second / week) into the real-time clock data built in PLC.

Where: s is the starting storage unit for saving the read time, occupying 7 consecutive variable units in total, and storing the data of year, month, day, hour, minute, second and week from the smallest to the largest【Example 一】



When M0 is on, the operation of the command is as follows:

Project	System variables		D after operation
Year (2000~2099)	D0	→	D8018
Month (1~12)	D1	→	D8017
Day (1~31)	D2	→	D8016
Hour (0~23)	D3	→	D8015
Minute (0~59)	D4	→	D8014
Second (0~59)	D5	→	D8013
Week [0(Day)~6]	D6	→	D8019

Note: Note: when writing the clock, all 7 data are written. When setting the value in advance, a certain variable cannot be missing. For example, if the week is not written, the default value is 0, which is Sunday. If the month is not assigned in advance, then the month variable is 0, which is considered by PLC to be wrong, and the Modify of the clock is invalid.

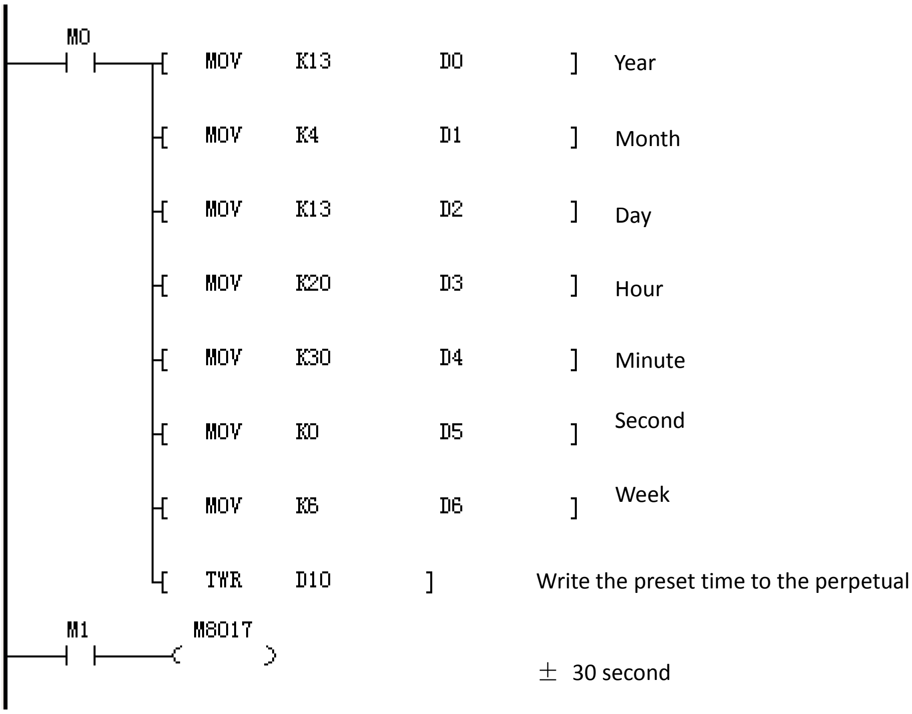
Every time m8017 is on, PLC internal clock will make ± 30 second correction action. Here correction means that when the second hand of PLC internal clock is between 1 and 29, it will be automatically classified as

When "0" second and minute hand remain unchanged, 30-59, it will also be automatically classified as "0" second, minute hand plus 1 minute.

Setting m8015 on can stop clock timing.

The timing method of PLC internal clock is as follows.

【Example 2】



Adjust the current time of PLC to 08:30:00 on April 13, 2013, Saturday; write it to d0-d6 some time in advance, when M0 is set to on, set this time into PLC.

M8017 can be adjusted for plus or minus 30 seconds at the instant of on

Note：

- In general, it is necessary to modify the clock of PLC, write the clock to d8013-d8019 with TWR instruction, and write m8015 to d8013-d8019 with MOV instruction.
- In the case of the 4-digit mode of the Gregorian calendar, the setting value (80-99) is equivalent to (1980-1999) and (00-79) is equivalent to (2000-2079). For example: 80 = 1980, 99 = 1999, 00 = 2000, 79 = 2079.

4.7 Bit instruction for pulse location

Pulse output	
PWM	PWM output
PLSY	Pulse output
PLSR	With acceleration and deceleration pulse output
Pulse location	
PLSV	Variable speed pulse output
PLSV2	Variable speed pulse output with acceleration and deceleration
DRVI	Relative position positioning
DRVA	Absolute position positioning
ZRN	Origin regression
Refresh processing	
REF	I / O refresh
REFF	Input filter time adjustment

4.7.1 Pulse output

Pulse output	
PWM	PWM output
PLSY	Pulse output
PLSR	With acceleration and deceleration pulse output

[note]: for detailed use of PWM, Plsy and PLSR instructions, please refer to the chapter "high speed output and bit setting instruction".

4.7.2 Pulse location

Pulsa location	
PLSV	Variable speed pulse output
PLSV2	Variable speed pulse output with acceleration and deceleration
DRVI	Relative position positioning
DRVA	Absolute position positioning
ZRN	Origin regression
DVIT	Interrupt location
DPIT	Maximum fixed length interrupt bit instruction

[note]: for detailed use of PLSV, DRVI, drva and ZrN instructions, please refer to the chapter "high speed output and bit setting instruction".

4.7.3 Refresh processing

Refresh processing	
REF	I / O refresh
REFF	Input filter time adjustment

REF I / O refresh instruction

1. Instruction form

Update the input or output image store now

REF S n		I / O refresh	Instruction execution
S	Bit element first address	Input or output bit component first address to be refreshed	16Bit instruction（5step） REF Continuous execution REFP Pulse execution
N	Number of bit elements	Number of input or output bit elements to refresh	

2. Operands

Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description: Update the status of n components at the beginning of D address immediately. Because the access port of PLC is accessed by byte, it is required to: Address of s should be x0, X10 Y0、Y10、…… Number elements with the lowest order of 0;

The value of N must be a multiple of 8 (n = 8-256).

Normally, the status reading of input port x is performed before the start of each program scanning, and the status refreshing of output port y is performed in batches after the completion of each program scanning (execution to end), so that IO processing will have a certain delay. If the application needs the latest input information and wants to output the operation results immediately, you can use the immediate refresh instruction Ref.

It can be used between for ~ next instructions, CJ instructions, etc.

It can be used to refresh the input and output in interrupt subroutine to obtain the latest input information and output the operation results in time.

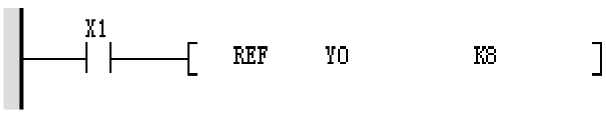
The actual state change delay of input port depends on the filtering time of input element. X0-x1 has digital filtering function. The filtering time can be set in the range of 0-60ms (Reff instruction). The rest IO ports are hardware filtering, and the filtering time is about 10ms.

The actual output port state change delay is determined by the response time of the output elements (such as relays). The output contact in the output refresh will act after the output relay (transistor) response time. The response lag time of relay output type is about 10ms (maximum 20ms), that of transistor output type is about 10 μ s and that of common point is about 0.5ms.

【Example 1】

When X1 is on, the state of 8 input points of x0-x7 will be read immediately, the input signal will be updated, and no input delay will be generated.

【Example 2】



When X1 is on, the status of Y0 ~ Y7 will be refreshed immediately, and the output signal will be updated immediately. You do not have to go to the end command to output.

REFF Input filter time adjustment

1. Instruction form

When the driving condition is satisfied, set the filter time constant of x0 ~ X1 input port to n milliseconds.

REFF n			Filter parameter adjustment	Instruction execution
N	Filtering time	Units：ms		16Bit instruction（5step） REFF Continuous execution REFFP Pulse execution

2. Operands

Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description: Set the filter time constant of x0 ~ X1 input port to n.

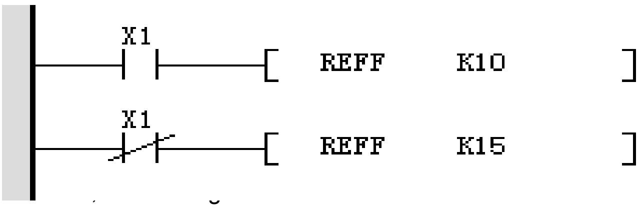
In the programmable controller, x0 ~ X1 uses a digital filter, the default filter time constant is set by d8020, and the d8020 can be changed to 0 ~ 60ms by the Reff command.

The rest of the X-port only has hardware RC filtering, and the filtering time constant is about 10ms, which cannot be modified;

When a high-speed counter or x input interrupt function is used, the filtering time of the relevant port is automatically the shortest time, and the filtering time of the irrelevant port is still the original set value.

The MOV command can also be used to directly assign values to d8020 to change the filtering time.

【Example】



When

When

4.8 Communication

Communication instructions	
MODBUS	MODBUS Communicaiton

MODBUSMODBUS Communication

1. Instruction form

MODBUS Communication reading and writing instructions

MODBUS S1 S2 n D		MODBUS Communication	Instruction execution
S1	Mailing address, function code	Slave address (high byte), communication command (low byte, defined according to Modbus Protocol)	16Bit instruction (9step) MODBUS Continuous execution
S2	Slave data first address	Register start address number of access slave	
N	Data length	Read or write data length	
D	Host data first address	Starting address of storage unit for reading or writing data, occupying subsequent address unit, length determined by n	

2. Operands

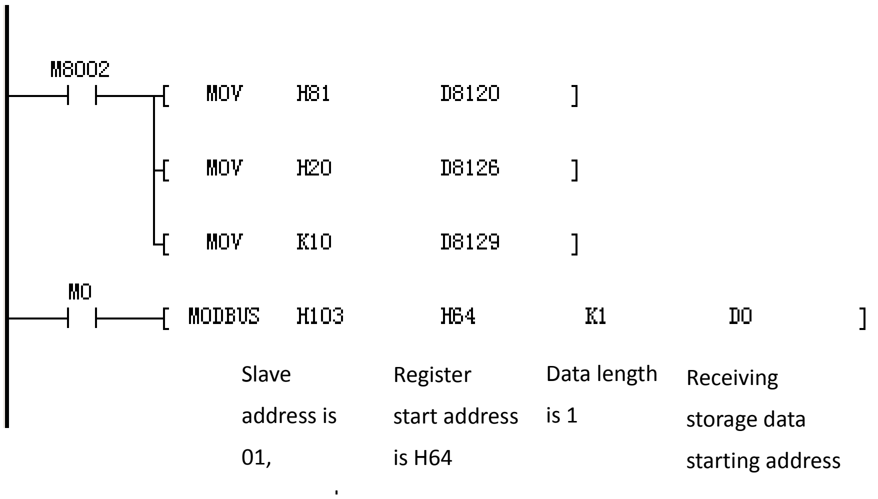
Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
N	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

- S1: slave address (high byte), communication command (low byte, defined according to Modbus Protocol).
- S2: register start address number of access slave station.
- N: read or write data length.
- D: starting address of storage unit for reading or writing data, occupying subsequent address unit, and the length is determined by n.

【Example】



When M0 is set to on, PLC continuously reads the value of register with address of H64 in slave 1, and stores the data in d0 unit.

4.9 Peripheral device

PID Operation	
PID	PID Operation
Other peripheral instructions	
ASC	ASCII Conversion

4.9.1 PID Operation

PID Operation	
PID	PID Operation

PID Operation

1. Instruction form

PID calculation is completed for the control of closed-loop control system

<table>
<tr><td colspan="2">PID S1 S2 S3 D</td><td>PID Operation</td><td>Instruction execution</td></tr>
<tr><td>S1</td><td>Target value</td><td>PID Set target value</td><td rowspan="4">16Bit instruction（9step）
PID Continuous execution</td></tr>
<tr><td>S2</td><td>Feedback value</td><td>Measured feedback value</td></tr>
<tr><td>S3</td><td>Operational parameters</td><td>Starting unit for storing operation results</td></tr>
<tr><td>D</td><td>Output value</td><td>PID output value storage unit</td></tr>
</table>

2. Operands

Operands	Bit soft component							Word soft element														
	System · user							System · user					Digit assignment					Indexed address		Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	E

Note: the soft components with gray background in the table indicate the soft components that can be supported.

3. Function and instruction description

1) PID function selection

Unit	Function	Significance		Remark
S3+0	Sampling period	Sampling and output refresh cycle		
S3+1	Operation mode selection	0x00**	Incremental PID	High 8-bit control instruction operation mode
		0x01**	Positional PID	

Choosing different function instructions, S3 unit has different meanings.

3. Incremental PID instruction

The function and setting method of each unit parameter value starting from S3 are described as follows:

Unit	Function	Setting instructions
S3	Sampling time	Setting range: 1 ~ 32767ms (default: 10ms)
S3+1	Action direction	Bit0:0 = positive action; 1 = reverse action Bit1: 0 = invalid input change alarm; 1 = valid input change alarm Bit2: 0 = invalid output change alarm; 1 = effective output change alarm Bit3: not available Bit4:0 = no action of self-tuning; 1 = perform self-tuning (this function is not provided in the current version) Bit5: 0 = the upper and lower limit setting of output value is invalid; 1 = the upper and lower limit setting of output value is valid Bit6 ~ bit15 not available In addition, do not set bit5 and bit2 on at the same time.
S3+2	Input filter constant	0 ~ 99 [%], 0 = no input filtering
S3+3	Proportional gain	0~32767[%]
S3+4	Integration time	0~32767（×100ms），0 = treated as ∞ (no integration)
S3+5	Differential gain	0~100[%]，0＝No differential gain
S3+6	Differential time	0~32767（×10ms），0＝Nondifferential processing
When Bit1 = 1, bit2 = 1 or bit5 = 1, S3 + (20-24) is occupied, which is defined as follows:		
S3+（7~19）	The internal processing of PID operation is occupied, and it should be cleared before operation	

S3+20	Input the alarm set value of change (increase side)	0~32767，（Valid when Bit1 = 1 of < Act >）
S3+21	Input the alarm set value of change (minus side)	0~32767，（Valid when Bit1 = 1 of < Act >）
S3+22	Output change (increase side) alarm set value	0~32767，（when <ACT> bit1=1 is valid ） Output upper limit setting value - 32768 ~ 32767, (< Bit1 = 0 for Act > and valid when bit5 = 1) Note 1
S3+23	Output change (minus side) alarm set value	0 ~ 32767 (valid when bit2 = 1 and bit5 = 0 for S3 + 1 < Act >) Output lower limit setting value - 32768 ~ 32767, (< Bit1 = 0 for Act > and valid when bit5 = 1) Note 1
S3+24	Alarm output	Bit0 input variation (increasing side) overflow Bit1 input variation (minus side) overflow Bit2 output change (increasing side) overflow Bit3 output change (minus side) overflow (< Act > Bit1 = 1 or bit2 = 1)
S3+25	Internal processing occupation of PID operation	

Note 1: when the output is limited, the PID output is the value after the limit. If the output changes after the limit, then the change is also based on the output value after the limit.

4. Position PID instruction (S3 + 1 select 0x01 * *)

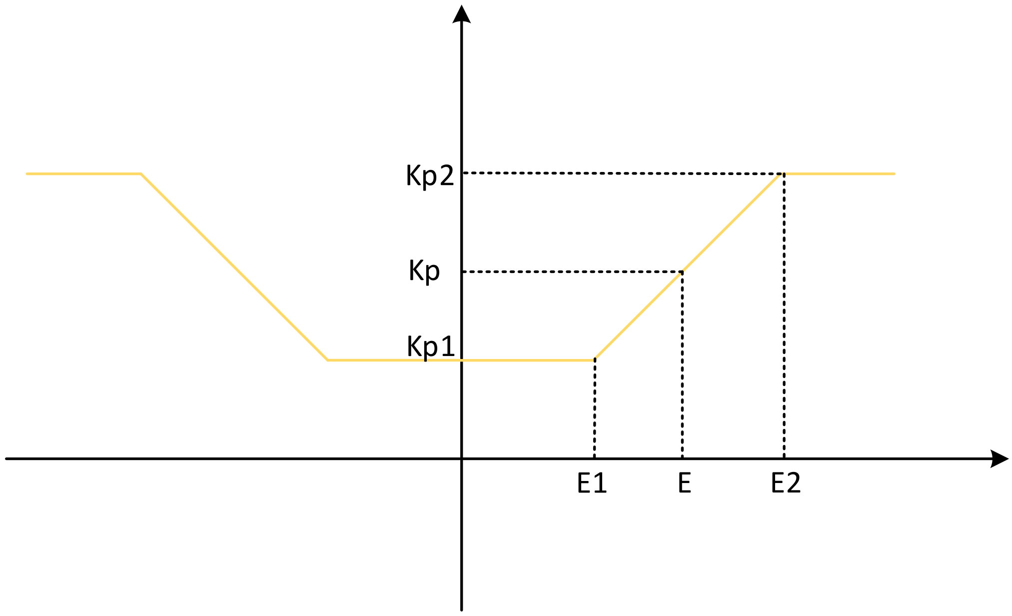
Unit	Function	Setting Instructions
S3	Sampling time	Setting range 1 ~ 32767ms (default 10ms)
S3+1	control model	0x0100：Forward 0x0101：Reverse
S3+2	Proportional gain 1	0~32767[%]
S3+3	Integral gain 1	0~32767[%]
S3+4	Differential gain 1	0~32767[%]
S3+5	Deviation dead zone	0 ~ 32767; 0: not effective; non-0: the deviation is 0 if it is less than this value
S3+6	Upper limit of output	-32768~32767；Output Max
S3+7	Lower limit of output	-32768~32767；Output min
S3+8	Upper limit of integration	-32768~32767；Maximum cumulative integral, note 1
S3+9	Integral lower limit	-32768~32767；Cumulative integral minimum, note 1
S3+10 S3+11	Accumulated points	32-bit floating point number
S3+12	Last deviation	-32768~32767；
S3+13	Proportional gain 2	0~32767[%]
S3+14	Integral gain 2	0~32767[%]
S3+15	Differential gain 2	0~32767[%]
S3+16	Parameter switching conditions	0: do not switch; 1: switch according to deviation, note 2

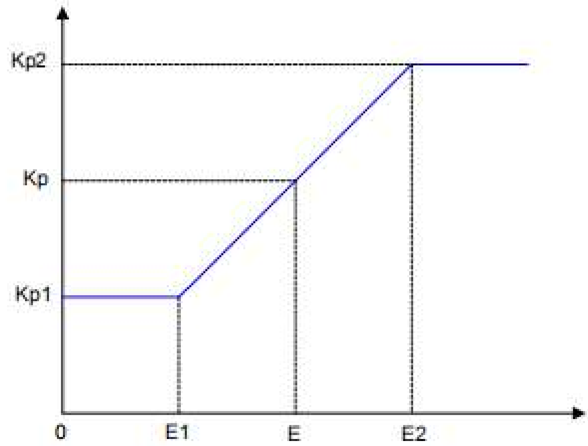
S3+17	Lower deviation limit ,note 3	0~32767；Deviation starting point or user-defined switching starting point
S3+18	Upper deviation limit, note 3	0~32767；Deviation end point or custom switch end point
S3+19	Reserve	
S3+20~ S3+26	Internal operation use	

Note 1: when the upper and lower limits of integration are set to 0, the upper and lower limits of integration will take effect according to the upper limit 32767 and the lower limit - 32768.

Note 2: deviation switching principle (proportional gain KP switching as an example).

Note 3: the lower limit and upper limit of deviation are absolute values of deviation.





Kp1	(S3 + 2)
Kp2	(S3 + 13)
E1	(S3 + 17)
E2	(S3 + 18)
E	切换参考

当 E ≤ E1，Kp = Kp1；

当 E1 < E < E2，Kp = (Kp2-Kp1)*E/(E2-E1)；

当 E ≥ E2，Kp = Kp2；

S3 + 16	0	不切换
	1	E = |Sv - Pv|
	2	E = S3 + 19

Error code	Content represented by error
6780	Unreasonable setting of sampling time
6781	Retain
6782	Input filter object is unreasonable
6783	Unreasonable proportion coefficient
6784	Unreasonable integral coefficient
6785	Unreasonable differential coefficient
6786	Output limit is abnormal (output lower limit is greater than upper limit)

4.9.2 Other peripheral instructions

Other peripheral instructions	
ASC	ASCII conversion

ASCASCII code conversion

1. Instruction form

When the driving condition is correct, the string input by the computer to S1 is converted into ASCII code and stored in the register with D1 as the first address.

ASC S D		ASCII conversion	Instruction execution
S	Data source	The maximum allowed length of an English string to perform ASCII conversion is 8 characters	16Bit instruction （11step） ASC Continuous execution
D	Conversion result	Stores the starting unit number of ASCII code, occupying the following 4 (m8161 = 0) or 8 variable units (m8161 = 1)	

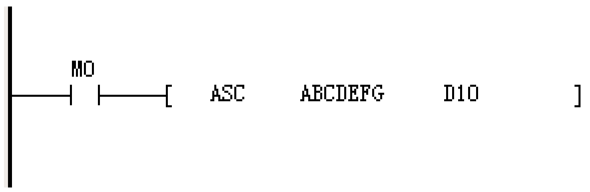
2. Instruction form

Operands	Bit soft component							Word soft element															
	System · user							System · user					Digit assignment					Indexed address			Constant		Real number
S1	User input corresponding letters																						
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: soft components with gray shading indicate that they can be supported.

Function Description:

【Example】



If m8161 = off, after the numeric string is converted into ASCII code, every 2 characters / 1 byte is saved in D in the order of low 8 bits and high 8 bits.

	High 8-bit	Low 8-bit
D	42（B）	41（A）
D+1	44（D）	43（C）
D+2	46（F）	45（E）
D+3	00	47（F）

If m8161 = on, after the numeric string is converted to ASCII code, it is transferred to the lower 8 bits (1 byte) of D in turn.

	High 8-bit	Low 8-bit
D	00	41（A）
D+1	00	42（B）
D+2	00	43（C）
D+3	00	44（D）
D+4	00	45（E）
D+5	00	46（F）

Appendix: ASCII code comparison table

10 binary digit	ASCII （16 hexadecimal number）
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39

10 binary digit	ASCII （16 hexadecimal number ）
0	30
1	31

English letter	ASCII （16 hexadecimal number ）	English letter	ASCII （16 hexadecimal number ）
A	41	N	4E
B	42	O	4F
C	43	P	50
D	44	Q	51
E	45	R	52
F	46	S	53
G	47	T	54
H	48	U	55
I	49	V	56
J	4A	W	57
K	4B	X	58
L	4C	Y	59
M	4D	Z	5A

4.10 Electronic cam command

Electronic cam command	
CAMWR	Write electronic cam data
CAMRD	Read the data of electronic cam
CAMSP	Spindle / slave position calculation

Please refer to "Chapter 6 electronic cam"

Chapter 5 High speed output and bit instruction

5.1 Instruction overview

The fixed bit instruction and trajectory control of v5-m104 standard type are realized by application instruction. The features are as follows:

- High speed output frequency range: shaft port is 1Hz ~ 3MHz, high speed port is 1Hz ~ 200kHz;
- Support trapezoid acceleration and deceleration, S-type acceleration and deceleration;

5.1.1 High speed output instruction attribute table

instructions	Pulse direction output	Trapezoid acceleration and deceleration	S curve acceleration and deceleration	Acceleration and deceleration time is set separately	Change frequency in operation	Changing the number of pulses in operation	Reversing in operation	Speed / position control
PLSY					√	√（M）		speed
PLSV	√				√		√	position
PLSV2	√	√		√（M）	√		√	Speed + position
ZRN		√		√（M）			√	speed
PLSR		√		√（M）		√（M）		speed
DRVA	√	√	√（M）	√（M）		√（M）		speed
DRVI	√	√	√（M）	√（M）		√（M）		position
DVIT	√	√		√（M）				position
DPIT	√	√		√（M）				position
PWM					√			Speed + position

- In the attribute table of each instruction, "√" indicates that the attribute is possessed / supported, and the blank space in the table indicates that the attribute is not possessed / supported;
- "√ (m)" in the attribute table of each instruction indicates that special soft components need to be set to enable the function;
- Whether the high-speed output instruction has acceleration and deceleration is determined by the attribute of the instruction itself, regardless of the acceleration and deceleration time. Fixed bit instruction acceleration and deceleration time range: 10ms-5000ms (interpolation instruction range acceleration and deceleration time range: 10ms-500ms), less than the minimum range value, calculated by the minimum range value, greater than the maximum range value, calculated by the maximum range value.

5.1.2 Description of pulse output port

V5-mc104 has 6 high-speed pulse output ports to choose from, including 4 shaft ports and 2 high-speed transistor output ports. See the following table for specific port definitions:

Output port No	Pulse output port	Pulse output type	structure	Pulse output frequency
1	Y300：PUL+/- Y301：DIR+/-	Dual differential output + 5V Single open collector output	Axle port structure Axle port	1Hz~3MHz
2	Y304：PUL+/- Y305：DIR+/-			
3	Y310：PUL+/- Y311：DIR+/-			
4	Y314：PUL+/- Y315：DIR+/-			
5	Y0	Pulse output type	IO terminal port	1Hz~200kHz
6	Y1			

5.1.3 Special soft element of pulse output port

The high-speed output instruction involves many special registers and relays. The special soft elements related to the pulse are defined as follows:

1）The definition of shaft mouth soft element is as follows:

axis 1	axis 2	axis 3	axis 4	Y0	Y1	Attribute
SD61	SD161	SD261	SD761	SD861	SD961	Pulse output form setting
Y302	Y306	Y312	Y316			0: direction / pulse
X301	X305	X311	X315			1:AB orthogonal
SD56	SD156	SD256	SD756			2:CW/CCW
SD38	SD138	SD238	SD738	SD838	SD938	Shaft opening enable output [1]

Note [1]: enable by setting soft element on.

[3]: the shaft port pulse input count is used to count the encoder, and the pulse input form is fixed as AB orthogonal form.

2）The definition of special M element is as follows:

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8340	M8360	M8380	M8400	M8420	M8440	In pulse output
M8341	M8361	M8381	M8401	M8421	M8441	The output of zero clearing signal such as ZrN is effective
M8342	M8362	M8382	M8402	M8422	M8442	Retain
M8343	M8363	M8383	M8403	M8423	M8443	Retain
M8344	M8364	M8384	M8404	M8424	M8444	Retain
M8345	M8365	M8385	M8405	M8425	M8445	Retain
M8346	M8366	M8386	M8406	M8426	M8446	Retain
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stop
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration and deceleration are set separately / pulse change

						is valid
M8352	M8372	M8392	M8412	M8432	M8452	Pulse output completion interrupt enable
M8353	M8373	M8393	M8413	M8433	M8453	Retain
M8354	M8374	M8394	M8414	M8434	M8454	Dszrinstruction execution exception end flag bit
M8355	M8375	M8395	M8415	M8435	M8455	Plsv2 acceleration flag
M8356	M8376	M8396	M8416	M8436	M8456	Plsv2 deceleration sign
M8357	M8377	M8397	M8417	M8437	M8457	Retain
M8358	M8378	M8398	M8418	M8438	M8458	Retain
M8359	M8379	M8399	M8419	M8439	M8459	Retain

1）The definition of special D element is as follows:

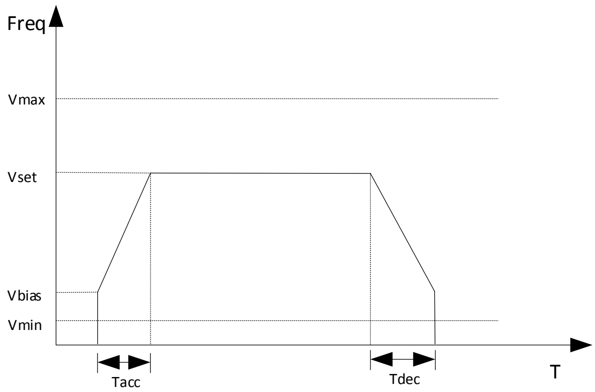
Y300	Y304	Y310	Y314	Y0	Y1	Attribute
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed when set separately (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8344	D8364	D8384	D8404	D8424	D8444	Origin regression speed (Hz)
D8345	D8365	D8385	D8405	D8425	D8445	
D8346	D8366	D8386	D8406	D8426	D8446	Creeping speed (Hz) when set separately
D8347	D8367	D8387	D8407	D8427	D8447	Base speed at individual setting (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time when set separately (MS)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time when set separately (MS)
D8350	D8370	D8390	D8410	D8430	D8450	Reset soft element No
D8500/D8501						Maximum speed at unified setting (Hz)
D8502						Base speed at unified setting (Hz)
D8503						Acceleration and deceleration time under unified setting (MS)

5.1.4 Output frequency and acceleration and deceleration time

The frequency of pulse output and acceleration and deceleration time shall follow the following principles:

- The output pulse frequency of the controller should be between the maximum frequency and the minimum frequency;
- The output frequency of the controller at the initial acceleration and the final deceleration stage is higher than the frequency set by the base speed;
- The acceleration time is the time when the base speed is accelerated to the set speed;
- The acceleration time is the time when the set speed decelerates to the base speed.

The relationship between output frequency and acceleration / deceleration time is shown in the figure below:



Among them:

Vset: pulse output frequency set by the user;

Vmin: minimum speed;

Vbias: base speed, special soft element setting, factory default is 500Hz;

Vmax: maximum frequency, special soft element setting, shaft port factory default is 3MHz, port factory default is 200kHz;

TACC: acceleration time, from base speed to set speed;

TDEC: deceleration time, from the set speed to the base speed.

5.2 List of positioning instructions

5.2.1 List of positioning instructions

Instruction	Explain
PLSY	Pulse output command
PLSV	Variable speed pulse output command
PLSV2	Variable pulse output command with acceleration and deceleration
PLSR	Pulse output command with acceleration and deceleration
DRVA	Absolute position control command
DRVI	Relative position control command

ZRN	Origin regression instruction
DSZR	Dogsearch origin regression
DVIT	Interrupt location
DPIT	Maximum fixed length interrupt bit instruction

5.2.2 Plsy pulse output command

1.Instruction form

Pulse output command to output the set number of pulses at the specified pulse output frequency.

PLSY S1 S2 D		Pulse output command	Instruction execution	
S1	Output frequency (Hz)	Set pulse output frequency	16Bit instruction （7step） PLSY Continuous execution	32Bit instruction （13step） DPLSY Continuous execution
S2	Number of outputs (PLS)	Set number of pulse output		
D	Output port	high speed pulse output port		

2. Operands

Operands	Bit soft component							Word soft element															
	System · user							System · user					Digit assignment					Indexed address			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: the soft components with gray background in the table indicate the soft components that can be supported.

① S1 is the pulse output frequency, unit: Hz. For 16bit instruction (Plsy), the setting range is 10 ~ 32767; for 32bit instruction (dplsy), the setting range is 10 ~ 3000000; the value of S1 can be modified during instruction execution. Assuming the set frequency is more than 3000000, the system will be limited to 3000000.

② S2 is the number of pulse outputs, the unit is pls, for 16bit instruction (Plsy), the setting range is - 32768 ~ 32767; for 32bit instruction (dplsy), the setting range is - 2147483648 ~ 2147483647; the value of S2 can be modified during instruction execution.

③ when the value of S2 is set to 0, it means that the positive pulse is always sent at the frequency of S1, and when the value of S2 is set to 0x80000000 (32bit instruction), it means that the negative pulse is always sent at the frequency of S1, and the pulse output will stop only when the energy flow of the instruction is off.

④ D is the pulse output port, and y300 / y304 / y310 / y314 / Y0 / Y1 can be specified.软元件

The function of the soft elements related to the pulse output is described in detail below.

◆ Pulse output count

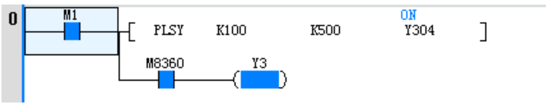
Y300	Y304	Y310	Y314	Y0	Y1	attribute
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output counter records the number of pulses output by the port. This element has the characteristics of D soft element. It can be cleared by relevant command or power off. When the controller state is run - > off, the element will not be cleared. The axis cannot be cleared during operation.

◆ In pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8340	M8360	M8380	M8400	M8420	M8440	In pulse output

As shown in the figure below, when y304 pulse is output, m8360 is set. When the pulse is output, m8360 is reset automatically.



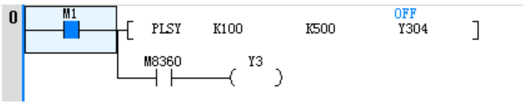
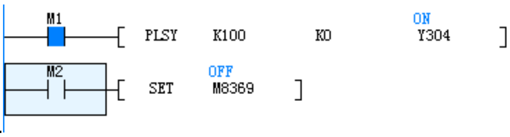


Figure a: in pulse output Figure B: pulse output complete

◆ Pulse output stop

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stop

When the pulse output stop flag is set, the pulse output stops. Even if the energy flow is valid, the pulse will not output. The effect of the soft element is shown in the figure below. Before the soft element is set, the pulse is output normally. After the soft element is set, even if the energy flow is effective, the pulse will not be output.



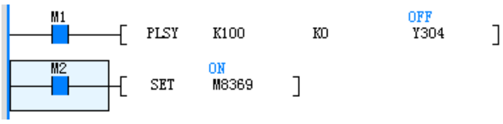


Figure a: before pulse output stop setting Figure B: after the pulse output stops setting

Pulse output complete

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8029						Pulse output complete

When the pulse output is completed, the corresponding m8029 will be set, but the completion of the pulse output does not affect the completion flag of other pulse instructions. As shown in the figure below, after the completion of the first pulse instruction execution, M10 is set, but M11 and M12 are not set.

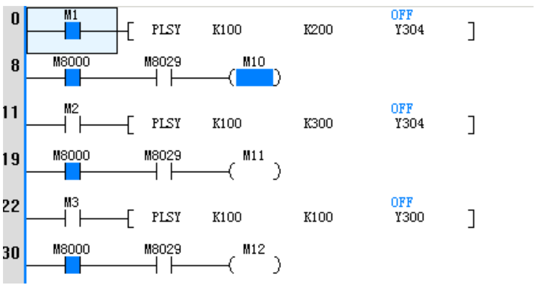


Figure A: m8029 soft element action

◆ Pulse output complete interrupt

If you want to enter the interrupt when the pulse output is completed, different output completion interrupt enabling soft elements can be opened for different Y ports. See the table below. If you want to enter the interrupt after y300 pulse output, you need to set m8352 to 1.

Y300	Y304	Y310	Y314	Y0	Y1	attribute
M8352	M8372	M8392	M8412	M8432	M8452	Output completion interrupt enable

◆ Acceleration and deceleration are not supported.

3. Changing parameters in operation

◆ In instruction execution, it is allowed to modify the number of pulse outputs

◆ Before the change, special soft components need to be set, as shown in the table below. When the effective soft components of pulse change are effective, the number of pulses can be changed。

Y300	Y304	Y310	Y314	Y0	Y1	attribute
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration and deceleration time is set separately / pulse change is valid

① when the number of pulse output is changed from 0 to non-0, the output mode changes from speed mode to position mode. After the change, the controller sends out the set number of pulses and stops the output.

② the number of pulse output cannot be changed from non-zero value to 0.

③ the number of changed pulses shall be greater than the current pulse position.

④ in the follow-up process, the position is still the absolute position relative to the first entry after multiple Modifys.

◆ In instruction execution, it is allowed to modify the pulse output frequency

① in the output process of the command, it is allowed to modify the frequency of the pulse output. The changed pulse output frequency can be greater than or smaller than the current pulse output frequency.

② the output direction of the command is controlled by the number of output pulses,

When the number of pulse output is set to be greater than 0, it is a positive pulse output,

When the number of pulse output is set to be less than 0, it is a negative pulse output.

③ pulse output frequency can only be set to a value greater than 0. That is to say, if the number of pulse output is set to 0 and Plsy instruction is used as speed mode, only forward speed instruction can be issued.

5.2.3 PLSV variable pulse output command

1. Instruction form

Variable pulse output command, specified pulse frequency and direction output.

<table>
<tr><td colspan="3">PLSV S1 D1 D2</td><td>Variable speed pulse output command</td><td colspan="2">Instruction execution</td></tr>
<tr><td>S1</td><td>Output frequency (Hz)</td><td colspan="2">Set pulse output frequency</td><td rowspan="3">16Bit instruction （7step） PLSV Continuous execution</td><td rowspan="3">32Bit instruction （13step） DPLSV Continuous execution</td></tr>
<tr><td>D1</td><td>Output port</td><td colspan="2">High speed pulse output port</td></tr>
<tr><td>D2</td><td>Output direction</td><td colspan="2">Pulse operation direction</td></tr>
</table>

2. Operands

Operands	Bit soft component							Word soft element																
	System · user							System · user					Digit assignment						Indexed address			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: the soft components with gray background in the table indicate the soft components that can be supported.

① S1 is the pulse output frequency, in Hz. For 16bit instruction (PLSV), the setting range is - 32768 ~ - 1; 1 ~ 32767; for 32bit instruction (dplsv), the setting range is - 3000000 ~ - 1; 1 ~ 3000000; the value of S1 can be modified during instruction execution.

② D1 is the pulse output port, and y300 / y304 / y310 / y314 / Y0 / Y1 can be specified.

③ D2 is the operation direction port or bit variable. When the output pulse is positive, the element state is on. When the pulse direction is negative, the element state is off.

◆ 3. soft components

The function of the soft elements related to the pulse output is described in detail below.

Pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	attribute
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output counter records the number of pulses output by the port. This element has the characteristics of D soft element. It can be cleared by relevant command or power off. When the controller state is run - > off, the element will not be cleared.

◆ In pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8340	M8360	M8380	M8400	M8420	M8440	In pulse output

As shown in the figure below, when y304 pulse is output, m8360 is set. When the pulse is output, m8360 is reset automatically.

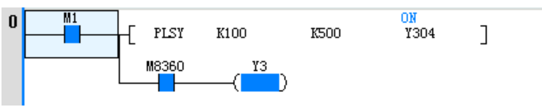


Figure a: in pulse output

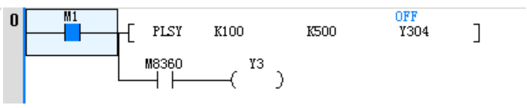
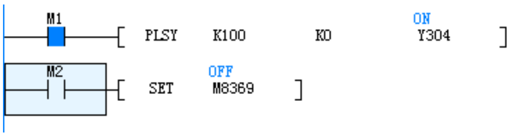


Figure B: pulse output complete

◆ Pulse output stop

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stop

When the pulse output stop flag is set, the pulse output stops. Even if the energy flow is valid, the pulse will not output. The effect of the soft element is shown in the figure below. Before the soft element is set, the pulse will output normally. After the soft element is set, even if the energy flow is effective, the pulse will not output.



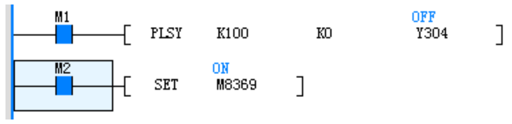


Figure a: before pulse output stop setting　　Figure B: after the pulse output stops setting

◆ No pulse output completion mark.

◆ No pulse output complete interrupt.

◆ No acceleration and deceleration time.

3. Change parameters during operation

In instruction execution, it is allowed to modify the pulse output frequency

① in the output process of the command, it is allowed to modify the pulse output frequency, which can be greater than or less than the current pulse output frequency.

② in the pulse output, it is allowed to change the pulse output direction, which can be realized by changing the set pulse output frequency; the output frequency is positive, the direction is positive, the output frequency is negative, and the direction is negative.

5.2.4 Plsv2 variable pulse output command with acceleration and deceleration

1.Instruction form

Variable pulse output command with acceleration and deceleration, specified pulse frequency and direction output.

PLSV2 S1 D1 D2		Variable pulse output command with acceleration and deceleration	Instruction execution	
S1	Output frequency (Hz)	Set pulse output frequency	16Bit instruction（7step）PLSV2 Continuous execution	32Bit instruction（13step）DPLSV2 Continuous execution
D1	Output port	High speed pulse output port		
D2	Output direction	Pulse operation direction		

2. Operands

Operands	Bit soft component							Word soft element															
	System · user							System · user					Digit assignment					Indexed address			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: the soft components with gray background in the table indicate the soft components that can be supported.

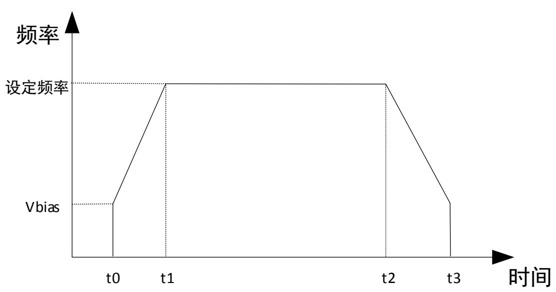
① S1 is the pulse output frequency, in Hz. For 16bit instruction (plsv2), the setting range is - 32768 ~ - 1; 1 ~ 32767; for 32bit instruction (dplsv2), the setting range is - 3000000 ~ - 1; 1 ~ 3000000; the value of S1 can be modified during instruction execution.

② D1 is the pulse output port, and y300 / y304 / y310 / y314 / Y0 / Y1 can be specified.

③ D2 is the operation direction port or bit variable. When the output pulse is positive, the element state is on. When the pulse direction is negative, the element state is off.

3. Pulse output

① the output diagram of plsv2 pulse is shown in the figure below. When the command energy flow is on, the pulse starts to output, and the frequency gradually accelerates from the base speed to the set speed.



② when the energy flow of the pulse output is off, the pulse output frequency will gradually slow down to the base frequency and then stop the output. Note that in the process of deceleration, when the flag in the pulse output is on, the command will not be driven again. After the output stops, when the energy flow is off - > on, the pulse output processing starts again.

3. soft components

The function of the soft elements related to the pulse output is described in detail below.

◆ pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output counter records the number of pulses output by the port. This element has the characteristics of D soft element. It can be cleared by relevant command or power off. When the controller state is run - > off, the element will not be cleared. The axis cannot be cleared during operation.

◆ In pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attribute

As shown in the figure below, when y304 pulse is output, m8360 is set. When the pulse is output, m8360 is reset automatically.

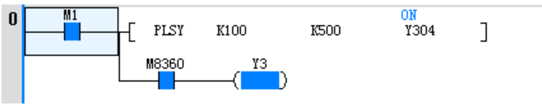


Figure a: in pulse output

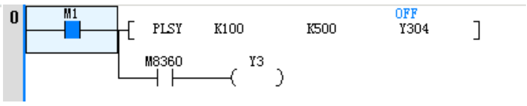


Figure B: pulse output complete

◆ Pulse output stop

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stop

When the pulse output stop flag is set, the pulse output stops. Even if the energy flow is valid, the pulse will not output. The effect of the soft element is shown in the figure below. Before the soft element is set, the pulse will output normally. After the soft element is set, even if the energy flow is effective, the pulse will not output.

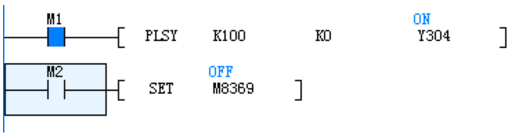


Figure a: before pulse output stop setting

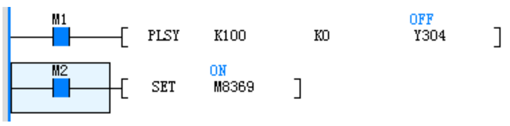


Figure B: after the pulse output stops setting

◆ No pulse output completion mark.

◆ No pulse output complete interrupt.

◆ Acceleration and deceleration time is set separately / pulse change is valid

If you want to have different acceleration and deceleration time for each output shaft, or you want to change the number of pulses during operation, you can set the following soft elements:

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration and deceleration time is set separately / pulse change is valid

Note: this soft element is a common soft element for "acceleration and deceleration are set separately" and "pulse change is effective".

When the above "acceleration and deceleration time is set separately" soft element is off, the following soft elements are used for corresponding shaft parameters:

Y300	attribute	attribute	attribute	attribute	attribute	Attribute
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (MS)

When the "acceleration and deceleration time is set separately" soft element of a shaft is on, the corresponding shaft parameters use the following soft elements:

Y300	Y304	Y310	Y314	Y0	Y1	attribute
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (MS)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (MS)

3. Modify parameters in operation

Change the pulse output frequency during operation, as shown in Figure A / B below. When changing the speed from V1 to V2 or V2 to V3, the speed will not change abruptly, but will accelerate or decelerate to the changed speed according to the set acceleration and deceleration time. In figure a, acceleration from V1 to V2 is the same as acceleration from base speed to V1; acceleration from V2 to V3 is the same as acceleration from V3 to base speed.

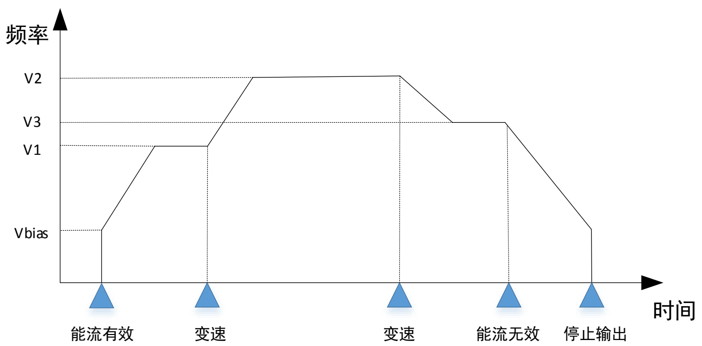


Figure a: plsv2 pulse output change diagram

In Figure B, the acceleration from base speed to V1 is the same as that from 0 to V2 and from 0 to V3; the acceleration from V1 to 0 is the same as that from V2 to 0 and from V3 to base speed.

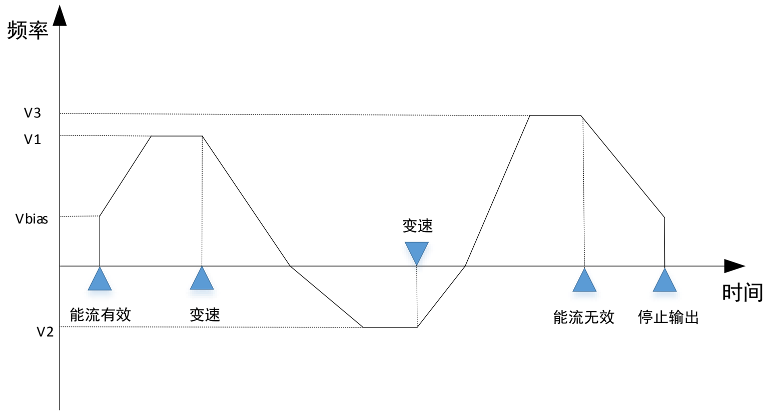


Figure B: plsv2 pulse output change diagram (reverse speed change)

5.2.5 PLSR with acceleration and deceleration pulse output

1.Instruction form

Send out the number of set pulses with the set acceleration and deceleration time and the specified pulse frequency.

PLSR S1 S2 S3 D		With acceleration and deceleration pulse output	Instruction execution	
S1	Output frequency (Hz)	Set pulse output frequency	16Bit instruction (9step) PLSR Continuous execution	32Bit instruction (17step) DPLSR Continuous execution
S2	Number of outputs (PLS)	Set number of pulse output		
S3	Acceleration and deceleration time (MS)	Set acceleration and deceleration time		
D	Output port	High speed pulse output port		

2. Operands

Operands	Bit soft component							Word soft element															
	System · user							System · user					Digit assignment					Indexed address			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: the soft components with gray background in the table indicate the soft components that can be supported.

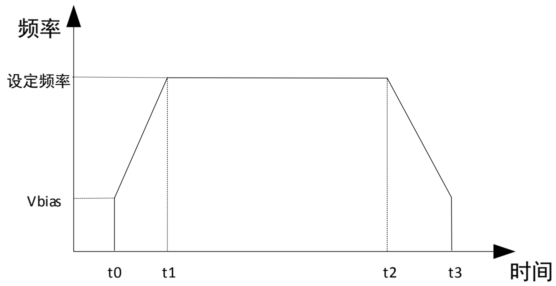
① S1 is the pulse output frequency, in Hz. For 16bit instruction (PLSR), the setting range is 10 ~ 32767; for 32bit instruction (dplsr), the setting range is 10 ~ 3000000; the value of S1 can be modified during instruction execution.

② S2 is the number of pulse output, unit: pls, for 16bit instruction (PLSR), the setting range is - 32768 ~ 32767; for 32bit instruction (dplsr), the setting range is - 2147483648 ~ 2147483647; the value of S2 can be modified in the process of instruction execution.

③ the acceleration and deceleration time set in S3 bit is in the range of 10-5000 (MS), and the default acceleration time is the same as the deceleration time. It can be modified by soft components.

④ D1 is the pulse output port, and y300 / y304 / y310 / y314 / Y0 / Y1 can be specified.

3. Pulse output



① after enabling the energy flow, the pulse output frequency will accelerate from the base speed to the set speed according to the set acceleration time (t1-t0). When the output frequency starts to decelerate, it will decelerate from the set speed to the base speed according to the set deceleration time (t3-t2).

② when the energy flow of the pulse output is off, the pulse output frequency will gradually slow down to the base frequency and then stop the output. Note that in the process of deceleration, when the flag in the pulse output is on, the command will not be driven again. After the output stops, when the energy flow is off - > on, the pulse output processing starts again.

③ for parameters that can be modified during operation, see "5. Change parameters during operation"

3. soft components

The function of the soft elements related to the pulse output is described in detail below.

◆ Pulse output count

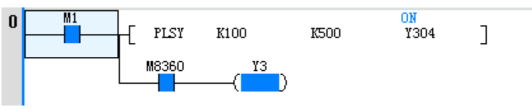
Y300	Y304	Y310	Y314	Y0	Y1	attribute
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output counter records the number of pulses output by the port. This element has the characteristics of D soft element. It can be cleared by relevant command or power off. When the controller state is run - > off, the element will not be cleared.

◆ In pulse output

Y300	Y304	Y310	Y314	Y0	Y1	attribute
M8340	M8360	M8380	M8400	M8420	M8440	In pulse output

As shown in the figure below, when y304 pulse is output, m8360 is set. When the pulse is output, m8360 is reset automatically.



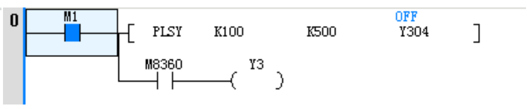


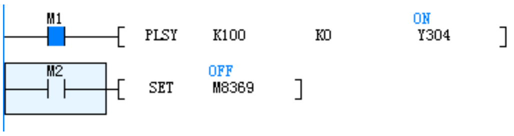
Figure a: in pulse output

Figure B: pulse output complete

◆ Pulse output stop

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stop

When the pulse output stop flag is set, the pulse output stops. Even if the energy flow is valid, the pulse will not output. The effect of the soft element is shown in the figure below. Before the soft element is set, the pulse will output normally. When the soft element is set, even if the energy flow is effective, the pulse will stop outputting.



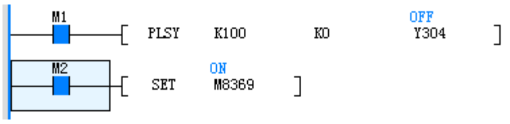


Figure a: before pulse output stop setting Figure B: after the pulse output stops setting

◆ ulse output complete

Y300	attribute	attribute	attribute	attribute	attribute	attribute
M8029						Pulse output complete

When the pulse output is completed, the corresponding m8029 will be set, but the completion of the pulse output does not affect the completion flag of other pulse instructions. As shown in the figure below, after the completion of the first pulse instruction execution, M10 is set, but M11 and M12 are not set.

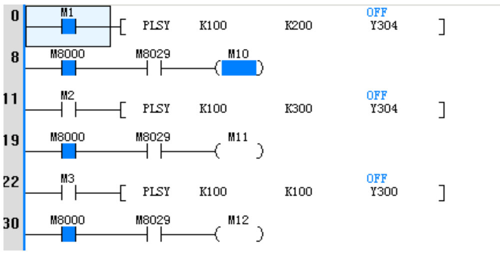


Figure a: m8029 soft element action

◆ Pulse output complete interrupt

If you want to enter the interrupt when the pulse output is completed, different output completion interrupt enabling soft elements can be opened for different Y ports. See the table below. If you want to enter the interrupt after y300 pulse output, you need to set m8352 to 1.

Y300	Y304	Y310	Y314	Y0	Y1	attribute
M8352	M8372	M8392	M8412	M8432	M8452	Output completion interrupt enable

◆ Acceleration and deceleration time is set separately / pulse change is valid

If you want to have different acceleration and deceleration time for each output shaft, or you want to change the number of pulses during operation, you can set the following soft elements:

Y300	Y304	Y310	Y314	Y0	Y1	attribute
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration and deceleration time is set separately / pulse change is valid

Note: this soft element is a common soft element for "acceleration and deceleration are set separately" and "pulse change is effective".

When the above "acceleration and deceleration time is set separately" soft element is off, the following soft elements are used for corresponding shaft parameters:

Y300	attribute	attribute	attribute	attribute	attribute	Attribute
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
Instruction itself setting (S3)						Acceleration and deceleration time (MS)

When the "acceleration and deceleration time is set separately" soft element of a shaft is on, the corresponding shaft parameters use the following soft elements:

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
D8342	D8362	D8382	D8402	D8422	D8442	MAX speed（Hz）
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
Instruction itself setting (S3)						Acceleration time（ms）
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (MS)

4. Changing parameters in operation

◆ In instruction execution, it is allowed to modify the number of pulse outputs

Before the change, special soft components need to be set, as shown in the table below. When the effective soft components of pulse change are effective, the number of pulses can be changed

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration and deceleration time is set separately / pulse / frequency change is valid

The value of changing the number of pulse output can be greater than the current position or less than the current position

Y300	Y304	Y310	Y314	Y0	Y1	Attribute
M8357	M8377	M8397	M8417	M8437	M8457	Effective method of location Modify

Note: position formula required for deceleration: 1 / 2A * (current speed ^ 2 - base speed ^ 2)

① when the number of modified pulse output is less than the current position, it is necessary to modify the setting position to take effect in a special soft element, otherwise it will alarm and operate according to the number of original set pulse output.

② when the number of modified pulse output is greater than the current position, as shown in figure a below, the red curve is the curve after the position is changed to small, and the green curve is the curve after the position is changed to large. Note that the change in size is the change in size relative to the target location.

③ in the follow-up process, the position is still the absolute position relative to the first entry after multiple Modifys.

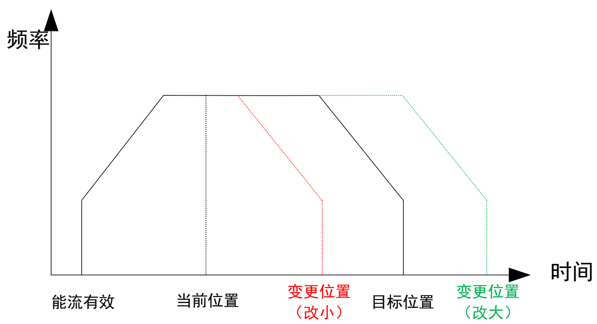


Figure a: the number of modified pulse outputs is greater than the current position

① When the number of modified pulse output is less than the position required for deceleration, there are two situations:

M8357 is off: the current modified position will not take effect, and the error "6713, the positioning position is too small to slow down".

M8357 is on: as shown in Figure B below, when the modified position is smaller than the current position, the modified pulse output frequency will gradually decelerate to 0 and then reverse to the modified position.

Note: the slope of pulse output speed in deceleration section is calculated according to deceleration time, and the slope of acceleration in reverse acceleration section is calculated according to acceleration time.

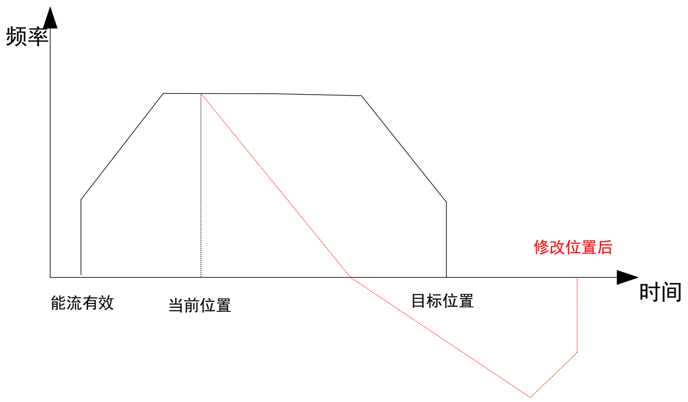


Figure B: the number of modified pulse outputs is less than the current position

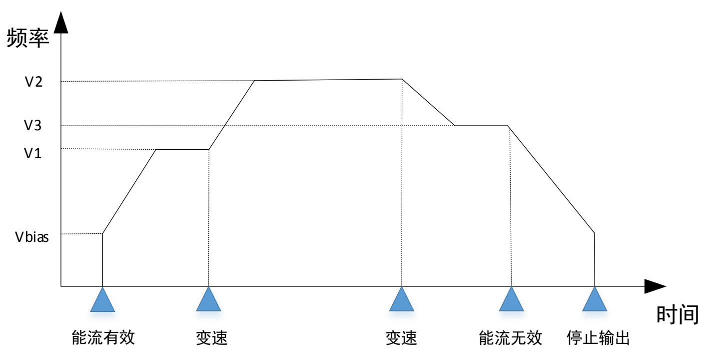
◆ In instruction execution, it is allowed to modify the pulse output frequency.

① during the output of instruction, it is allowed to modify the frequency of pulse output. The changed pulse output frequency can be greater than or less than the current pulse output frequency, with the range of [103000000].

② when the frequency is more than 3000000, it will be limited to 3000000, when the frequency is less than 10, it will report error 6706, and the data is unreasonable.

③ in the process of speed change, the acceleration and deceleration slope remain unchanged.

④ if the number and frequency of pulses are modified at the same time, the effective frequency of the next scanning cycle will be determined according to the position.



◆ S curve is not supported.

◆ The acceleration Modify is not supported.

5.2.6 DRVA Absolute positional positioning

1. Command form

The set number of pulses is sent at the set output port and the specified pulse frequency and direction, and it moves basing on the relative position.

DRVA S1 S2 D1 D2		Absolute positional positioning	Command execution	
S1	Pulse number	Pulse output number setting (PLS)	16-bit command (9step) DRVA continuous execution	32-bit command (17step) DDRVA continuous execution
S2	Pulse frequency	Pulse output frequency setting(Hz)		
D1	Output port	High speed pulse output port		
D2	Output direction	Pulse running direction port or bit variable		

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that can be supported.

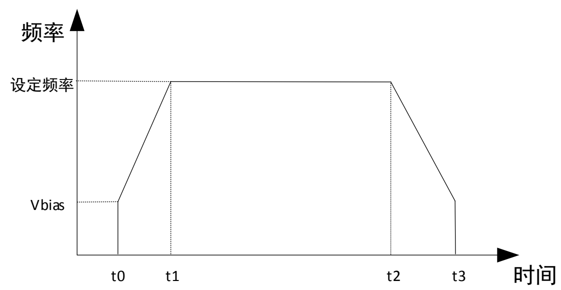
① S1 is the number of pulse outputs, the unit is PLS. For 16-bit commands (DRVA), the setting range is -32768~32767. For 32-bit commands (DDRVA), the setting range is -2,147,483,648~2,147,483,647; in the process of command execution, the value of S1 can be modified.

② S2 is the pulse output frequency, the unit is Hz. For 16-bit command (DRVA), the setting range is 10~32767Hz; for 32-bit command (DDRVA), the setting range is 10~3,000,000Hz; in the process of command execution, the value of S2 can be modified.

③ D1 is the pulse output port, and Y300/Y304/Y310/Y314/Y0/Y1 can be specified.

④ D2 is the running direction output port or bit variable. The state of this bit is determined by the controller's own pulse output state. When the pulse output is in forward running, the state of D2 is ON. When the pulse output is in reverse running, the state of D2 is OFF.

3. Pulse output



①After the energy flow is enabled, the pulse output frequency will accelerate from the base speed to the set speed according to the set acceleration time (t1-t0). When the output frequency starts to decelerate, it will follow the set deceleration time (t3-t2) and decelerates from set speed to base speed.
②When the energy flow of the pulse output is OFF, the pulse output frequency will gradually decelerate to the base frequency and then stop output. Note that during the deceleration, when the pulse output flag
③is ON, the re-drive command will not be accepted. After the output is stopped, the pulse output processing is restarted when the energy flow is OFF->ON.
The parameters can be modified during operation. Please refer to “5. Changing parameters during operation”.

4. Soft component

The function of the soft component related to the pulse output will be described in detail below.

◆ Pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count value (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output count value records the pulse number outputed by the port. This component has the characteristics of the D soft component and can be cleared by related commands or cleared by power-off. The component will not be cleared when the controller status is RUN->OFF.

◆ During pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8340	M8360	M8380	M8400	M8420	M8440	During pulse output

As shown in the figure below, when the Y304 pulse is output, M8360 is set. When the pulse output is completed, the M8360 is automatically reset.

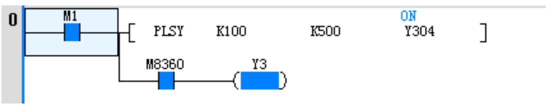


Figure a: During pulse output

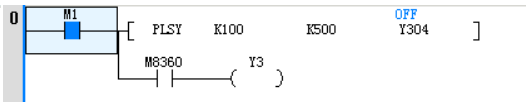


Figure b: Pulse output completed

◆ Pulse output stops

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stops

When the pulse output stop flag is set, the pulse output stops, and even if the energy flow is valid, the pulse will not output. The effect of the soft component is shown as the figure below. Before the soft component is set, the pulse is output normally. When the soft component is set, the pulse will stop output even if the energy flow is valid.

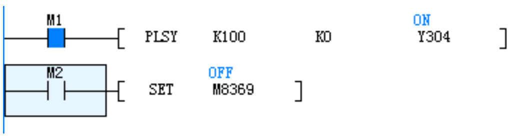


Figure a: Before the pulse-output-stop is set

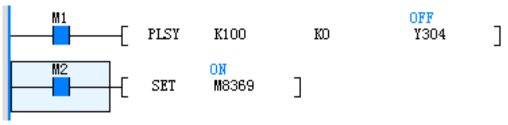


Figure b: After the pulse-output-stop is set

◆ Pulse output completed

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8029						Pulse output completed

When this pulse output is completed, the corresponding M8029 will be set, but the completion of this pulse output does not affect the pulse output completion flag of other pulse commands. As shown in the figure below, after the execution completetion of the first pulse command, M10 is set, but neither M11 nor M12 are set.

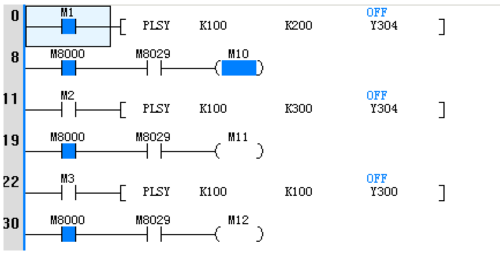


Figure a: M8029 soft components action

◆ Pulse output completion interrupt

If you want to enter the interrupt when the pulse output is completed, you can turn on the different output completion interrupt enable soft components for different Y ports. As shown in the table below, if you want to enter the interrupt after the Y300 pulse output is completed, you need to set M8352 to 1.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8352	M8372	M8392	M8412	M8432	M8452	Output completion interrupt enable

◆ Acceleration/deceleration time separate setting/pulse change is valid

If you want each output axis to have different acceleration/deceleration time, or you want to change the number of pulses during operation, you can set the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

Note: This soft component is a soft component shared by "Acceleration/Deceleration Separate Setting" and "Pulse Change Valid".

When the above "Acceleration/deceleration time alone setting" soft component is OFF, the following parameters are used for the corresponding axis parameters:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (ms)

When the "Acceleration/deceleration time alone setting" soft component of an axis is ON, the corresponding axis parameters use the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms)

5. Change parameters during operation

◆ he number of pulse outputs allowed to be modified during command execution

Before changing, you need to set special soft components as the table below; when the pulse change valid soft component is valid, the number of pulses can be changed.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

The value of pulse output Modify number can be more than the current position or less than the current position.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8357	M8377	M8397	M8417	M8437	M8457	Position Modify valid method

Note: The position formula required for deceleration: 1/2a* (current speed ^2 - base speed ^2)

① When the number of modified pulse output is less than the current position, the set position must be modified to be effective mode special soft component, otherwise it will alarm and run according to the originally set pulse output number.

② When the number of modified pulse outputs is more than the current position, as shown in the following figure a, the red curve is the curve after the position is changed smaller, and the green curve is the curve after the position is changed bigger. Note that the changes here (bigger or smaller) are refered to the target position.

③ In the subsequent process, even if it is modified several times, the position is still the absolute position relative to the first entry.

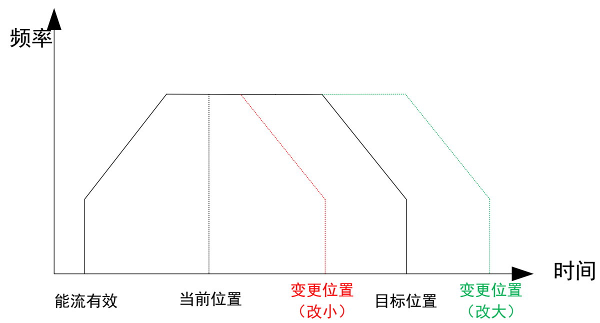


Figure a: The number of modified pulse outputs is more than the current position

④ When the number of modified pulse outputs is less than the position required for deceleration, there are two cases:

M8357 is OFF: The currently modified position will not take effect, and the error is “6713. The positioning position is too small to decelerate”.

M8357 is ON: As shown in the following figure b, when the modified position is smaller than the current position, the modified pulse output frequency will gradually decelerate to 0 and then reverse to the modified position.

Note: The slope of the pulse output speed in the deceleration section is calculated according to the deceleration time, and the acceleration slope of the reverse acceleration section is calculated according to the acceleration time.

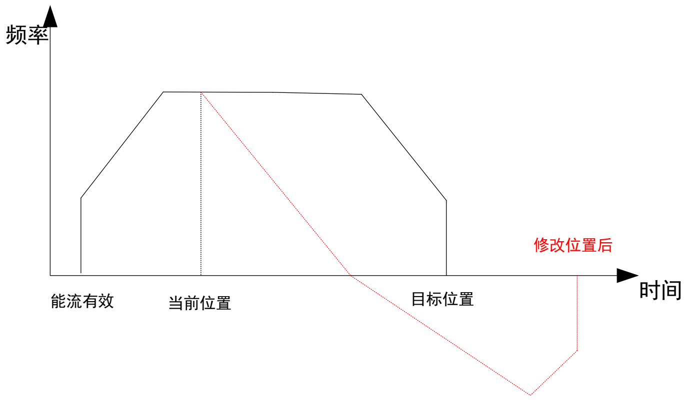


Figure b: The number of modified pulse outputs is less than the current position

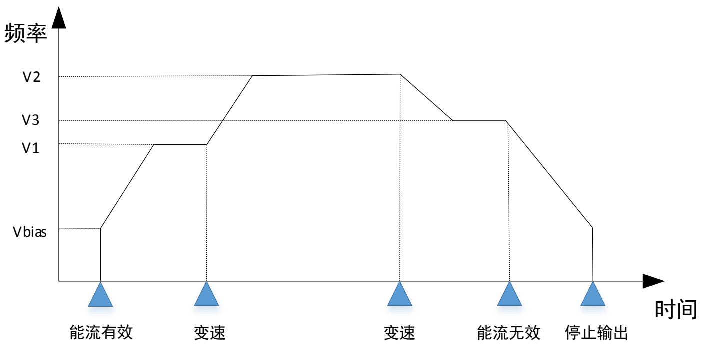
◆ During the execution of the command, the pulse output frequency is allowed to be modified.

① During the output of the command, the pulse output frequency can be modified. The changed pulse output frequency can be more than or less than the current pulse output frequency, and the range is [10, 3000000].

② When the frequency is more than 3000000, it will be limited to 3000000. When the frequency is less than 10, it will report 6706, and the data is unreasonable.

③ During the speed change, the acceleration and deceleration slopes remain unchanged.

④ If the number of pulses and the frequency are modified at the same time, the position wil be valid at first and frequency will be valid for the next scan cycle.



◆ The acceleration is allowed to be modified during command execution.

(Refer to supplementary documentation)

◆ Support S curve function

The distinction is made by setting the special soft component "S-curve acceleration/deceleration enable flag". If the flag is not set, the default is trapezoidal acceleration and deceleration. S-curve acceleration and deceleration is shown in the following table:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8347	M8367	M8387	M8407	M8417	M8437	S curve acceleration and deceleration enable flag
D8351	D8371	D8391	D8411	D8431	D8451	S curve time

DRVA, DRVI supports S-curve acceleration and deceleration, and it can increase the target speed under the same mechanical stability conditions, so it can shorten the positioning time and improve the processing efficiency.

S curve time range: [2~499].--- Need to expand the range, please contact the manufacturer.

Actual acceleration/deceleration time = T-type acceleration/deceleration + S-curve time.

5.2.7 DRVI Relative Positioning

1. Command form

The set number of pulses is sent at the set output port and the specified pulse frequency and direction, and it moves basing on the relative position.

DRVI S1 S2 D1 D2			Relative positional positioning	Command execution	
S1	Pulse number	Pulse outputs number setting (PLS)		16-bit command (9step) DRVA continuous execution	32-bit command (17step) DDRVA continuous execution
S2	Pulse frequency	Pulse output frequency setting (Hz)			
D1	Output port	High speed pulse output port			
D2	Output direction	Pulse running direction port or bit variable			

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that can be supported.

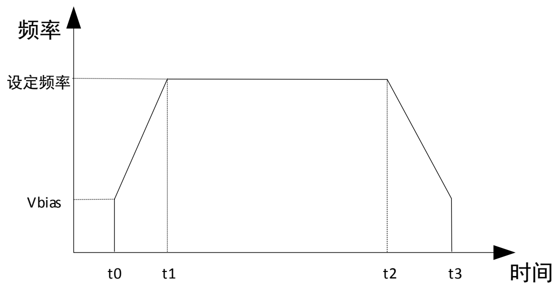
① S1 is the number of pulse outputs, the unit is PLS. For 16-bit commands (DRVI), the setting range is -32768~32767. For 32-bit commands (DDRVI), the setting range is -2,147,483,648~2,147,483,647; In the process of command execution, the value of S1 can be modified.

② S2 is the pulse output frequency, the unit is Hz. For 16-bit command (DRVI), the setting range is 10~32767Hz; for 32-bit command (DDRVI), the setting range is 10~3,000,000Hz; In the process of command execution, the value of S2 can be modified.

③ D1 is the pulse output port, and Y300/Y304/Y310/Y314/Y0/Y1 can be specified.

④ D2 is the running direction output port or bit variable. The state of this bit is determined by the controller's own pulse output state. When the pulse output is in forward running, the state of D2 is ON. When the pulse output is in reverse running, the state of D2 is OFF.

3. Pulse output



① When the energy flow is enabled, the pulse output frequency will accelerate from the base speed to the set speed according to the set acceleration time (t1-t0). When the output frequency starts to decelerate, it will decelerate from the set speed to the base speed according to the set deceleration time (t3-t2).

② When the energy flow of the pulse output is OFF, the pulse output frequency will gradually decelerate to the base frequency and stop output. Note that during the deceleration, when the pulse output flag is ON, the re-drive command will not be accepted. After the output is stopped, the pulse output processing is restarted when the power flow is OFF->ON.

③ Parameters can be modified during operation. For details, please refer to “5. Changing parameters during operation”.

4. Soft components

The soft components functions related to the pulse output will be described in detail below.

◆ Pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count value (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output count value records the pulse number outputed by the port. This component has the characteristics of the D soft component and can be cleared by related commands or cleared by power-off. The component will not be cleared when the controller status is RUN->OFF.

◆ During pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8340	M8360	M8380	M8400	M8420	M8440	During pulse output

As shown in the figure below, when the Y304 pulse is output, M8360 is set. When the pulse output is completed, the M8360 is automatically reset.

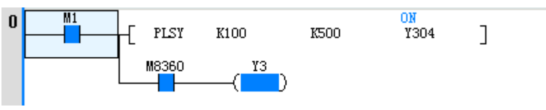


Figure a: During pulse output

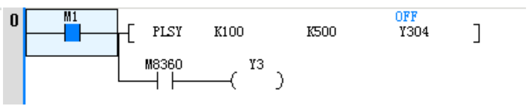


Figure b: Pulse output completed

◆ Pulse output stops

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stops

When the pulse output stop flag is set, the pulse output stops, and even if the energy flow is valid, the pulse will not output. The effect of the soft component is shown as the figure below. Before the soft component is set, the pulse is output normally. When the soft component is set, the pulse will stop output even if the energy flow is valid.

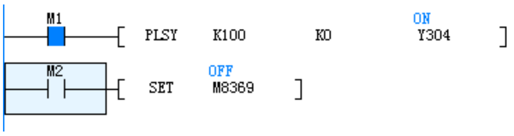


Figure a: Before the pulse-output-stop is set

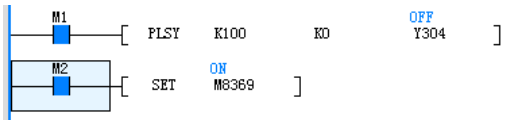


Figure b: After the pulse-output-stop is set

◆ Pulse output completed

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8029						Pulse output completed

When this pulse output is completed, the corresponding M8029 will be set, but the completion of this pulse output does not affect the pulse output completion flag of other pulse commands. As shown in the figure below, after the execution of the first pulse command is completed, M10 is set, but neither M11 nor M12 are set.

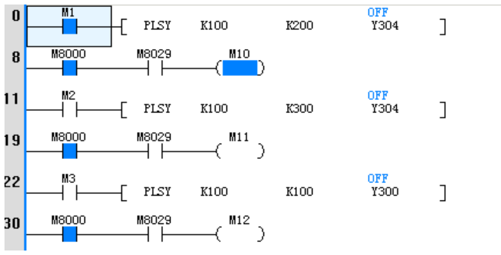


Figure a: M8029 soft component action

◆ Pulse output completion interrupt

If you want to enter the interrupt when the pulse output is completed, you can turn on the different output completion interrupt enable soft components for different Y ports. As shown in the table below, if you want to enter the interrupt after the Y300 pulse output is completed, you need to set M8352 to 1.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8352	M8372	M8392	M8412	M8432	M8452	Output completion interrupt enable

◆ Acceleration/deceleration time separate setting/pulse change is valid

If you want each output axis to have different acceleration/deceleration time, or you want to change the number of pulses during operation, you can set the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

Note: This soft component is a soft component shared by “Acceleration/Deceleration Separate Setting” and “Pulse Change Active”.

When the above “Acceleration/deceleration time alone setting” soft component is OFF, the following parameters are used for the corresponding axis parameters:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (ms)

When the “Acceleration/deceleration time alone setting” soft component of an axis is ON, the corresponding axis parameters use the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms)

5. Change parameters during operation

◆ The number of pulse outputs allowed to be modified during command execution

Before changing, you need to set special soft components as the table below; when the pulse change valid soft component is valid, the number of pulses can be changed.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

The value of pulse output Modify number can be more than the current position or less than the current position.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8357	M8377	M8397	M8417	M8437	M8457	Position Modify method

Note: The position formula required for deceleration: 1/2a* (current speed ^2 - base speed ^2)

① When the number of modified pulse output is less than the current position, the set position must be modified to be effective mode special soft component, otherwise it will alarm and run according to the originally set pulse output number.

② When the number of modified pulse outputs is more than the current position, as shown in the following figure a, the red curve is the curve after the position is changed smaller, and the green curve is the curve after the position is changed bigger. Note that the changes here (bigger or smaller) are refered to the target position.

③ In the subsequent process, even if it is modified several times, the position is still the absolute position relative to the first entry.

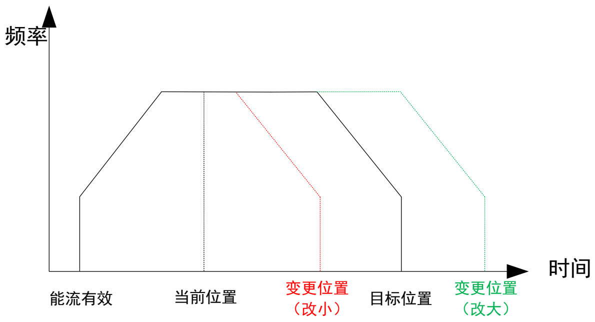


Figure a: The number of modified pulse outputs is more than the current position

④ When the number of modified pulse outputs is less than the position required for deceleration, there are two cases:

8357 IS OFF: The currently modified position will not take effect, and the error is “6713. the positioning position is too small to decelerate”.

M8357 is ON: As shown in the following figure b, when the modified position is smaller than the current position, the modified pulse output frequency will gradually decelerate to 0 and then reverse to the modified position.

Note: The slope of the pulse output speed in the deceleration section is calculated according to the deceleration time, and the acceleration slope of the reverse acceleration section is calculated according to the acceleration time.

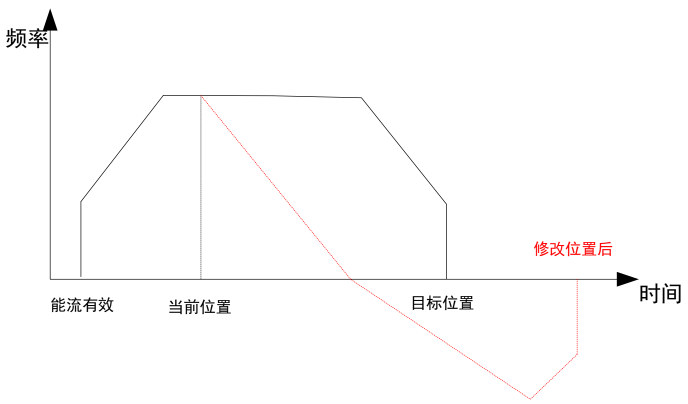


Figure b: The number of modified pulse outputs is less than the current position

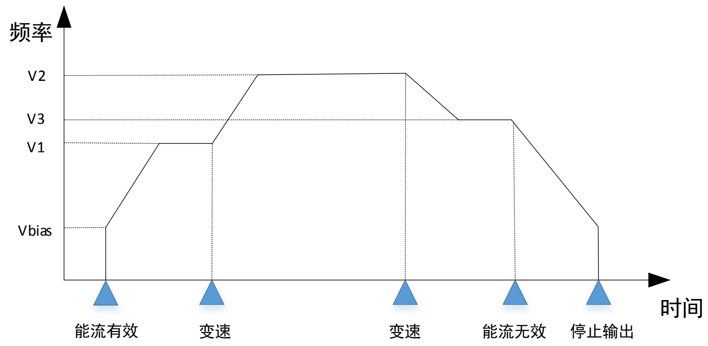
◆ During the execution of the command, the pulse output frequency is allowed to be modified.

① During the command outpu, the frequency of the pulse output can be modified. The changed pulse output frequency can be more than or less than the current pulse output frequency, and the range is [10, 3000000].

② When the frequency is more than 3000000, it will be limited to 3000000. When the frequency is less than 10, it will report 6706, and the data is unreasonable.

③ During the speed change, the acceleration and deceleration slopes remain unchanged.

④ If the number of pulses and the frequency are modified at the same time, the position wil be valid at first and frequency will be valid for the next scan cycle.



- The acceleration is allowed to be modified during command execution.
  (Refer to supplementary documentation)

◆ Support S curve function

The distinction is made by setting the special soft component “S-curve acceleration/deceleration enable flag”. If the flag is not set, the default is trapezoidal acceleration and deceleration. S-curve acceleration and deceleration is shown in the following table:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8347	M8367	M8387	M8407	M8417	M8437	S curve acceleration and deceleration enable flag
D8351	D8371	D8391	D8411	D8431	D8451	S curve time

DRVA, DRVI supports S-curve acceleration and deceleration, so it can increase the target speed under the same mechanical stability conditions, to shorten the positioning time and improve the processing efficiency.

S curve time range: [2~499]. --- Need to expand the range, please contact the manufacturer.

Actual acceleration/deceleration time = T-type acceleration/deceleration + S-curve time.

5.2.8 ZRN Origin Return

1. Command form

After starting, it accelerates to the set return output frequency, and the actuator moves to the origin (DOG). After detecting the DOG signal, it decelerates to the creeping speed. When the DOG signal is OFF, the output stops.

<table>
<tr><td colspan="2">ZRN S1 S2 S3 D</td><td>Origin return</td><td colspan="2">Command execution</td></tr>
<tr><td>S1</td><td>Returm frequency</td><td>Set origin return frequency (Hz)</td><td rowspan="4">16-bit command (9step) ZRN continuous execution</td><td rowspan="4">32-bit command (17step) DZRN continuous execution</td></tr>
<tr><td>S2</td><td>Creeping frequency</td><td>Set creeping frequency (Hz)</td></tr>
<tr><td>S3</td><td>DOG signal</td><td>Specified origin input signal (DOG)</td></tr>
<tr><td>D</td><td>Output port</td><td>High speed pulse output port</td></tr>
</table>

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that can be supported.

① This command is output from the specified port at the specified pulse speed when the controller works with the servo driver, so that the actuator moves to the origin (DOG) until the origin signal meets the condition position.

② S1 is the origin return action frequency, and the unit is Hz. For 16-bit commands (ZRN), the setting range is 10~32767Hz; for 32-bit commands (DZRN), the setting range is 10~3,000,000Hz.

③ S2 is the creeping speed after the origin signal turns ON, and the setting range is 10~32767Hz.

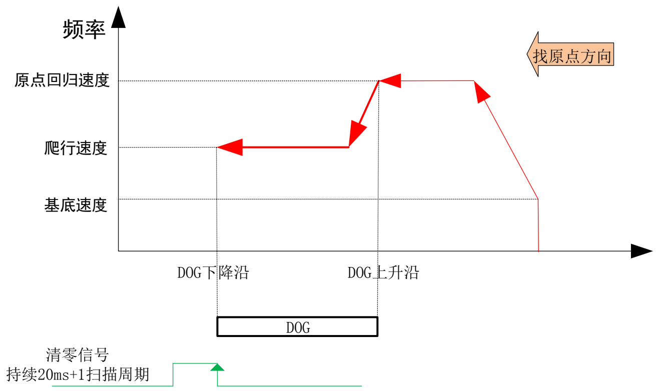
④ S3 is the origin signal (DOG) input. Although the X/Y/M/S signals are all OK, only the X signal has the best timeliness.

⑤ D is the pulse output port, and Y300/Y304/Y310/Y314/Y0/Y1 can be specified.

3. Pulse output

① During power-on and initial operation, the origin position return command ZRN is generally executed to write the origin position data of the mechanical action in advance. If the position information has the power-down save function, the command does not need to be run every time the power is on; In the process, only the negative direction can be moved, so the origin return action must be performed at the front end of the DOG signal and return to the origin in the negative direction.

② Pulse output as the figure below, the pulse frequency gradually decelerates from the base speed to the origin return speed, and drives the servo to move in the negative direction. When the rising edge of the DOG signal is encountered, it starts to decelerate to creeping speed and continues to move in the negative direction. When the falling edge of the DOG signal is encountered, it is considered that the origin is found and the pulse output is stopped immediately.



③ When the command energy flow is OFF, it will stop immediately. When the power flow is OFF->ON, the pulse output restarts.

④ When the command is executed, the M8029 flag is turned ON.

4. Soft components

The soft components functions related to the pulse output will be described in detail below.

◆ Pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count value (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output count value records the pulse number outputted by the port. This component has the characteristics of the D soft component and can be cleared by related commands or cleared by power-off. The component will not be cleared when the controller status is RUN->OFF.

◆ During pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8340	M8360	M8380	M8400	M8420	M8440	During pulse output

◆ Pulse output stops

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stops

When the pulse output stop flag is set, the pulse output stops, and even if the current flow is valid, the pulse will not be output.

◆ No pulse output completion and pulse output completion interrupt

◆ Acceleration/deceleration time separate setting/pulse change is valid

If you want each output axis to have different acceleration/deceleration time, or you want to change the number of pulses during operation, you can set the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

Note: This soft component is a soft component shared by "Acceleration/Deceleration Separate Setting" and "Pulse Change Valid".

When the above "Acceleration/deceleration time alone setting" soft component is OFF, the following parameters are used for the corresponding axis parameters:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (ms)

When the "Acceleration/deceleration time separate setting" soft component of an axis is ON, the corresponding axis parameters use the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms)

◆ Clear signal output

The pulse clear signal can be output by setting the special soft component "clear signal output valid flag". This signal can be connected to the servo pulse clear port to clear the servo pulse deviation, so that the servo can stop at the falling edge of DOG accurately. Clear singnal output as shown in the following table:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8341	M8361	M8381	M8401	M8421	M8441	DSZR/ZRN Clear signal output valid flag

The clear signal can be specified by a special register and can only be a Y output port. For example, D8350 specifies Y300 as the clear output port. When D8350 is specified as 5, the pulse clear output port is Y5, as shown in the following table:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8350	D8370	D8390	D8410	D8430	D8450	Clear soft component No. (decimal)
5 (default)	6 (default)	7 (default)	8 (default)	9 (default)	10 (default)	Specify output port defaults

Note: The value of the clear soft component here is a decimal value, and the Y port is defined in octal, so you need to convert it when setting. For example, if D8430 is set to 9, then 9 in decimal is equal to 11 in octal, so it corresponds to port Y11.

◆ Clear logical position and encoder position

After the zero return is completed, the system will wait for a delay time of SD65, automatically clearing the current logical position (D8340, D8341) and the encoder feedback position (SD56, SD57).

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
SD65	SD165	SD265	SD765	SD865	SD965	Zero return delay time (default 20ms)
SD56	SD156	SD256	SD756	SD856	SD956	Encoder feedback position (PLS)
SD57	SD157	SD257	SD757	SD857	SD957	

Note: SD65 setting is too small, due to servo hysteresis, may cause residual retention of SD56.

5.2.9 DSZRDOG Search Origin Return (Under Development)

1. Command form

After starting, it accelerates to the set return output frequency, and the actuator moves to the origin (DOG). After detecting the DOG signal, it decelerates to the creeping speed. When detecting that the zero signal is OFF to ON and stops outputting.

DSZR S1 S2 D1 D2			DOG search origin return	Command execution	
S1	DOG signal	Specified origin input signal (DOG)		16-bit command (9step) DSZR continuous execution	32-bit command (17step) DDSZR continuous execution
S2	Zero signal	Specified zero input signal			
D1	Output port	High speed pulse output port			
D2	Output direction	Pulse running direction port or bit variable			

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

注：Note: The gray soft components in the table indicate the soft components that can be supported.

① When the controller works with the servo driver, the pulse is output from the specified port at the pulse speed specified by the special register, so that the actuator moves according to the preset motion origin. After the near-point signal (DOG) is turned ON to OFF during the operation, when the zero signal is detected from OFF to ON, the pulse output is stopped immediately.

② The rotation direction signal is output during the return process, and it outputs clear signal after return completion.

③ In the system with the forward/reverse limit setting, it can return to the origin by enabling the DOG search mode; in the system where the forward/reverse limit is not set or the forward/reverse limit is not used for the origin return, you can specify the origin return direction to perform the origin return. Among them:

S1 is a near-point signal (DOG) input. Although the X/Y/M/S signals are all OK, only the X signal has the best timeliness; S2 is the zero signal input and indicates the exact position of the motion origin. Only the X signal can be specified;

D1 is the pulse output port. Y300/Y304/Y310/Y314/Y0/Y1 can be specified;

D2 is the rotation direction output port. ON: Forward (pulse output increases the current value); OFF: Reverse (pulse output reduces the current value).

3. Soft components

The soft components functions related to the pulse output will be described in detail below.

◆ Special D component definition:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count value (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz) [default 3000000]
D8343	D8363	D8383	D8403	D8423	D8443	
D8344	D8364	D8384	D8404	D8424	D8444	Origin return speed (HZ) [default 50000]
D8345	D8365	D8385	D8405	D8425	D8445	
D8346	D8366	D8386	D8406	D8426	D8446	Creeping speed (HZ) [default 2000]
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz) [default 500]
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms) [default 100]
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms) [default 100]
D8350	D8370	D8390	D8410	D8430	D8450	Clear soft components No.

◆ Special M component definition:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8343	M8363	M8383	M8403	M8423	M8443	Forward limit
M8344	M8364	M8384	M8404	M8424	M8444	Reversal limit
M8345	M8365	M8385	M8405	M8425	M8445	Near-point signal logic reverse [1]
M8346	M8366	M8386	M8406	M8426	M8446	Zero signal logic reverse [1]

M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stop flag [1]
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate set and pulse change is valid
M8351	M8371	M8391	M8411	M8431	M8451	Port output initialization flag
M8354	M8374	M8394	M8414	M8434	M8454	DSZR command execution abnormal end flag

[1]: RUN→STOP, Clear.

Maximum speed, origin return speed, creeping speed, base speed, please follow:

Base speed ≤ origin return speed ≤ maximum speed;

Base speed ≤ creeping speed ≤ maximum speed;

Parameter setting range:

Origin return speed, ranging from 10~3100,000Hz;

Creeping speed, ranging from 10~32,767Hz;

Base speed, ranging from 10 to 32,767 Hz;

Specify the origin return direction: Specify the return origin direction according to the ON/OFF of the origin return direction designation flag. The base speed is accelerated to the origin return speed, and is moved in the direction specified by the origin return direction designation flag. It senses that the near-point signal (DOG) specified by S1 is ON and starts to decelerate to the creeping speed. When the near-point signal (DOG) specified by S1 is turned from ON to OFF, if the zero-point signal specified by S2 is detected from OFF to ON, the pulse output is immediately stopped.

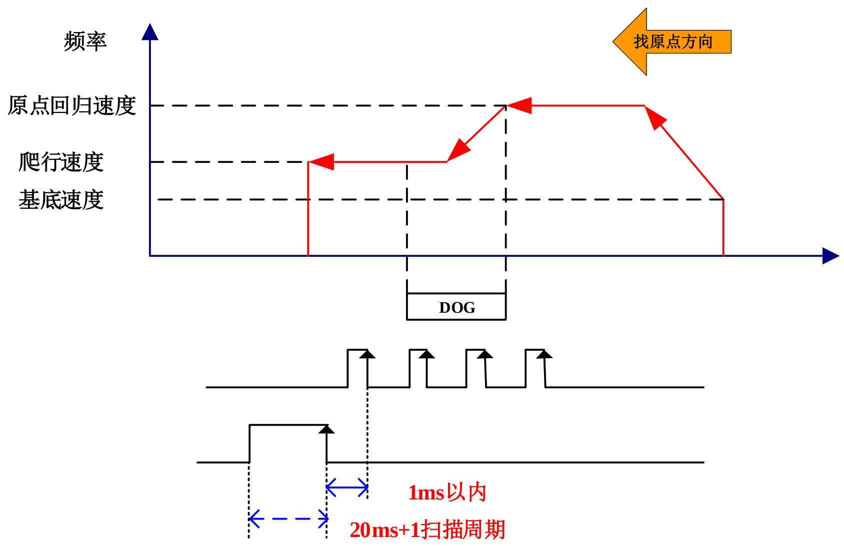
If the near-point signal and the zero-point signal specify the same input, the zero-point signal is not used as in the ZRN command, and the pulse output is stopped immediately after the near-point signal is turned from ON to OFF; if the near-point signal and the zero-point signal specify the same input, then if the logic inversion flag is ON, the logic is subject to that of the near-point signal.

When the clear signal output function is enabled (ON), after the pulse output is stopped (within 1 ms), the clear signal remains ON for [20ms+1 calculation cycle]. The command execution end flag (M8029) is turned ON, and the origin return operation is ended.

This is the description of the near-point signal logic inversion flag bit and the zero-point signal logic inversion flag bit being OFF. If the logic flag is ON, the corresponding near-point and zero-point signals is turned ON to OFF, and OFF is turned to ON.

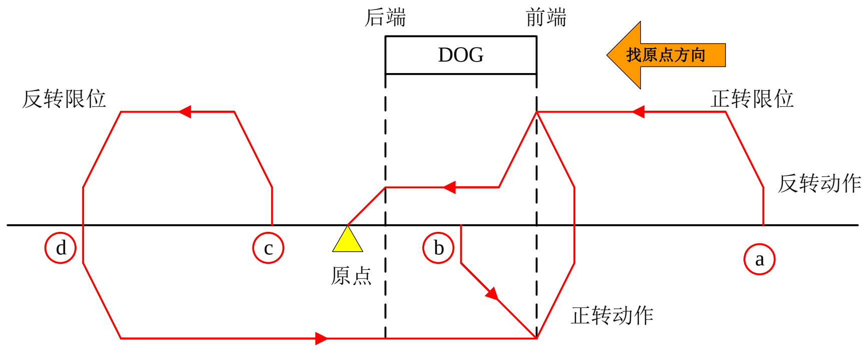
When the command energy flow is OFF, it will stop immediately; when the energy flow is OFF→ON, the pulse output processing restarts; when the command is executed, the M8029 flag is turned ON;

The pulse output diagram is as follows:



◆ DOG search function

When the design has a forward limit and a reverse limit, the origin return using the DOG search function is executed. At this time, due to the different starting position of the origin return, the original returning action is also different.



a) When the starting position is before the DOG, including the forward rotation limit 1 is ON:

① Start the origin return operation by executing the origin return command.

② Start moving toward the origin return direction at the origin return speed.

③ Once the front end of the DOG is detected, it begins to decelerate to the creeping speed.

④ When the back end of the DOG is detected, it stops when the first zero signal is detected.

b) When the starting position is within the DOG:

① Start the origin return operation by executing the origin return command.

② Start moving toward the opposite direction of origin return at the origin return speed.

③ Deceleration stops after detecting the front end of the DOG. (Leaving DOG)

④ art moving toward the origin return direction at the origin return speed. (Enter DOG again)

⑤ Once the front end of the DOG is detected, it begins to decelerate to the creeping speed.

⑥ When the back end of the DOG is detected, it stops when the first zero signal is detected.

c) When the near position signal is OFF (after passing DOG):

① Start the origin return operation by executing the origin return command.

② Start moving toward the origin return direction at the origin return speed.

③ Deceleration stop when reverse limit 1 (reverse limit) is detected.

④ Start moving toward the opposite direction of origin return at the origin return speed.

⑤ Deceleration stops after detecting the front end of the DOG. (check out (leave) DOG)

⑥ Start moving toward the origin return direction at the origin return speed. (Enter DOG again)

⑦ Once the front end of the DOG is detected, it begins to decelerate to the creeping speed.

⑧ When the back end of the DOG is detected, it stops when the first zero signal is detected.

d) When the limit switch in the origin return direction (reverse limit 1) is ON:

① Start the origin return operation by executing the origin return command.

② Start moving toward the opposite direction of origin return at the origin return speed.

③ Deceleration stops after detecting the front end of the DOG. (Leaving DOG)

④ Start moving toward the origin return direction at the origin return speed. (Enter DOG again).

⑤ Once the front end of the DOG is detected, it begins to decelerate to the creeping speed.

⑥ When the back end of the DOG is detected, it stops when the first zero signal is detected.

Note: When designing the near-point signal (DOG), please consider having enough ON time to decelerate to the creeping speed; please make the creeping speed as slow as possible because it stops without deceleration, and if the creeping speed is too fast, it may cause positional shift.

◆ During pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8340	M8360	M8380	M8400	M8420	M8440	During pulse output

◆ Pulse output stops

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stops

When the pulse output stop flag is set, the pulse output stops, and even if the current flow is valid, the pulse will not be output.

◆ No pulse output completion and pulse output completion interrupt

◆ Acceleration/deceleration time separate setting/pulse change is valid

If you want each output axis to have different acceleration/deceleration time, or you want to change the number of pulses during operation, you can set the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

Note: This soft component is a soft component shared by “Acceleration/Deceleration Separate

Setting" and "Pulse Change Valid".

When the above "Acceleration/deceleration time alone setting" soft component is OFF, the following parameters are used for the corresponding axis parameters:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (ms)

When the "Acceleration/deceleration time alone setting" soft component of an axis is ON, the corresponding axis parameters use the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms)

◆ Clear signal output

The pulse clear signal can be output by setting the special soft component "clear signal output valid flag". This signal can be connected to the servo pulse clear port to clear the servo pulse deviation, so that the servo can stop at the falling edge of DOG accurately. Clear singnal output as shown in the following table:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8341	M8361	M8381	M8401	M8421	M8441	DSZR/ZRN Clear signal output valid flag

The clear signal can be specified by a special register and can only be a Y output port. For example, D8350 specifies Y300 as the clear output port. When D8350 is specified as 5, the pulse clear output port is Y5, as shown in the following table:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8350	D8370	D8390	D8410	D8430	D8450	Clear soft component No. (decimal)
5 (default)	6 (default)	7 (default)	8 (default)	9 (default)	10 (default)	Specify output port defaults

Note: The value of the clear soft component here is a decimal value, and the Y port is defined in octal, so you need to convert it when setting. For example, if D8430 is set to 9, then 9 in decimal is equal to 11 in octal, so it corresponds to port Y11.

◆ Signal logic reverse

OFF: Positive logic (when the input is ON, the near/zero signal is ON);

ON: Negative logic (when the input is OFF, the near/near signal is ON).

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8345	M8365	M8385	M8405	M8425	M8445	Near-point signal logic reverse
M8346	M8366	M8386	M8406	M8426	M8446	Zero signal logic reverse

5.2.10 DVIT Interrupt Location (Not Developed)

1. Command form

After starting, it accelerates to the set speed output frequency. When the interrupt input signal is detected, it immediately accelerates or decelerates to the position segment output frequency, and outputs the set number of pulses.

DVIT S1 S2 S3 D1 D2 S4		Interrupt positioning	Command execution	
S1	Number of pulses	Set the number of position pulse output after interrupt	16-bit command (13step) DSZR continuous execution	32-bit command (25step) DDSZR continuous execution
S2	Output frequency 1	Set speed segment pulse output frequency		
S3	Output frequency 2	Set position segment pulse output frequency after interrupt		
D1	Output port	High speed pulse output port		
D2	Output direction	Pulse running direction port or bit variable		
S4	Interrupt input	Interrupt input signal port (range X0-X4)		

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S4	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

注：Note: The gray soft components in the table indicate the soft components that can be supported.

① When the controller works with the servo driver, it outputs the pulse from the specified port at the specified pulse speed. After the interrupt signal is detected during the operation, the given number pulse is output according to the position segment pulse output frequency after the interrupt. The mechanism makes an offset movement based on the current position.

② S1 is the number of pulse output, the unit is PLS. For 16-bit command (DVIT), the setting range is -32768~32767; for 32-bit command (DDVIT), the setting range is -2,147,483,648~2,147,483,647;

③ S2 is the pulse output frequency of the speed segment before interruption, the unit is Hz. For the 16-bit command (DVIT), the setting range is 10~32767Hz; for the 32-bit command (DDVIT), the setting range is 10~3,000,000Hz; The value of S2 can be modified during the execution.

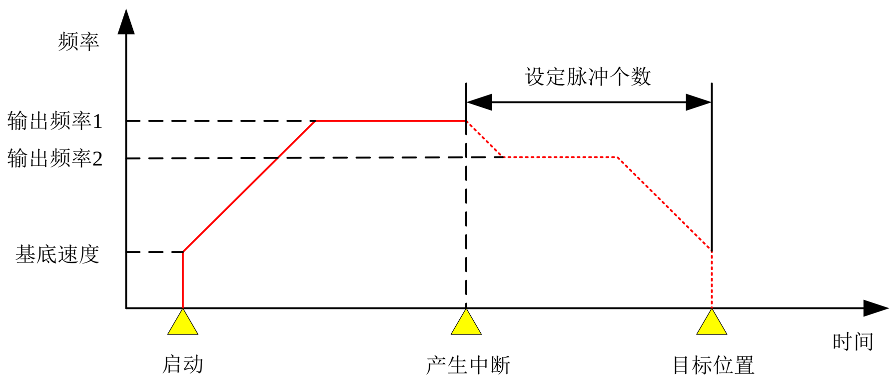
④ S3 is the pulse output frequency of the position segment after interruption, the unit is Hz. For 16-bit command (DVIT), the setting range is 10~32767Hz; for 32-bit command (DDVIT), the setting range is 10~3,000,000Hz; The value of S2 can be modified during the execution.

⑤ D1 is the pulse output port, and Y300/Y304/Y310/Y314/Y0/Y1 can be specified.

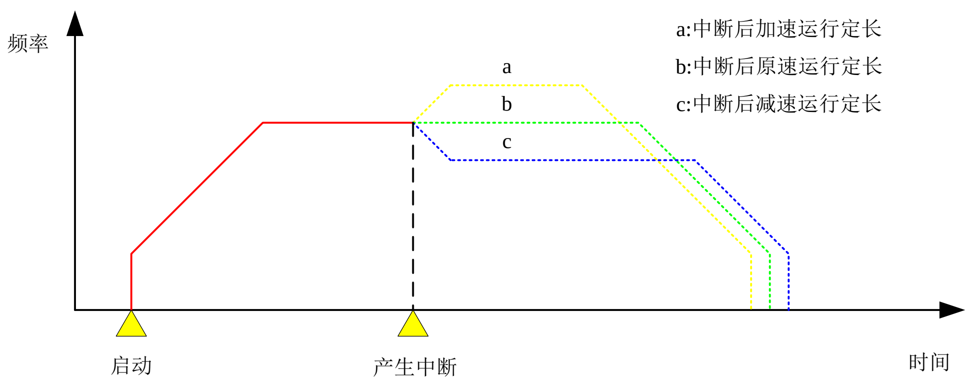
⑥ D2 is the running direction output port or bit variable. The state of this bit is determined by the controller's own pulse output state. When the pulse output is in forward running, the state of D2 is ON. When the pulse output is reverse running, the state of D2 is OFF.

⑦ S4 is the interrupt signal input port, and X0-X4 can be specified.

3. Pulse output



Output frequency change after interruption:



4. Soft components

The soft components functions related to the pulse output will be described in detail below.

◆ Pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count value (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output count value records the pulse number outputted by the port. This component has the characteristics of the D soft component and can be cleared by related commands or cleared by power-off. The component will not be cleared when the controller status is RUN->OFF.

◆ During pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8340	M8360	M8380	M8400	M8420	M8440	During pulse output

As shown in the figure below, when the Y304 pulse is output, M8360 is set. When the pulse output is completed, the M8360 is automatically reset.

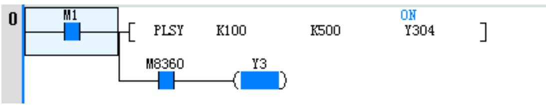


Figure a: During pulse output

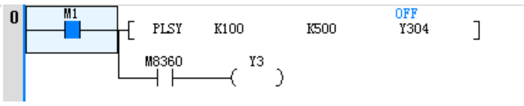
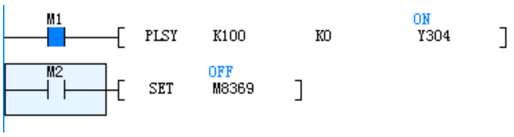


Figure b: Pulse output completed

◆ Pulse output stops

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stops

When the pulse output stop flag is set, the pulse output stops, and even if the energy flow is valid, the pulse will not output. The effect of the soft component is shown as the figure below. Before the soft component is set, the pulse is output normally. When the soft component is set, the pulse will stop output even if the energy flow is valid.



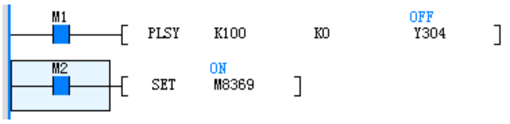


Figure a: Before the pulse-output-stop is set   Figure b: After the pulse-output-stop is set

◆ Pulse output completed

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8029						Pulse output completed

When this pulse output is completed, the corresponding M8029 will be set, but the completion of this pulse output does not affect the pulse output completion flag of other pulse commands. As shown in the figure below, after the execution of the first pulse command is completed, M10 is Set, but neither M11 nor M12 are set.

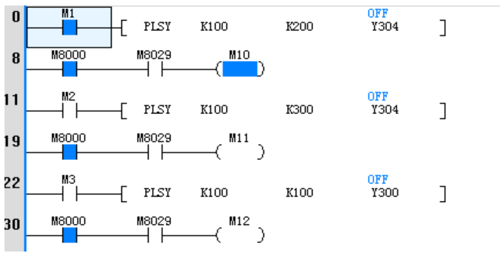


Figure a: M8029 soft component action

◆ Pulse output completion interrupt

If you want to enter the interrupt when the pulse output is completed, you can turn on the different output completion interrupt enable soft components for different Y ports. As shown in the table below, if you want to enter the interrupt after the Y300 pulse output is completed, you need to set M8352 to 1.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8352	M8372	M8392	M8412	M8432	M8452	Output completion

						interrupt enable

◆ Acceleration/deceleration time separate setting/pulse change is valid

If you want each output axis to have different acceleration/deceleration time, or you want to change the number of pulses during operation, you can set the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

Note: This soft component is a soft component shared by “Acceleration/Deceleration Separate Setting” and “Pulse Change Valid”.

when the above “Acceleration/deceleration time alone setting” soft component is OFF, the following parameters are used for the corresponding axis parameters:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (ms)

When the “Acceleration/deceleration time alone setting” soft component of an axis is ON, the corresponding axis parameters use the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	ase speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms)

5. Modify the frequency during operation

Modify the pulse output frequency during operation as shown in Figure a/b below. When modifying from V1 to V2 or from V2 to V3, if the speed is changed, the speed will not change suddenly, but will be accelerated or decelerated to the modified speed according to the set acceleration/deceleration time. In Figure a, the acceleration speed from V1 accelaration to V2 is the same as the acceleration speed from the base speed acceleration to V1; the acceleration speed from V2 deceleration to V3 is the same as the acceleration of the V3 deceleration to the base speed.

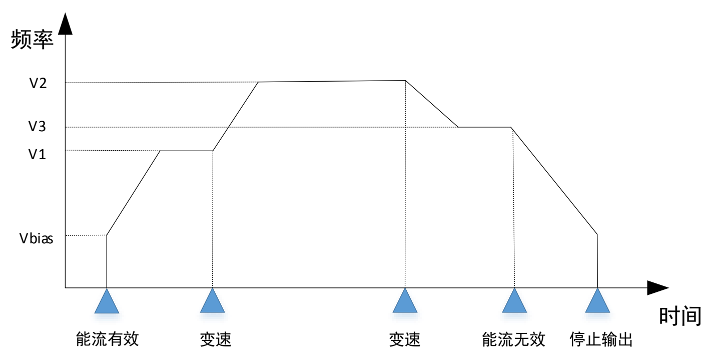


Figure a: Schematic diagram of PLSV2 pulse output change

In Figure b, the acceleration speed from base speed acceleration to V1 is the same as the acceleration speed from 0 acceleration to V2 and from 0 acceleration to V3; the acceleration from V1 deceleration to 0 is the same as the acceleration from V2 acceleration to 0 and from V3 decelarion to base speed.

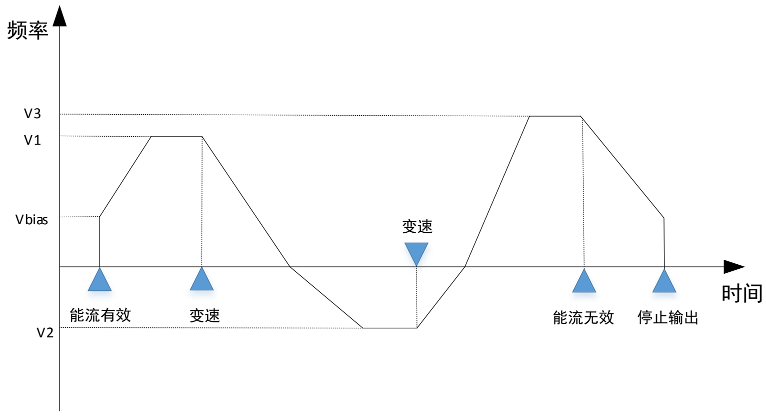


Figure b: Schematic diagram of PLSV2 pulse output change (reverse speed change)

5.2.11 DPIT Maximum Fixed Length Interrupt Positioning

Command

1. Command form

After starting, it accelerates to the set speed output frequency. When the interrupt input signal is detected, it immediately accelerates or decelerates to the output frequency of the position segment, and outputs the set number of pulses. When no interrupt input signal is detected, the output is the setting maximum pulse number.

<table>
<tr><td colspan="3">DPIT S1 S2 D1 D2 S3</td><td>Maximum Fixed Length Interrupt Positioning Command</td><td colspan="2">Command execution</td></tr>
<tr><td>S1</td><td>Maximum number of pulses</td><td colspan="2">S1: the setting number of position pulse output after interrupt
16-bit command S1+1, 32-bit command S1+2: the setting pulse output frequency after interruption</td><td rowspan="5">16-bit command (11step) DSZR continuous execution</td><td rowspan="5">32-bit command (21step) DDSZR continuous execution</td></tr>
<tr><td>S2</td><td>Output frequency</td><td colspan="2">S2: set speed segment pulse output frequency setting
16-bit command S2+1, 32-bit command S2+2: the position segment pulse output number setting after interruption</td></tr>
<tr><td>D1</td><td>Output port</td><td colspan="2">High speed pulse output port</td></tr>
<tr><td>D2</td><td>Output direction</td><td colspan="2">Pulse running direction port or bit variable</td></tr>
<tr><td>S3</td><td>Interrupt input</td><td colspan="2">Interrupt input signal port (range X0-X4)</td></tr>
</table>

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

注：Note: The gray soft components in the table indicate the soft components that can be supported.

① When the controller works with the servo driver, it outputs the pulse from the specified port at the specified pulse speed. After the interrupt signal is detected during the operation, the given number of pulse is output according to the position segment pulse output frequency after interrupt. The mechanism makes an offset motion based on the current position; if no interrupt signal is detected during operation, the set maximum pulse number is output.

② 1 is the number of pulse outputs, the unit is PLS. For 16-bit commands (DPIT), the setting range is

-32768~32767. For 32-bit commands (DDPIT), the setting range is -2,147,483,648~2,147,483,647; its positive and negative determines the pulse output direction.

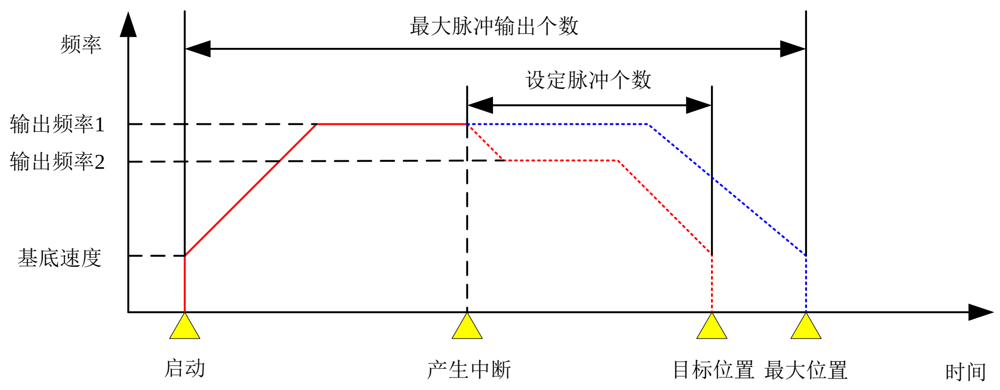
③ S2 is the pulse output frequency before and after the interrupt, the unit is Hz. For 16-bit command (DPIT), the setting range is 10~32767Hz; for 32-bit command (DDPIT), the setting range is 10~3,000,000Hz; The value of S2 can be modified during the command execution.

④ D1 is the pulse output port, and Y300/Y304/Y310/Y314/Y0/Y1 can be specified.

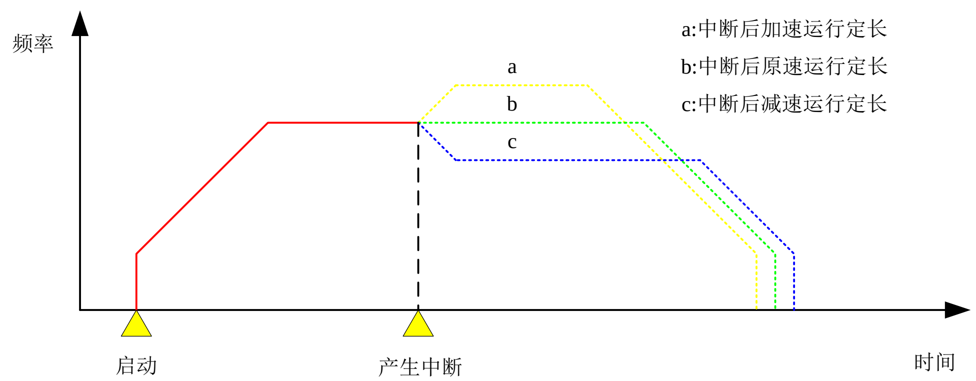
⑤ D2 is the running direction output port or bit variable. The state of this bit is determined by the controller's own pulse output state. When the pulse output is in forward running, the state of D2 is ON. When the pulse output is reverse running, the state of D2 is OFF.

⑥ S4 is the interrupt signal input port, and X0-X4 can be specified.

3. Pulse output



Output frequency change after interruption:



4. Soft components

The soft components functions related to the pulse output will be described in detail below.

◆ Pulse output count

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8340	D8360	D8380	D8400	D8420	D8440	Pulse output count value (PLS)
D8341	D8361	D8381	D8401	D8421	D8441	

The pulse output count value records the pulse number outputed by the port. This component has the characteristics of the D soft component and can be cleared by related commands or cleared by power-off. The component will not be cleared when the controller status is RUN->OFF.

◆ During pulse output

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8340	M8360	M8380	M8400	M8420	M8440	During pulse output

As shown in the figure below, when the Y304 pulse is output, M8360 is set. When the pulse output is completed, the M8360 is automatically reset.

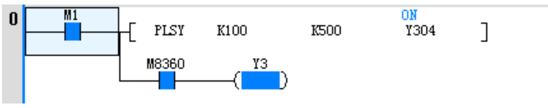


Figure a: During pulse output

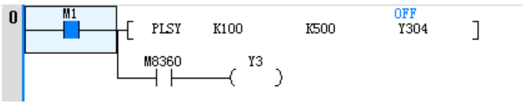


Figure b: Pulse output completed

◆ Pulse output stops

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8349	M8369	M8389	M8409	M8429	M8449	Pulse output stops

When the pulse output stop flag is set, the pulse output stops, and even if the energy flow is valid, the pulse will not output. The effect of the soft component is shown as the figure below. Before the soft component is set, the pulse is output normally. When the soft component is set, the pulse will stop output even if the energy flow is valid.

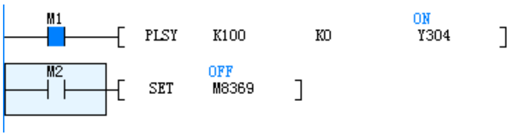


Figure a: Before the pulse-output-stop is set

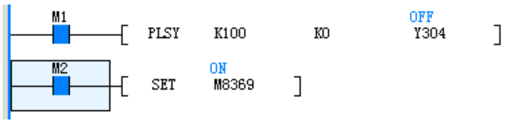


Figure b: After the pulse-output-stop is set

◆ Pulse output completed

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8029						Pulse output completed

When the pulse output of this section is completed, the corresponding M8029 will be set, but the completion of this pulse output does not affect the pulse output completion flag of other pulse commands. As shown in the figure below, after the execution of the first pulse command is completed, M10 is set, but neither M11 nor M12 are set.

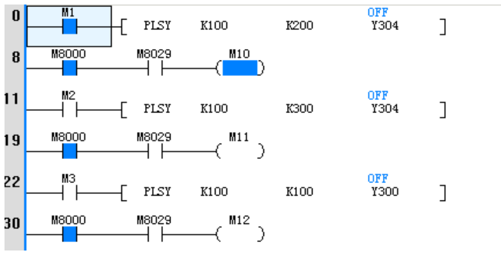


Figure a: M8029 soft component action

◆ ulse output completion interrupt

If you want to enter the interrupt when the pulse output is completed, you can turn on the different output completion interrupt enable soft components for different Y ports. As shown in the table below, if you want to enter the interrupt after the Y300 pulse output is completed, you need to set M8352 to 1.

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8352	M8372	M8392	M8412	M8432	M8452	Output completion

						interrupt enable

◆ Acceleration/deceleration time separate setting/pulse change is valid

If you want each output axis to have different acceleration/deceleration time, or you want to change the number of pulses during operation, you can set the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
M8350	M8370	M8390	M8410	M8430	M8450	Acceleration/deceleration time separate setting/pulse change is valid

Note: This soft component is a soft component shared by “Acceleration/Deceleration Separate Setting” and “Pulse Change Active”.

When the above “Acceleration/deceleration time alone setting” soft component is OFF, the following parameters are used for the corresponding axis parameters:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8500/D8501						Maximum speed (Hz)
D8502						Base speed (Hz)
D8503						Acceleration and deceleration time (ms)

When the “Acceleration/deceleration time alone setting” soft component of an axis is ON, the corresponding axis parameters use the following soft components:

Y300	Y304	Y310	Y314	Y0	Y1	Attributes
D8342	D8362	D8382	D8402	D8422	D8442	Maximum speed (Hz)
D8343	D8363	D8383	D8403	D8423	D8443	
D8347	D8367	D8387	D8407	D8427	D8447	Base speed (Hz)
D8348	D8368	D8388	D8408	D8428	D8448	Acceleration time (ms)
D8349	D8369	D8389	D8409	D8429	D8449	Deceleration time (ms)

5. Modify frequency during operation

Modify the pulse output frequency during operation as shown in Figure a/b below. When modifying from V1 to V2 or from V2 to V3, if the speed is changed, the speed will not change suddenly, but will be accelerated or decelerated to the modified speed according to the set acceleration/deceleration time. In Figure a, the acceleration speed from V1 acceleration to V2 is the same as the acceleration speed from the base speed acceleration to V1; the acceleration speed from V2 deceleration to V3 is the same as the acceleration of the V3 deceleration to the base speed.

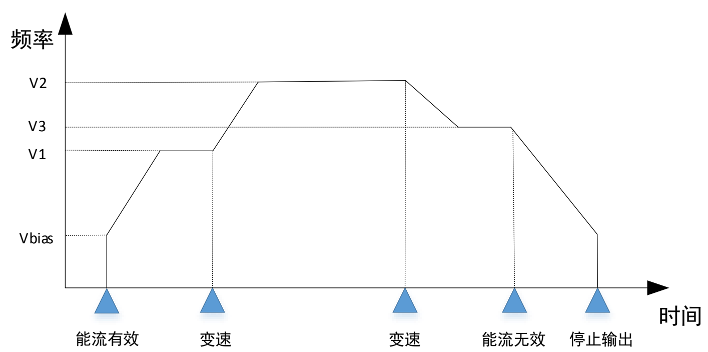


Figure a: Schematic diagram of PLSV2 pulse output change

In Figure b, the acceleration from base speed acceleration to V1 is the same as the acceleration from 0 to V2 and from 0 to V3; the acceleration from V1 deceleration to 0 is the same as the acceleration from V2 acceleration to 0 and from V3 decelarion to base speed.

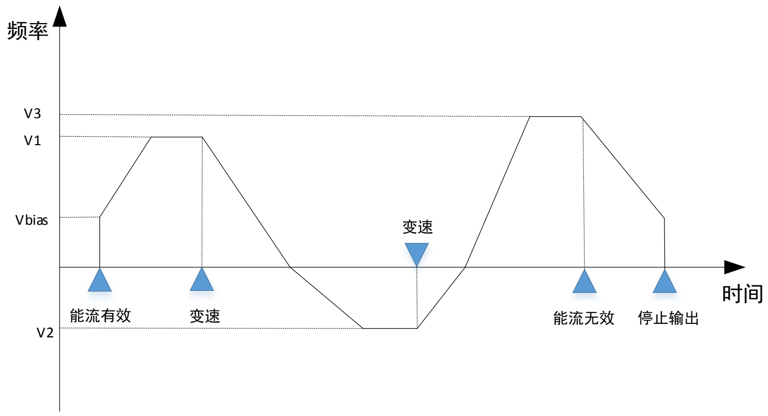


Figure b: Schematic diagram of PLSV2 pulse output change (reverse speed change)

5.3 High-speed Processing Commands

5.3.1 PWM Pulse Width Modulation Output Command

1. Command form

The modulated square wave is output according to the set pulse width and period.

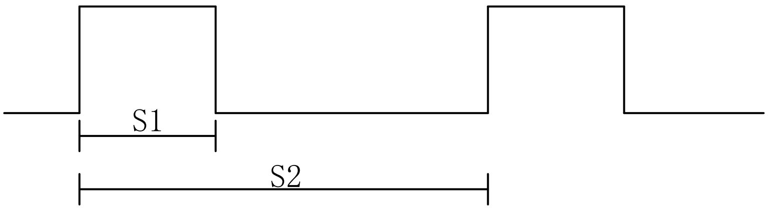
PWM S1 S2 D		Pulse width modulation output command	Command execution
S1	Output pulse width	Output pulse width setting	16-bit command (7step) PWM continuous execution
S2	Pulse period	Pulse period setting	
D	Output port	High speed pulse output port	

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

注：Note: The gray soft components in the table indicate the soft components that can be supported.

① The role of S1 and S2 is shown in the figure below. S1 is the high level duration and S2 is the time of one PWM period. S1 ≤ S2 should be satisfied.



② S1 is the output pulse width setting, which must satisfy S1≤S2, and the setting range is 0~32,767ms.

③ S2 is the output period setting, which must satisfy S1≤S2, and the setting range is 1~32,767ms.

④ D is the pulse output port, and Y0 or Y1 can be selected as the PWM pulse output.

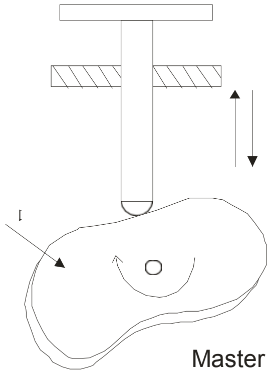
⑤ S1, S2 can be changed when the PWM command is executed.

Chapter 6 Electronic CAM

6.1 Introduction of Electronic CAM (E-CAM)

The traditional mechanical cam mechanism is composed of three parts: a cam, a follower and a frame.

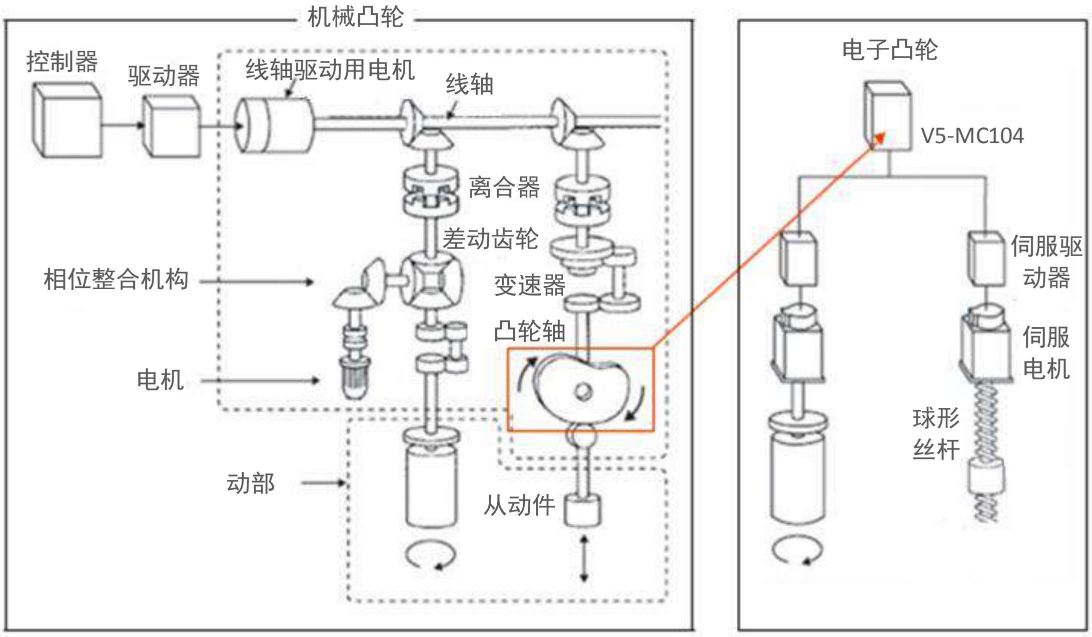
The mechanical cam is an irregularly shaped machine member, generally an input member of equal rotational speed, which can transmit motion to the follower via direct contact, so that the follower moves according to the set regularity. The follower is a passive member driven by a mechanical cam, and is generally an output member that generates unequal speed, discontinuity, and irregular motion. The frame is used to support the mechanical cam and the follower.



Follower

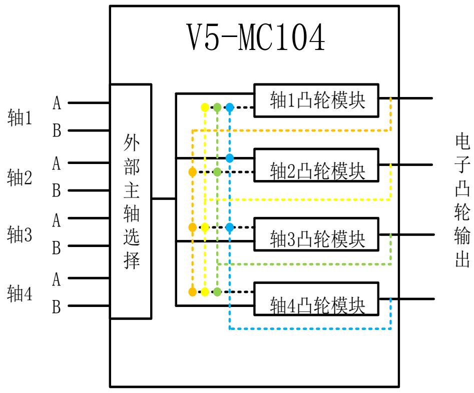
In contrast to traditional methods, the use of E-CAMs has the following advantages:

- More friendly user interface
- Different products require different cam curves, and software can be used to modify the E-CAM data in the E-CAM table without modifying the mechanism.
- Can have higher acceleration
- The operation can be smoother



6.2 Implementation of E-CAM

- Architecture diagram



Note: The solid line in the figure selects the external pulse input for the spindle; the dotted line in the figure is the spindle to select the virtual connection.

◆ Operation steps

First step	Second step	Third step
Initial setting	Spindle selection / cam table selection	Start/stop E-CAM
①Establish E-CAM data ②Terminal setting ③Input/output pulse form setting ④Input/output magnification ⑤Delayed start setting ⑥E-CAM start position setting	①Select real axis / virtual axis ②Select cam table	①E-CAM start ②E-CAM stop ③E-CAM forced stop

6.2.1 First step: Initial setting

6.2.1.1 Establish E-CAM data

The V5-MC104 provides two methods for establishing E-CAM data, each with its own advantages and disadvantages, and is suitable for different occasions. The method of establishing cam data is as follows:

Methods	Suitable occassions
Generate E-CAM data via VCAutoDesignsoft's CAM chart	No need to update E-CAM data points in real time
Write E-CAM data by CAMWR command	Real-time updating of E-CAM data points is required

6.2.1.2 Input/output pulse form setting

- If the input source of the slave axis pulse is a virtual axis, the spindle pulse output form can be determined according to the actual needs. The pulse input form of the slave axis defaults to the spindle pulse output form, and no user configuration is required.
- The output source of the slave axis pulse is the actual axis. Since the pulse count of the slave axis is fixed to the AB orthogonal input, the pulse output form of the spindle needs to be set to the AB orthogonal form (SDx61=1).

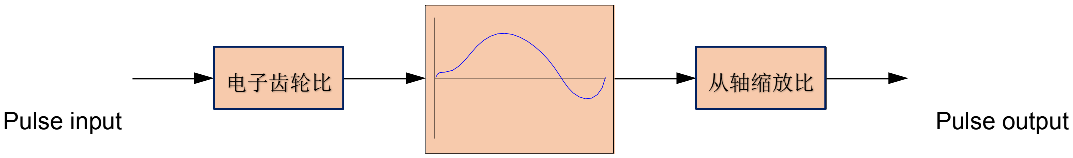
Slave pulse source	Slave pulse input form	Spindle pulse output form
Virtual spindle	The default follows the virtual spindle pulse output form, no configuration is required.	Choose according to actual needs Direction + pulse (SDx61[Note 1] = 0) A/B orthogonal (SDx61=1) CW/CCW (SDx61=2)
Actual spindle [note 2]	Fixed to AB orthogonal input	A/B orthogonal (SDx61=1)

Note 1: SDx61 is a pulse output type selection soft component. Axis 1 is SD61, axis 2 is SD161, axis 3 is SD261, axis 4 is SD761, Y0 is SD861, and Y1 is SD961.

Note 2: The actual spindle form refers to a form in which the spindle pulse output is connected to the slave pulse input port by wiring.

6.2.1.3 Input/Output Scale

The E-CAM can scale the cam table by scaling.



The special components used for the electronic gear ratio and the slave axis scaling ratio are as follows:

Electronic gear ratio / spindle scaling						Salve axis scaling					
Axis1	Axis2	Axis3	Axis4	Y0	Y1	Axis1	Axis2	Axis3	Axis4	Y0	Y1
SD44/SD45	SD144/SD145	SD244/SD245	SD744/SD745	SD844/SD845	SD944/SD945	SD73/100	SD173/100	SD173/100	SD773/100	SD873/100	SD973/100
spindle scaling						Salve axis scaling, default 100 for SD component system startup, that is, the ratio is 1					

After the electronic gear ratio and the slave axis scaling ratio change, the default is effective when the cam is started next time. If it needs to be valid in the currently running cam, you need to set the cam table data to modify the special SM component, which will take effect in the next cam cycle of the current running. After the valid, the cam table data Modify special SM component is automatically reset.

Cam table data Modify	Axis1	Axis2	Axis3	Axis4	Y0	Y1	Description
	SM83	SM183	SM283	SM783	SM883	SM983	Automatically reset to OFF after data Modify takes effect

6.2.1.4 E-CAM delay start

The E-CAM/electronic gear can be delay started by the SM soft component.

The special soft components for delayed start function are as follows:

Delay start soft components						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF

SM75	SM175	SM275	SM775	SM875	SM975	Enable delay start	Prohibit delayed start

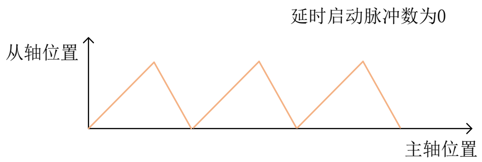
When the delay start function is turned on and the cam is enabled, the slave axis will start running after the spindle has run the set position.

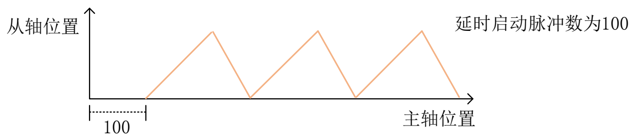
Salve axis start position setting soft component

Delay start pulse number (32bit)						Unit
Axis1	Axis2	Axis3	Axis4	Y0	Y1	
SD78,SD79	SD178,SD179	SD278,SD279	SD778,SD779	SD878,SD879	SD978,SD979	Pulse number

Note: The starting position can be set up to 1000000 pulses.

Example:





6.2.1.5 E-CAM starting position

The E-CAM/electronic gear can be set the starting position by the SM soft component. When the starting position is set to ON, the start position of the slave axis will change when the E-CAM is activated.

The starting position setting SM soft component is as follows:

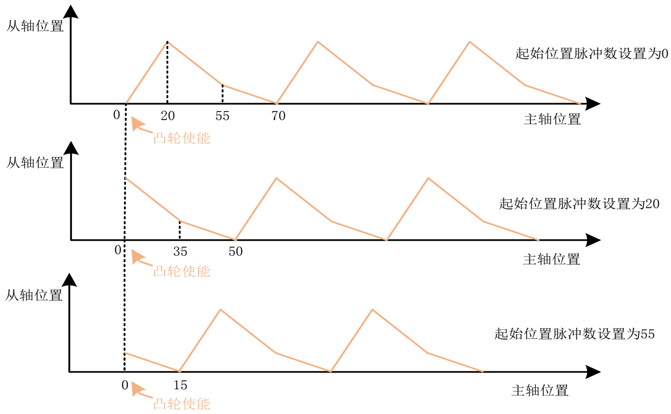
Starting position setting soft component						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF
SM68	SM168	SM268	SM768	SM868	SM968	Enable start position setting	Turn off start position setting

When the E-CAM is enabled after the start position setting is enabled, the correspondence between the position of the slave axis and spindle will be offset, depending on the pulse number at starting position.

The starting position pulse number SD soft component is as follows:

Starting position pulse number (32bit)						Unit
Axis1	Axis2	Axis3	Axis4	Y0	Y1	
SD68,SD69	Pulse number	SD268,SD269	SD768,SD769	SD868,SD869	SD968,SD969	Pulse number

Example



6.2.2 Second step: Spindle selection / cam table selection

6.2.2.1 Spindle selection

The E-CAM or gear function requires a spindle signal, and the source of the spindle signal can be selected as an external input or an internal virtual connection. The spindles are selected using special components as follows:

SM Spindle selection setting SM						SD Spindle selection SD					
Axis 1	Axis2	Axis3	Axis4	Y0	Y1	Axis 1	Axis2	Axis3	Axis4	Y0	Y1
SM71	SM171	SM271	SM711	SM871	SM971	SD71	SD171	SD271	SD771	SD871	SD971
OFF: Internal virtual connection						1: Axis 1 as a virtual spindle 2: Axis 2 as a virtual spindle 3: Axis 3 as a virtual spindle 4: Axis 4 as a virtual spindle 5: Y0 as a virtual spindle 6: Y1 as a virtual spindle					
ON: external input						1: Pulse input of axis 1 as the spindle 2: Pulse input of axis 2 as the spindle 3: Pulse input of axis 3 as the spindle 4: Pulse input of axis 4 as the spindle 5: Y0 pulse input as the spindle 6: Y0 pulse input as the spindle					

If the spindle is set to an internal virtual connection, the other axes can be used as virtual spindles without external wiring. For example, if axis 1 is the slave axis, axis 2, axis 3, axis 4, Y0, and Y1 can be selected as the virtual spindle of axis 1, but it is not possible to select itself as the spindle.

If the spindle is set to external input, the input channels of axis 1, axis 2, axis 3, axis 4, Y0, and Y1 can be arbitrarily selected as the external input spindle according to the value of the spindle selection SD component.

Note: When the spindle is set to external input, the external input pulse form must be AB orthogonal. See "4.2.1.3 Input/Output Pulse Form Settings" for details.

6.2.2.2 Cam table selection

By setting different cam tables to select SD component values, different cam tables or electronic gears can be selected for execution.

The special soft components selected for cam table are as follows:

Cam table selection					
Axis1	Axis2	Axis3	Axis4	Y0	Y1
SD70	SD170	SD270	SD770	SD870	SD970

The cam table selects the special component setting value and the cam table relationship:

Cam selects SD component value	Description
10	Electronic gear
11	Cam table 1
12	Cam table 2
13	Cam table 3
14	Cam table 4
15	Cam table 5
16	Cam table 6
Others	Not executed, reported 16262 error

6.2.3 Third step: Start/stop E-CAM

6.2.3.1 Start E-CAM

The E-CAM/electronic gear start can be selected as a software start or event trigger start;

The startup mode SM components are as follows:

Startup mode soft component						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF
SM70	SM170	SM270	SM770	SM870	SM970	Event triggered startup	Software startup

When the startup mode is selected as software startup, the cam enable soft component is as follows:

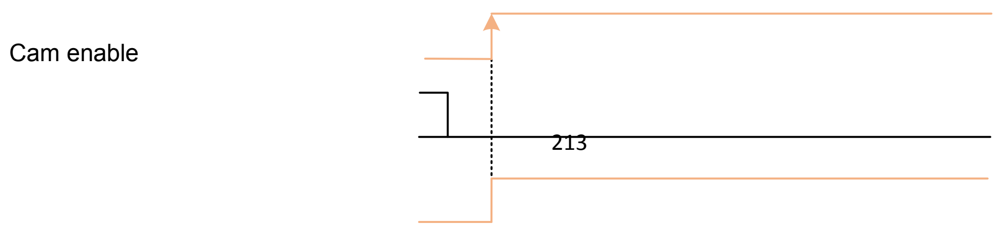
Cam enable soft component						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF
SM78	SM178	SM278	SM778	SM878	SM978	Cam enable	Cam not enabled

When the startup mode is selected as the event trigger start, the trigger startup mode is set as follows: (under development)

Cam enable soft component						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF
SM60	SM160	SM260	SM760	SM860	SM960	External input trigger startup	Compare interrupt trigger startup

Note: When the startup mode is selected as the event trigger startup, the cam enable soft component needs to be set to ON. For example, when the startup mode of the axis 1 is selected as the event trigger startup (SM70=1), the trigger startup mode is set to ON (SM60=1), then if you want the axis 1 to start the E-CAM when the external input triggers, you need to turn the cam enable position ON (SM78=1).

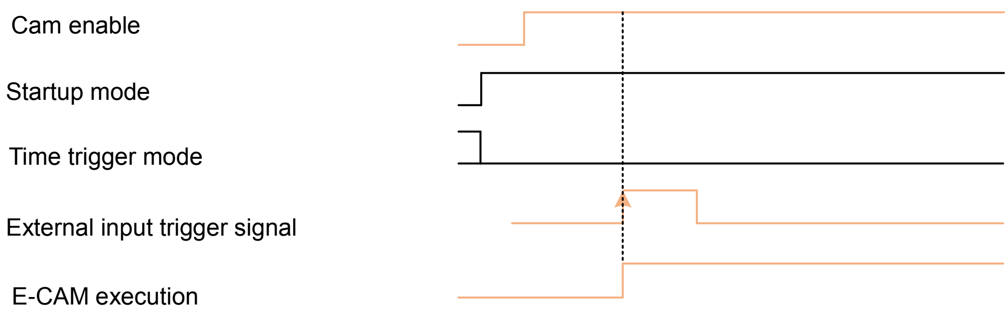
Software startup timing sequence:



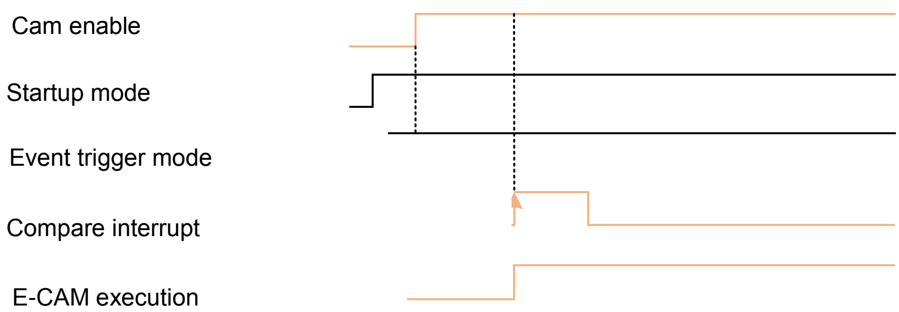
Startup mode

E-CAM execution

External input trigger startup timing sequence: (under development)



Compare interrupt trigger start timing sequence: (under development)



Note: Currently only software startup is supported, SM70 needs to be set to OFF.

6.2.3.2 Stop the E-CAM

When the E-CAM is executed, the cam enable special soft component can be turned OFF to stop the execution of the E-CAM. The stop of the E-CAM selects whether to stop immediately or stop after this cycle completed according to the stop mode.

E-CAM stop mode setting:

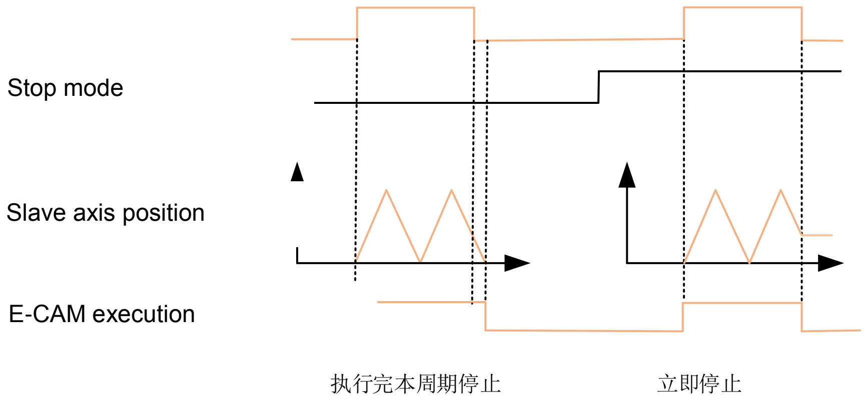
Stop mode setting						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF
SM81	SM181	SM281	SM781	SM881	SM981	Stop immediately Note 1	Stop after this cycle is completed Note 2

Note 1: Immediate stop means that the E-CAM stops immediately when the cam enable is turned OFF or the stop signal is valid when the E-CAM is executed.

Note 2: Stop after this cycle is completed, it means that when the cam enable is turned OFF or the stop signal is valid when the E-CAM is executed, the E-CAM stops after executing this cycle.

Stop timing sequence:

Cam enabled



When the stop mode of the E-CAM is set to stop after this cycle is executed, the cam enable position is OFF, the E-CAM will stop after running this cycle, but if the E-CAM execution is needed to stop immediately, the E-CAM running flag can be turned ON to OFF to forced stop the E-CAM.

Forced stop soft component

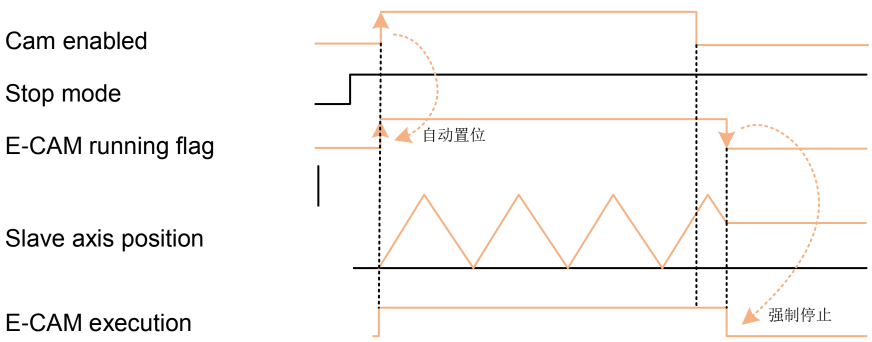
Forced stop soft component						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	ON→OFF
SM89	SM189	SM289	SM789	SM889	SM989	E-CAM execution note 1	Forced stop cam execution Note 2

Note 1: The E-CAM is in the monitoring state during execution, indicating that the E-CAM is being executed and could be automatically set by the software.

Note 2: If the E-CAM is being executed, turn off the stop soft component to forced stop the E-CAM.

Forced stop timing sequence

The following figure shows the E-CAM diagram in which the stop mode is set to stop after this cycle is executed. When the E-CAM enable is turned OFF, the E-CAM will not stop immediately, but if the E-CAM running flag is in the OFF position, it will force stop the E-CAM.



6.2.3.3 Periodic/non-periodic selection

E-CAMs can be selected for periodic or non-periodic execution, with special SM and SD components.

The periodic/non-periodic selection uses special components as follows:

SM Periodic/non-periodic SM						SD Cycle number setting SD					
Axis1	Axis2	Axis3	Axis4	Y0	Y1	Axis1	Axis2	Axis3	Axis4	Y0	Y1
SM73	SM173	SM273	SM773	SM873	SM973	SD72	SD172	SD272	SD772	SD872	SD972
OFF: Non-periodic execution						Non-periodic execution times, up to 255 cycles					
ON: Periodic execution						—					

Periodic execution: After the E-CAM is started, the relationship of the E-CAM table setting is executed continuously until a stop command is received;

Non-periodic execution: After the E-CAM is started, it automatically stops after the set cycle is executed. The number of cycles of non-periodic execution is set by SD components (SD72, SD172, SD272, SD772, SD872, SD972), and the maximum can be set to 255 cycles.

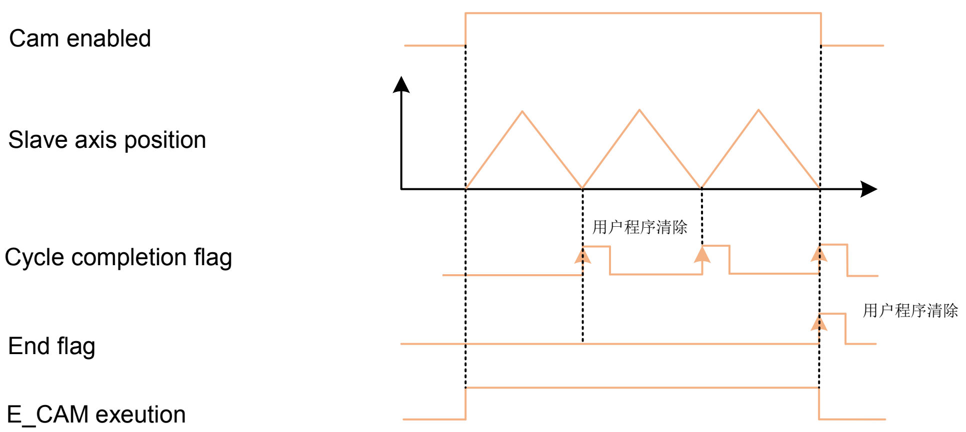
6.2.3.4 Cycle Completion and End Flag

Each time the E-CAM completes a cycle, the system automatically sets the cycle completion flag special SM component to ON. After the cycle completion flag is set, it remains ON. If it is necessary to detect the completion of the next cycle, the user program needs to clear the cycle completion flag to OFF. When the next cycle is completed, the system sets the cycle completion flag to ON again.

When the execution of the E-CAM/electronic gear ends, the system automatically sets the end flag special SM component to ON. The end flag is cleared to OFF by system each time the cam is enabled, and can also be cleared by the user program.

The special SM components used for the cycle completion and end flags are as follows:

	Axis1	Axis2	Axis3	Axis4	Y0	Y1	Description
Cycle completion flag	SM79	SM179	SM279	SM779	SM879	SM979	The E-CAM is set to ON every time one cycle is completed.
End flag	SM80	SM180	SM280	SM780	SM880	SM980	Turn ON when the E-CAM/ electronic gear completes execution.



6.3 E-CAM key point Modify

6.3.1 DCAMWR E-CAM data Modify

1.Command form

Modify the E-CAM table data.

DCAMWR m1 m2 D n		E-CAM data Modify	Command execution
m1	Cam table	Specify the CAM table to be modified	32-bit command (17step) DCAMWR continuous execution
m2	Starting point	Set the starting point of the cam table data to be modified	
D	data	Data storage address to be modified	
n	Points number	Number of key point data points to be modified	

2.Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that are not supported.

- ■ m1 specifies the cam table to be modified, m1=k11~k16 respectively specify the E-CAM table ID1~ID6;
- ■ m2 sets the starting point of the E_CAM key point that needs to be modified, m2=k2~k360;
- ■ D modifies the starting address of the data, occupying multiple consecutive address units starting with D. Each key point occupies two 32-bit registers to mark the spindle and slave axis positions respectively, that is, each key point needs to occupy 4 address units;
- ■ n The number of key points to be modified, m2+n-1 needs to be less than or equal to the number of downloaded key points;

3.Attention points

- ● The DCAMWR command can only execute one at a time. If more than two DCAMWR commands are required in the program, the next command can be started after the previous command stops for one scan cycle.
- ● DCAMWR is a multi-cycle execution command, and the special register SM82 changes from OFF to ON to indicate that the Modify is completed.
- ● DCAMWR Modify completion means that the cam table data in the PLC has been changed. At this time, the cam table data is uploaded or read back to the modified data. If the E-CAM is running, the data after the Modify cannot be effective immediately, and the special flag bits (SM83, SM183, SM283, SM783) need to be set, so that it can be valid for the next cam cycle; if

the E-CAM is not executed, or it does not need to be active in the running E-CAM, the next time the E-CAM is started, the cam table data is automatically valid after Modify.

- The first point of the E-CAM table is the starting point data, which cannot be modified, so m2 must be more than 1; the command parameter m2+n-1 needs to be less than or equal to the number of downloaded key points.
- When modifying the cam table data, the spindle position data must be more than the spindle position of the previous point and less than the spindle position data of the latter point, otherwise the motion controller reports 16268 error.
- The DCAMWR command specifies a modified cam table that must be present in the motion control system, that is, the cam table has been downloaded to the motion control system via VCAutoDesignsoft.

4.E-CAM table data Modify related special register

The special function registers associated with E-CAMs are listed below.

<table>
<tr><th>Soft components</th><th>Axis No.</th><th>Definition</th></tr>
<tr><td>SM82</td><td>All the axis</td><td>SM82 changes from OFF to ON to indicate that the Modify is completed.</td></tr>
<tr><td>SM83</td><td></td><td rowspan="6">The effective flag of the E-CAM Modify operation:
ON: The modified data is valid in the next cam cycle, and is automatically reset to OFF after it takes effect;
OFF: The modified data takes effect the next time the cam is started.</td></tr>
<tr><td>SM183</td><td></td></tr>
<tr><td>SM283</td><td></td></tr>
<tr><td>SM783</td><td></td></tr>
<tr><td>SM883</td><td>Y0</td></tr>
<tr><td>SM983</td><td>Y1</td></tr>
</table>

Supportable commands

NO	m2 value	Description
1	K2~K360	Single or multiple point Modify commands
2	K361	Ejector pin Modify commands
3	K362	Single data Modify command (one data in a set of points)
4	K10000	Full points Modify
5	K10002	Flying shear
6	K10004	Chasing
7	K10030	Packing machine specified
8	K10031	Packing machine specified
9	K10032	Packing machine specified
10	K10033	Packing machine specified
11	K10034	Packing machine specified

6.3.2 DCAMRD reads E-CAM data (under development)

1.Command form

Read E-CAM table data.

DCAMRD m1 m2 D	Read E-CAM table data.	Command execution

n			
m1	Cam table	Specify the cam table to read	32-bit command (17step) DCAMWR continuous execution
m2	Starting point	Set the starting point of the cam table data to read	
D	data	Store the read cam table data address	
n	Points number	Number of keypoint data points that need to be read	

2.Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that can be supported.

- m1 specifies the cam table to read, m1=k11~k16 respectively specify the E-CAM table ID1~ID6;
- m2 sets the starting point of the E_CAM key point that needs to read, m2=k2~k360;
- D stores the read data starting address of cam table, occupying multiple consecutive address units starting with D. Each key point occupies two 32-bit registers to mark the spindle and slave axis positions respectively, that is, each key point needs to occupy 4 address units;
- n The number of key points to read, m2+n-1 needs to be less than or equal to the number of downloaded key points;

3.Attention points

- DCAMRD reads the E-CAM table data, and the specified cam table must exist in the motion control system, that is, the cam table has been downloaded to the motion controller via VCAutoDesignsoft.
- The parameter m2+n-1 needs to be less than or equal to the number of downloaded keypoints.

6.3.3 Full points Modify commands

1. Command form

Modify the E-CAM table data.

DCAMWR m1 m2 D n		Full points Modify commands	Command execution
m1	Cam table	Specify the CAM table to be modified	32-bit command (17step) DCAMWR continuous execution
m2	Starting point	Set the starting point of the cam table data to be modified	
D	data	Data storage address to be modified	
n	Points number	Number of key point data points to be modified	

2. Operands

Operands	Bit soft components							Word soft components																
	System. User							System. User					Digit designation					Indexing			Constant		Real number
m1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
m2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
n	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that are not supported.

When m2=K10000, it is the full point Modify function. The parameters of CAMWR are defined as below:

Parameters	Definition	Description	Values
M1	Cam table	Cam table number, to select the table to modify	K11~K16
D	Start register	Data point storage register	D component
n	Modify points	E-CAM total points	2~25

For example:

DCAMWR K11 K10000 D100 K3

The command modifies the E-CAM table 1 and modifies all the E-CAM tables. The modified number of points is 3 points, and the starting position of the point is D100. The specific data to be modified is shown in the table below.

Modify points	Position	Data storage register
Point 1	Spindle axis position	D100D101
	Slave axis position	D102D103
	Speed ratio (float type)	D104D105
	Line type	D106D107
Point 2	As above	D108D109

Point 3	As above	D116D117

6.3.4 Single Data Modify Command DCAMWR Command

1. Command form

et cam execution data

DCAMWR S1 S2 D1 D2		Single data Modify command	Command execution
S1	Cam table		32-bit command (13setp) Continuous execution
S2	Fixed K362		
D1	Data address (spindle, slave axis, speed ratio, type)		
D2	Row/ Column of data		

2. Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
s1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
s2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that are not supported.

Function and action description:

S1	Cam table K11: the first cam table; K12, K13 are second, third, etc.
S2	Fixed to K362, decimal
D1	Data address
D2	The row/column of the data is calculated as follows: row = D2 / 100; column = D2% 100;

Note 1: The row of data should not exceed the maximum number of points.

Note 2: The data column ranges from 1 to 4, corresponding to the spindle, slave axis, speed ratio, and type.

Note 3: When modifying the data type, pay attention to the type of the data.

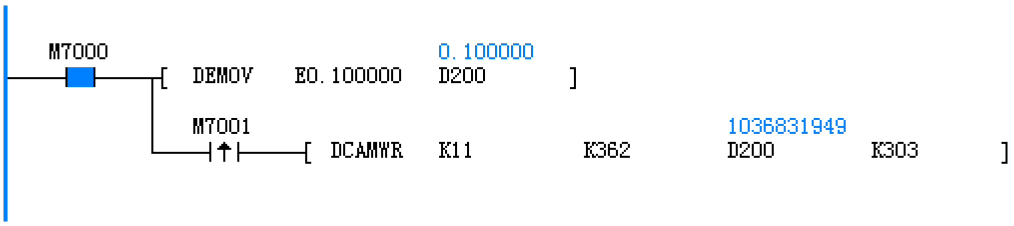
Data column	1	2	3	4
Data type	Spindle axis	Slave axis	Speed compensation	Type
Corresponding	32 bit	32 bit	Float	16 bit

For example:

The original cam table is:

	Add	Del	M-Pos	S-Pos	PU Speed	Type
1			0	0	1.00	NA
2	⊞	☒	2500	7000	2.80	Line
3	⊞		10000	30000	3.07	Line

The Modify commands are as follows:



The execution result: the third point, the third column data 3.07 is modified to 0.1.

Note: Try to modify it with the eadge trigger method.

Note 2: SM82 will be set to ON after the Modify takes effect and will be set to OFF if it fails.

Note 3: The use of this function, please note that the data protection while Modify is at least the previous point, otherwise it will lead to unpredictable errors.

6.3.5 Ejector Modify command

1.Command form

Modify the E-CAM table data.

DCAMWR S1 S2 D1 D2		Ejector Modify command	Command execution
S1	Cam table		32-bit command (13setp) Continuous execution
S2	Fixed K361		
D1	Starting address of ejector pins		
D2	Number of ejector pins		

2.Operands

Operands	Bit soft components							Word soft components															
	System. User							System. User					Digit designation					Indexing			Constant		Real number
s1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
s2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: The gray soft components in the table indicate the soft components that are not supported.

Function and action description:

S1	Cam table K11: the first cam table; K12, K13 are second, third, etc.
S2	Fixed to K361, decimal

D1	Data starting address, an example as below:
D2	Number of ejector pins, 0 position does not set ejector data

Note: a. Before the cam is enabled, it can be only written by this command.

b. Modifys are not allowed during operation.

Data format table:

No.	Address	Number of digits	Range	Description
1	D1	16 bit	0,1	Component Type 0: M Component 1: Y Component
2	D1+1	16 bit	Variable	The address of M component is 0~7679, and the address of Y component is 0~191.
3	D1+2	16 bit	0,1	Action type 0: OFF, 1: ON
4	D1+3	16 bit		Reserved
5	D1+4	32 bit		Ejector pins data, pulse unit

Note: Pay attention to octal and decimal conversion for Y address, and K8 for Y10.

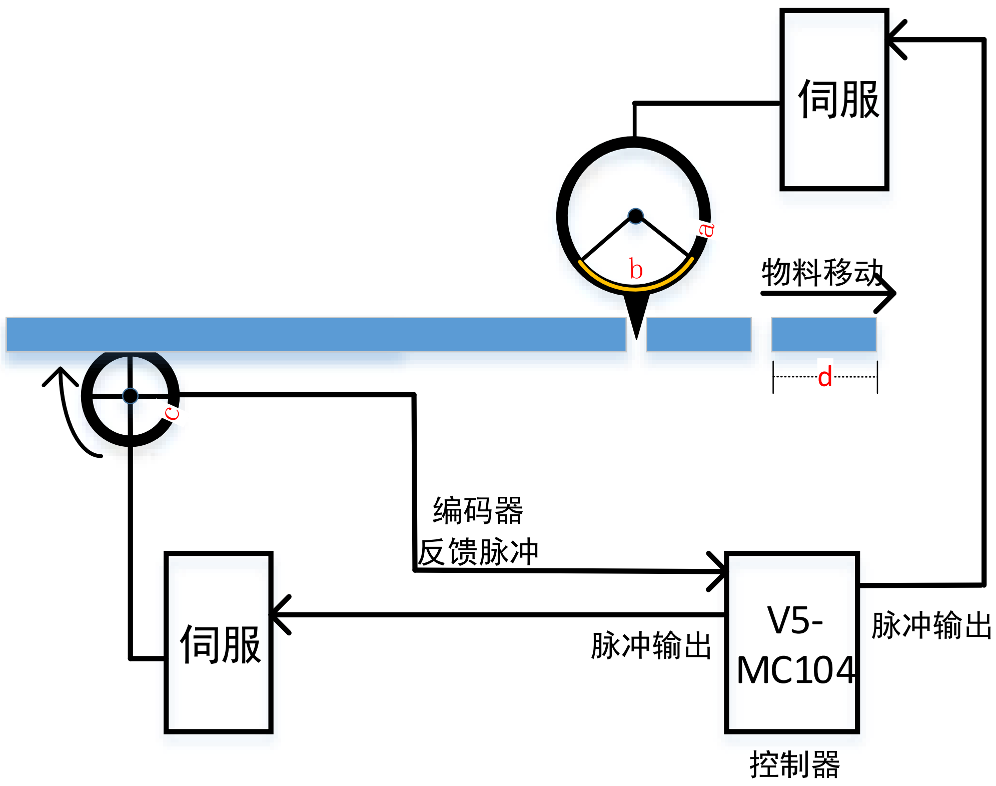
6.4 Flying shears

6.4.1 Function Description

6.4.1.1 Introduction of flying shear

In the feeding and cutting application, the traditional method is to use the stop-and-go mode. The feeding shaft first goes to a fixed length, and then the cutting shaft acts, and then repeats the process of “feeding stop” and “cutting stop”. The shortcoming of this method is that the acceleration and deceleration in the process of stop-and-go the feed shaft can not improve the production efficiency. Therefore, the new method is to achieve the feeding without stopping. Generally, there are two methods of feeding and cutting: flying shears and chasing, and the difference between them is that the chasing is a reciprocating motion, while the flying shear is a co-directional motion, and the set CAM table curve is also different. The flying shear function will be described in detail below.

6.4.1.2 Principle of flying shear



In the picture:

a: One circle length of slave axis (mm)

b: slave axis synchronization zone length (mm)

c: One circle length of spindle axis(mm)

d: material length (mm)

The one circle length of slave axis: the length of slave axis rotating one circle, that is, the circumference of slave axis, which is equal to the diameter of the slave axis *π

Slave synchronization zone length: The area where the spindle speed and the slave axis speed are the same is the synchonization zone length.

The one circle length of spindle axis: the length of spindle axis rotating one circle, that is, the circumference of the spindle axis, which is equal to the diameter of the slave axis *π

Material Length: The length of the material to be cut.

6.4.1.3 Slave axis movement mode

In the actual chasing system, the slave axis follows the spindle for E-CAM motion. The relationship between the length of the material and the circumference of the cutter is different, and the key points of the E-CAM are also different. The following describes how the slave axis follows the spindle motion.

◆ Cut short materials

When 2* one circle length of slave axis > material length, the system will determine that the material is short material, and plan according to the curve planning method of short material. The planning point of short material is 4 points.

As shown in Figure 1.1 below, the starting point of the camming is the midpoint of the synchronization zone. The slave axis moves synchronously from the yellow dot to the green dot, then moves from the green dot to the red dot, and then moves from the red dot to the yellow dot to complete a periodic movement.

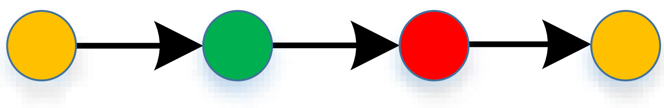
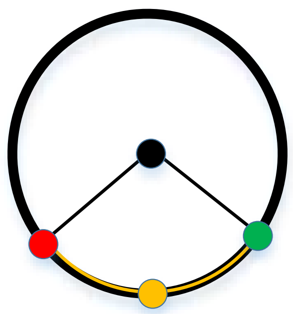
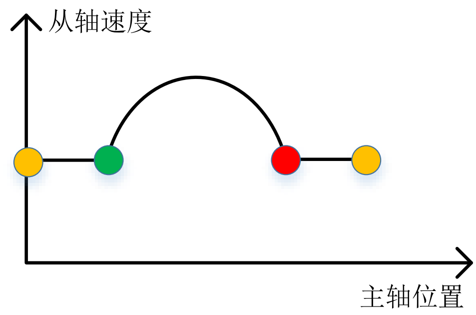


Figure 1.2 shows the curve relationship between spindle and slave axis when cutting short materials. The yellow to green segments and the red to yellow segments are all synchronization zones; the green to red segments are shifting zones.





Short material point planning Short-term spindle-slave relationship curve

◆ Spindle-slave position calculation

Sign	Name
L	Cutting length (mm)
L1	One circle length of slave axis (mm)
P1	One circle pulse number of slave axis
D	Synchronization zone length (mm)
L2	One circle length of spindle axis(mm)
P2	One circle pulse number of spindle axis

Keypoints	Position (number of pulses)	Calculation formula
Point 1	Spindle axis	0
	Slave axis	0
Point 2	Spindle axis	$\frac{P2}{2 \times L2} \times D$

	Slave axis	$\frac{P1}{2\times L1}\times D$
Point 3	Spindle axis	$\frac{2\times L-D}{2\times L2}\times P2$
	Slave axis	$P1-\frac{P1}{2\times L1}\times D$
Point 4	Spindle axis	$\frac{P2}{L2}\times L$
	Slave axis	P1

◆ Cut long material

When 2* one circle length of slave axis < material length, the system will determine that the material is long material, and plan according to the curve planning of long material. The planning point of long material is 6 points. When the long material is cut, the slave axis has a waiting area, at which time the spindle moves and the slave axis is at rest.

As shown in Figure 1.3 below, the cam meshing point is the midpoint of the synchrosphere. The slave axis runs synchronously from the yellow dot to the green dot, then runs from the green dot to the blue dot, then from the blue dot to the red dot, and finally from the red dot synchronously to the yellow dot.

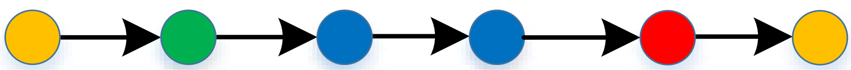
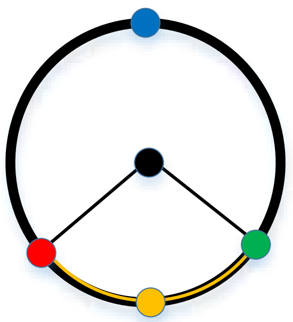
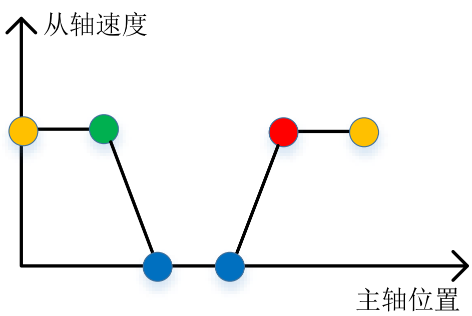


Figure 1.4 shows the curve relationship between the spindle and the slave axis when cutting long materials. The yellow dot to the green dot and the red dot to the yellow dot are the synchronization areas, and the two blue dots are the stop areas. Green dot to blue dot, blue dot to red dot are non-synchronized zones.



Long material point planning



Long-term spindle-slave relationship curve

◆ Spindle and slave position calculation:

Sign	Name
L	Cutting length (mm)
L1	One circle length of slave axis (mm)
P1	One circle pulse number of slave axis
D	Synchronization zone length (mm)
L2	One circle length of spindle axis(mm)
P2	One circle pulse number of spindle axis

Keypoints	Position (number of pulses)	Calculation formula
Point 1	Spindle axis	0
	Slave axis	0
Point 2	Spindle axis	$\frac{P2}{2 \times L2} \times D$
	Slave axis	$\frac{P1}{2 \times L1} \times D$
Point 3	Spindle axis	$\frac{L1 - D}{L2} \times P2 + \frac{P2}{2 \times L2} \times D$
	Slave axis	$\frac{P1}{2}$
Point 4	Spindle axis	$\frac{P2}{L2} \times L - \left(\frac{L1 - D}{L2} \times P2 + \frac{P2}{2 \times L2} \times D\right)$
	Slave axis	$\frac{P1}{2}$
Point 5	Spindle axis	$\frac{P2}{L2} \times L - \frac{P2}{2 \times L2} \times D$
	Slave axis	$\mathrm{P1} - \frac{P1}{2 \times L1} \times D$
Point 6	Spindle axis	$\frac{P2}{L2} \times L$
	Slave axis	P1

6.4.1.4 Flying shear operation steps

Step 1: return to the origin (if the spindle-slave data is saved, the origin return is not necessary).

Step 2: Initialize the E-CAM and write the flying shear data by the CAMWR command.

Step 3: start the E-CAM according to the current spindle-slave position relationship.

Please refer to "1.2 Application Examples" for more details.

6.4.1.1 DCAMWR_K10002 Command

This function is written by the DCAMWR command. The form of the command is as follows:

DCAMWR m1 m2 D n			Flying shear parameter writing	Command execution
m1	Cam table	Specify the CAM table to be modified		32-bit command (17step) DCAMWR continuous execution
m2	Subset selection			
D	Data	Data storage address to be modified		
n	Reserved			

- m1 position cam table selection function, select the modified cam table, m1 value range is K11~K16.
- m2 is a subset of the application function. When m2 is set to K10002, it is a dedicated function of flying shears.
- D is the storage address of the data to be modified.
- For the dedicated function block of the flying shear, n is a reserved bit.

For example:

DCAMWR K11 K10002 D100 K0

This command indicates that a set of flying shear parameters headed by D100 is written into the CAM table 1. The relevant parameters are stored in the 12 D soft components headed by D100. The parameters are as follows:

Data type	Register	Definition	Unit
Float	D100D101	One circle length of spindle axis	mm
Unsigned integer	D102D103	One circle pulse number of spindle axis	pcs
Float	D104D105	One circle length of slave axis	mm
Unsigned integer	D106D107	One circle pulse number of slave axis	pcs
Float	D108D109	Synchronization zone length	mm
Float	D110D111	Cutting length	mm

6.4.1.2 Parameters take effect

After the flying shear function parameter is written, it can be selected to take effect with the next cam cycle, or it can be selected when the E-CAM is started next time. The effective mode depends on the M soft components.

Effective mode						Soft component status	
Axis 1	Axis 2	Axis 3	Axis 4	Y0	Y1	ON	OFF
SM83	SM183	SM283	SM783	SM883	SM983	Effective for the next cam cycle	Effective the next time the cam is started

When the corresponding M soft component is set to ON, it will take effect for the next cam cycle. When the corresponding M soft component is set to OFF, it will take effect when the E-CAM is started next time.

When the flying shear parameters take effect, the software automatically turns the SM82 soft component to ON (this soft component is common to all axes).

6.4.2 Application examples

6.4.2.1 Basic case

A flying shear case will be given below. Its flying shear parameters are as follows:

- Spindle diameter: 150mm
- Spindle transmission ratio: 1:10
- Diameter of the slave axis: 200mm
- Synchronization area angle: 60°
- Slave axis transmission ratio: 1:3
- Cutting material length: 500mm

According to the above system parameters, the value that needs to be set by the CAMWR command is calculated.

(Servo adopts SD700 servo, internally set 10000 pulses for servo motor one circle)

One circle length of spindle axis =150*π=471

One circle pulse number of spindle axis = 10000*10=100000

One circle length of slave axis = 200 * π = 628

One circle pulse number of slave axis= 10000*3=30000

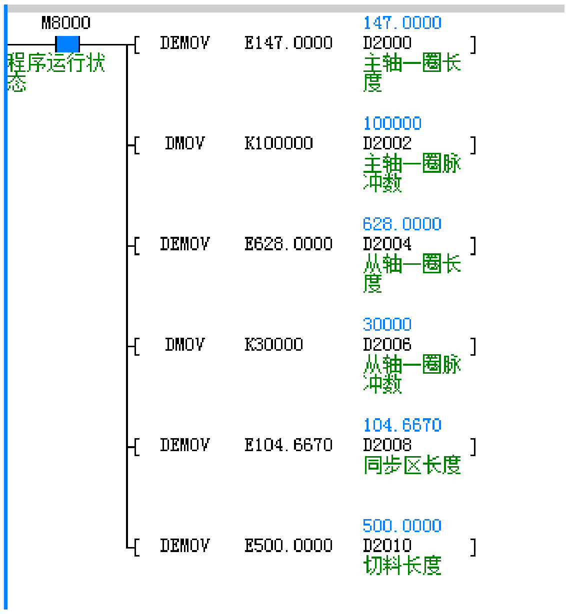
Synchronization zone length = 60 ° / 360 ° * one circle length of slave axis = 104.667

Parameters	Unit	Setting value
One circle length of spindle axis	mm	471
One circle pulse number of spindle axis	pcs	100000
One circle length of slave axis	mm	628
One circle pulse number of slave axis	pcs	30000
Synchronization zone length	mm	104.667
Material length	mm	500

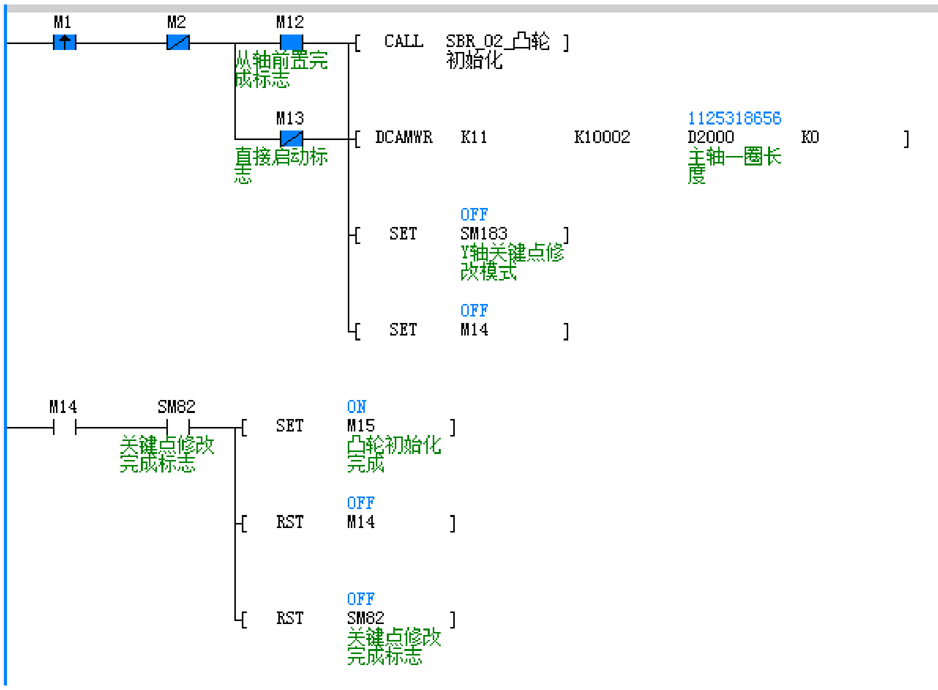
Here's a simple example of using chasing:

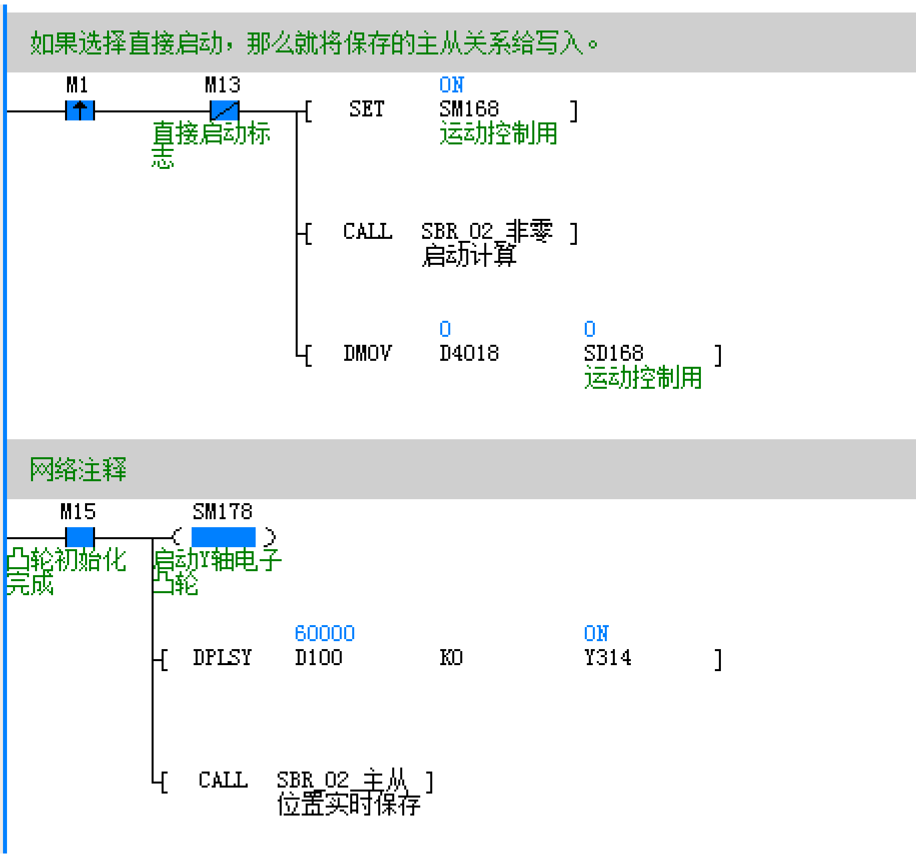
Step 1: set the basic parameters of the E-CAM

Step 2: Set the flying shear parameters

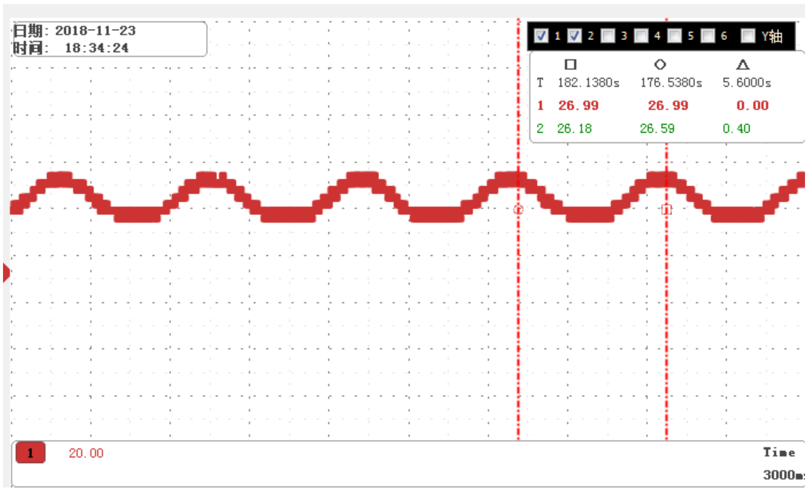


Step 3 : After the initialization, start the E-CAM and run according to the flying shear curve. The main function is shown below:





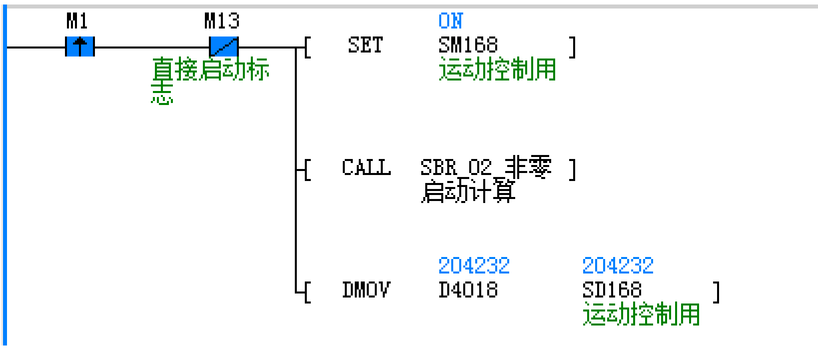
The actual running results are shown in the following figure:



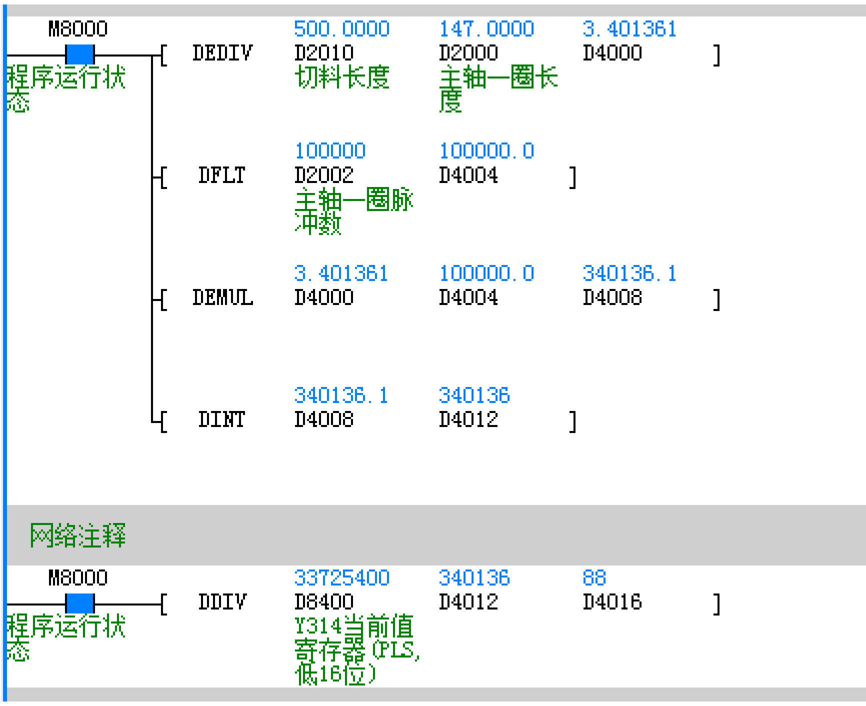
6.4.2.2 Power-on operation case

If it needs to start the system immediately, you can do with a non-zero start function.

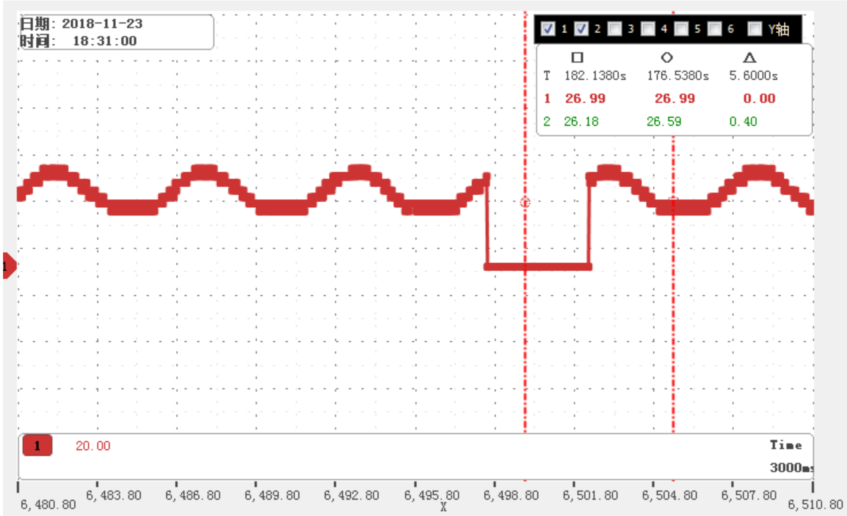
The ladder diagram is as shown below



The ladder diagram of the non-zero start calculation is shown below:



The final operation result is shown in the figure below. It can be seen from the figure that when the PLC is turned OFF, if the PLC is turned ON again, it can be started immediately, and the startup still runs from the position when stop.

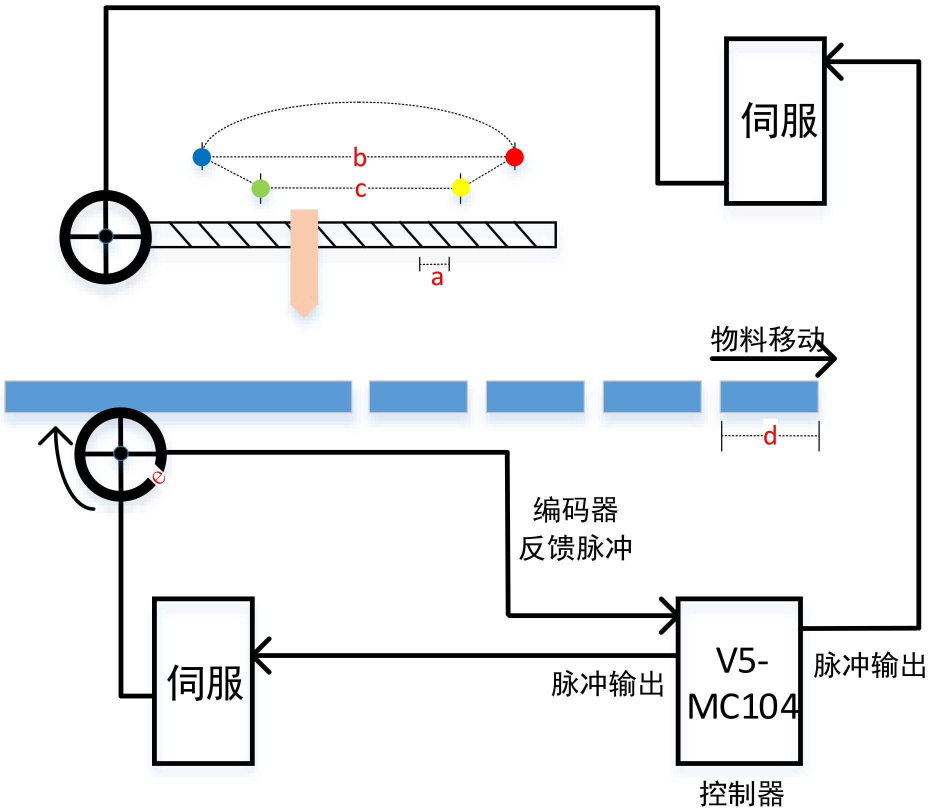


6.5 Chasing

6.5.1 Function Description

6.5.1.1 Introduction of chasing

Chasing is a process of vertically cutting and filling in motion. The whole process is shown in the figure below. The V5-MC104 with two servo axes---feeding axis and chasing axis: The feeding axis mainly drives the material to move forward and the chasing axis mainly tracks the feeding axis, so that they are in the synchronization area.



In the picture:

a: One circle lead of slave axis (mm)

b: Movement range of slave axis (mm)

c: synchronization area (mm)

d: material length (mm)

e: One circle length of spindle axis (mm)

One circle lead of the slave axis is the slave axis movement distance when the slave motor rotates one revolution. If the slave axis adopts the screw structure, the value refers to the lead of the screw.

The movement range of slave axis refers to the movement range of the slave axis. The slave axis moves only within the safe range. The blue dot is the starting point of the movement range of the axis, and the red dot is the end point of the movement range of the slave axis.

The synchronization area refers to the area where the spindle and the slave axis are at the same speed. In this area, the spindle moving speed is the same as the slave axis moving speed, and some cutting operation can be performed in this area.

Material length refers to the length of the material to be cut.

One circle length of the spindle axis refers to the circumference of the spindle axis transmission wheel, and its value is equal to the diameter of spindle axis transmission wheel *π.

6.5.1.2 Slave axis movement mode

Long and short material judgment: material length > slave axis movement range × (2+ return coefficient / 50) - synchronous length is considered to be long material, otherwise it is judged as short material.

The movement sequence of salve axis varies depending on the material length. The specific slave axis movement sequence is as follows:

- Long material movement order:

First point: The first point is the green point, which is the starting dot of the synchronization area.

Second point: the second point is the yellow dot, which is the end dot of the synchronization area, and the slave axis moves from the green dot to the yellow dot at a constant speed in the synchronization zone.

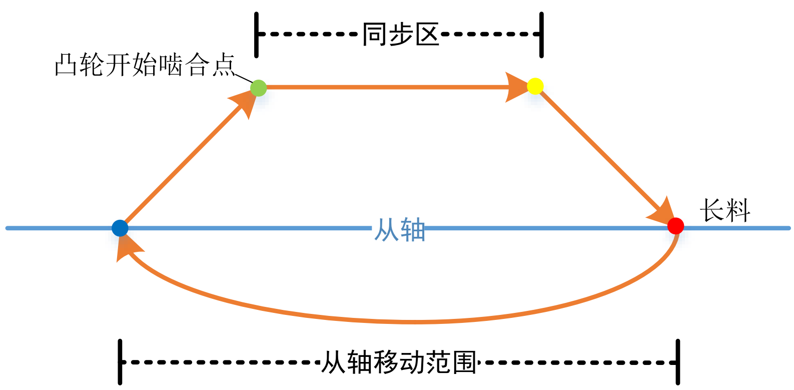
Third point: the third point is the red dot, which is the end dot of the movement range of slave axis, and the slave axis gradually decelerates from the second dot to the third dot.

Fourth point: the fourth point is the blue dot, which is the starting dot of the movement range of slave axis, and the slave axis returns from the third dot to the fourth dot and stops the waiting dot.

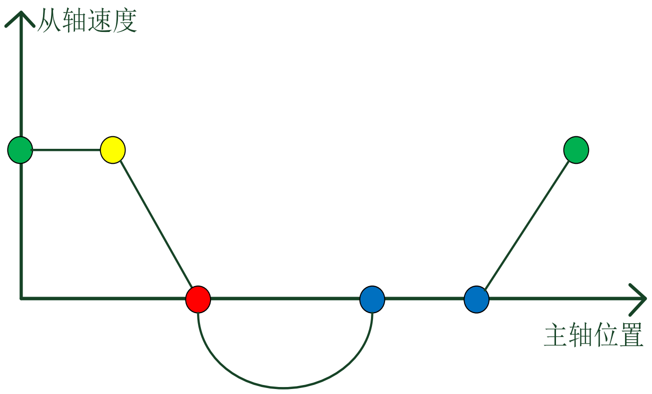
Fifth point: The fifth point is the blue dot, which is the waiting time that slave axis accelerates to the start of the synchronization area.

Sixth point: The sixth point is the green dot, the same as the first point.

The schematic diagram of the long material movement is shown in the following figure:



Long material point planning



Long material spindle and slave relationship curve

- **Spindle and slave position calculation:**

Sign	Name
L	material length (mm)
L1	One circle lead of slave axis (mm)
L2	Slave axis movement range (mm)
P1	One circle pulse number of slave axis
D	Synchronization zone(mm)
L3	One circle length of spindle axis(mm)
P2	One circle pulse number of spindle axis
K	Return coefficient
R	Angular velocity ratio

Angular velocity ratio $\mathbf{R}=\frac{(\boldsymbol{L3}\times\boldsymbol{P1})}{(\boldsymbol{P2}\times\boldsymbol{L1})}$

Keypoints	Position (number of pulses)	Calculation formula
Point 1	Spindle axis	0
	Slave axis	0
Point 2	Spindle axis	$\frac{P2}{L3}\times D$
	Slave axis	$\frac{P2}{L3}\times D\times R$
Point 3	Spindle axis	$\frac{P2}{L3}\times L2$
	Slave axis	$\frac{L2+D}{2}\times\frac{P2}{L3}\times R$
Point 4	Spindle axis	$\left(1+\frac{K}{50}\right)\times\frac{P2}{L3}\times L2$
	Slave axis	$-\left(\frac{L2-D}{2}\times\frac{P2}{L3}\times R\right)$
	Spindle axis	$\frac{P2}{L3}\times(L-L2+D)$

Point 5	Slave axis	$-\left(\frac{L2-D}{2}\times\frac{P2}{L3}\times R\right)$
Point 6	Spindle axis	$L\times\frac{P2}{L3}$
	Slave axis	0

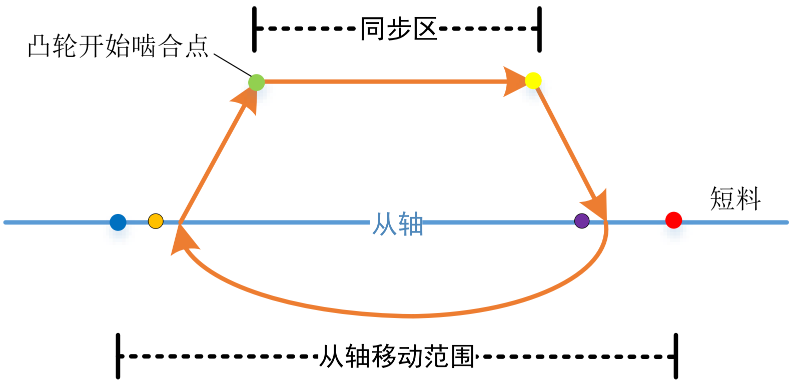
◆ **Short material movement order:**

When the material is short, if the slave axis still moves according to the five points of the long material, it may cause the slave axis to return too fast and the servo may not be able to track. So for the case of short cutting materials, we have modified the moving points.

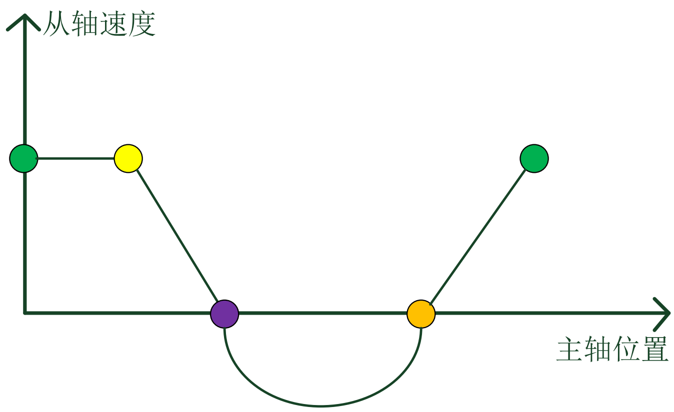
Modifys:

1) Reduce the deceleration time, that is, the slave axis decelerates from the yellow dot, and decelerates rapidly to 0, and it does not decelerate to the red dot, but to the position behind the red dot.

2) Reduce the acceleration time, that is, the slave axis does not accelerate from the blue dot, but accelerates from the position before the blue dot. This requires the slave axis does not completely return to the blue dot position, but before the blue dot when returning.



Short material point planning



Short material spindle and slave relationship curve

Note: The material length should not be too short. When the material length is too short, it may cause the return speed from the axis to be too fast, resulting in servo alarm.

◆ **Spindle and slave position calculation:**

Sign	Name
L	material length (mm)
L1	One circle lead of slave axis (mm)
L2	Slave axis movement range (mm)
P1	One circle pulse number of slave axis
D	Synchronization zone(mm)
L3	One circle length of spindle axis(mm)
P2	One circle pulse number of spindle axis
K	Return coefficient
R	Angular velocity ratio

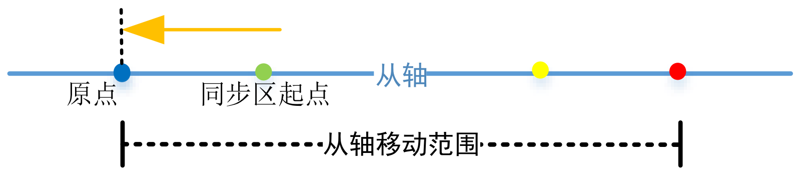
Angular velocity ratio: $\mathbf{R} = \frac{(L3\times P1)}{(P2\times L1)}$

Keypoints	Position (number of pulses)	Calculation formula
Point 1	Spindle axis	0
	Slave axis	0
Point 2	Spindle axis	$\frac{P2}{L3}\times D$
	Slave axis	$\frac{P2}{L3}\times D\times R$
Point 3	Spindle axis	$\frac{D+L2}{2}\times\frac{P2}{L3}$
	Slave axis	$\frac{3\times D+L2}{4}\times\frac{P2}{L3}\times R$
Point 4	Spindle axis	$\left(\frac{D-L2}{2}+L\right)\times\frac{P2}{L3}$
	Slave axis	$-\left(\frac{L2-D}{4}\times\frac{P2}{L3}\times R\right)$
Point 5	Spindle axis	$\frac{P2}{L3}\times L$
	Slave axis	0

6.5.1.3 Chasing operation steps

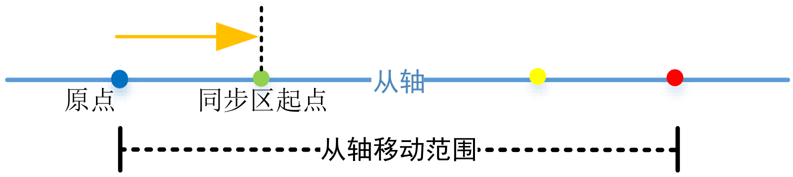
According to the actual application of chasing, there are long length and short length materials, but the starting point of the synchronization area is fixed regardless of the material length, so we set the starting point of the whole operation at the starting point of the synchronization area. The operation flow is as follows.

Step 1: Origin return. The slave axis returns to the origin position with the controller's origin return command. Here we use the blue dot as the origin, that is, control the slave axis back to the blue dot.



Step 2: Preset the slave axis. After returning to the origin, it is necessary to control the slave axis movement from to the start position of synchronization area, and the start position of the synchronization area is the starting point of the cam engagement.

Pre-set value= (slave axis movement range - synchronization area length) / 2



Note: Step 1 and Step 2 generally only need to be executed once when debugging. After the position of spindle and slave axes is saved, the cam can be engaged immediately after power-on.

Step 3: Configure the general purpose cam and write the chasing parameters through the DCAMWR command. Please refer to "2.1.4 DCAMWR Flying Shear Parameter Writing Function" for more details.

Step 4: Enable the cam. The spindle and the slave axis are engaged.

Step 5: The controller controls the spindle movement. The spindle-slave relationship operates in accordance with the set cam relationship.

6.5.1.1 DCAMWR_K10004 Command

This function is written by the DCAMWR command. The form of the command is as follows:

DCAMWR m1 m2 D n		Chasing parameter write	Command execution
m1	Cam table	Specify the CAM table to be modified	32-bit command (17step) DCAMWR continuous execution
m2	Subset selection		
D	Data	Data storage address to be modified	
n	Reserved		

- m1 is the cam table selection function. Select the modified cam table. The value range of m1 is K11~K16.
- m2 is a subset of the application function. When m2 is set to K10004, it is a special function for chasing.
- D is the storage address of the data that needs to be modified.
- Chasing special function blocks, n is a reserved bit.

For example

DCAMWR K11 K10004 D100 K0

This command indicates that a set of chasing parameters headed by D100 is written into the CAM table 1. The relevant parameters are stored in the 16 D soft components headed by D100. The parameters are as follows:

Data type	Register	Definition	Unit	Value range
Float	D100D101	One circle length of spindle axis	mm	
Unsigned integer	D102D103	One circle pulse number of spindle axis	pcs	
Float	D104D105	One circle lead of slave axis	mm	
Unsigned integer	D106D107	One circle pulse number of slave axis	pcs	
Float	D108D109	Synchronization zone length	mm	
Float	D110D111	Slave axis movement range	mm	
Unsigned integer	D112D113	Return coefficient		40-500
Float	D114D115	material length	mm	

6.5.1.2 Chasing parameters

Parameters	Function	Calculation
One circle length of spindle axis	One circle length of spindle axis	Spindle wheel diameter *π
One circle pulse number of spindle axis	One circle pulse number of spindle axis rotating one turn	Servo motor rotates one revolution pulse number / transmission ratio
One circle lead of slave axis	Lead of the screw rotating one turn	Actual depends
Synchronization zone length	Synchronization zone length	Actual depends
Slave axis movement range	Slave axis movement range	Actual depends
Return coefficient	Adjust the return rate from the axis Note 1	The recommended default is 100 (valid for long material)
material length	The length of the material that needs to be cut	Actual depends

Note 1: The return coefficient can be used to adjust the return rate of the slave axis. The value ranges from 40 to 500. The larger the value, the slower the return speed. The smaller the value, the faster the return speed will be. Therefore, when the material is short, this value can be adjusted appropriately, but it may cause the servo overspeed.

6.5.1.3 Chasing Parameters take effect

After the chasing function parameter is written, it can be selected to take effect with the next cam cycle, or it can be selected when the E-CAM is started next time. The effective mode depends on the M soft components.

Effective mode						Soft component status	
Axis1	Axis2	Axis3	Axis4	Y0	Y1	ON	OFF
SM83	SM183	SM283	SM783	SM883	SM983	Effective for the next cam cycle	Effective the next time the cam is started

When the corresponding M soft component is set to ON, it will take effect for the next cam cycle. When the corresponding M soft component is set to OFF, it will take effect when the E-CAM is started next time.

When the chasing parameters take effect, the software automatically turns the SM82 soft component to ON (this soft component is common to all axes).

For example:

The material cutting length is modified during operation, and the corresponding M soft component is set after Modify to make the Modify take effect. The operation diagram is as shown below:

As can be seen from the figure:

a) This is a schematic diagram of a long and short material switch. When the cutting length is switched from short material to long material, there is no waiting period with short material running, and there is a certain waiting time with long material running.

b) Since only the material length is switched, there is no influence on the speed of synchronous zone. After the short material is switched to the long material in the figure, only the spindle axis length changes, and speed does no change in the synchronous zone r.

c) Switching occurs at the starting point of the synchronization area to ensure accurate material length before and after switching.

6.5.2 Application examples

6.5.2.1 Basic case

A chasing case will be given below and the parameters are as follows:

- Spindle diameter: 60mm
- Spindle transmission ratio: 1:10
- Slave axis screw spacing: 20mm
- Slave axis reduction ratio: 1:1
- Slave axis movement range: 400mm
- Slave axis synchronization length: 300mm
- Cutting material length: 1300mm

The parameters’ value that needs to be set by the CAMWR command is calculated as below:

(Servo adopts SD700 servo, internally set 10000 pulses for servo motor one circle)

One circle length of spindle axis =60*π=188.4

One circle pulse number of spindle axis = 10000*10=100000

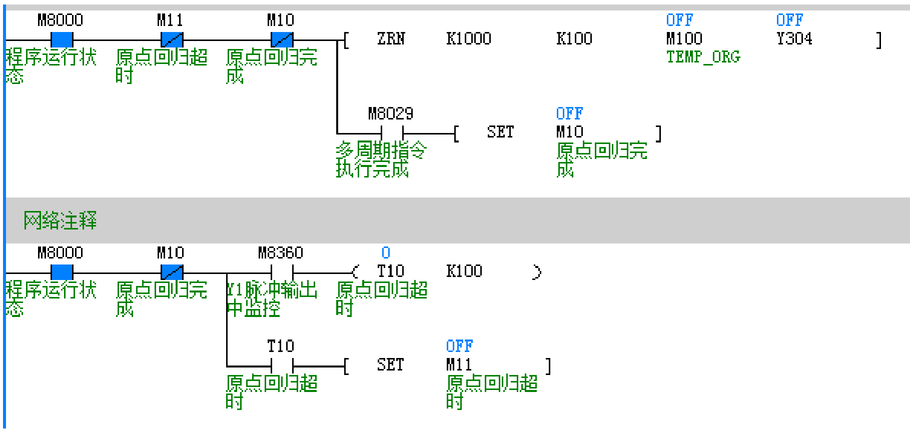
One circle pulse number of slave axis= 10000*1=10000

The return coefficient is recommended to 100

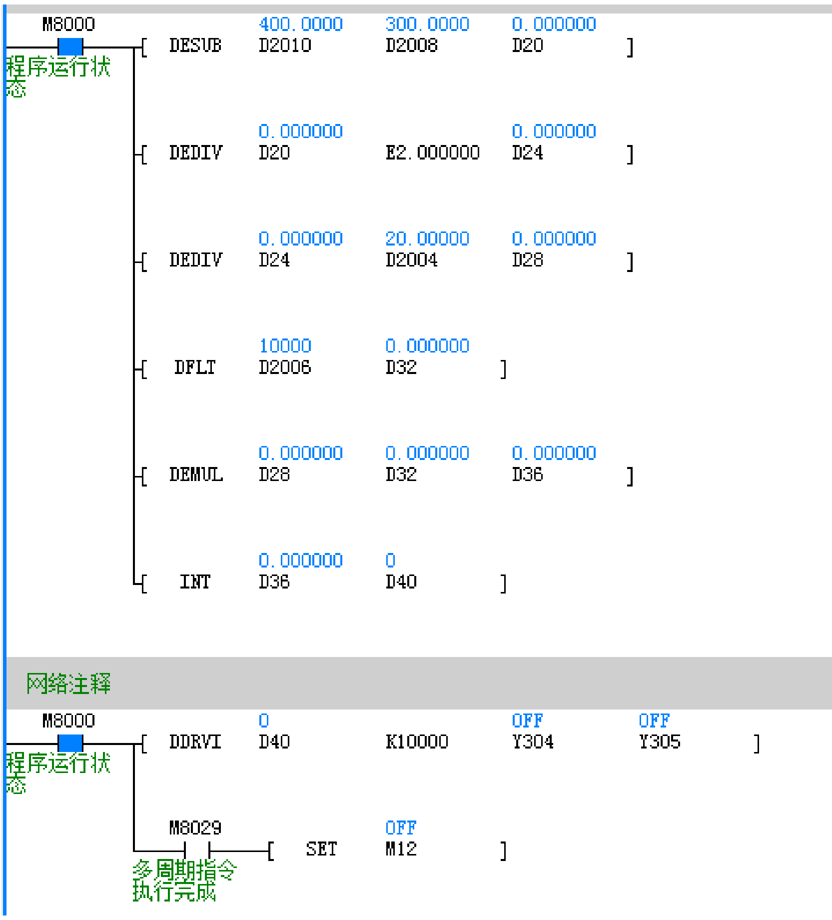
Parameters	Unit	Setting value
One circle length of spindle axis	mm	188.4
One circle pulse number of spindle axis	pcs	100000
One circle lead of slave axis	mm	20
One circle pulse number of slave axis	pcs	10000
Synchronization zone length	mm	300
Slave axis movement range	mm	400
Return coefficient		100
material length	mm	1320

Next, we will follow the steps in "2.1.3 Chasing Operation Steps" to run our chasing system step by step.

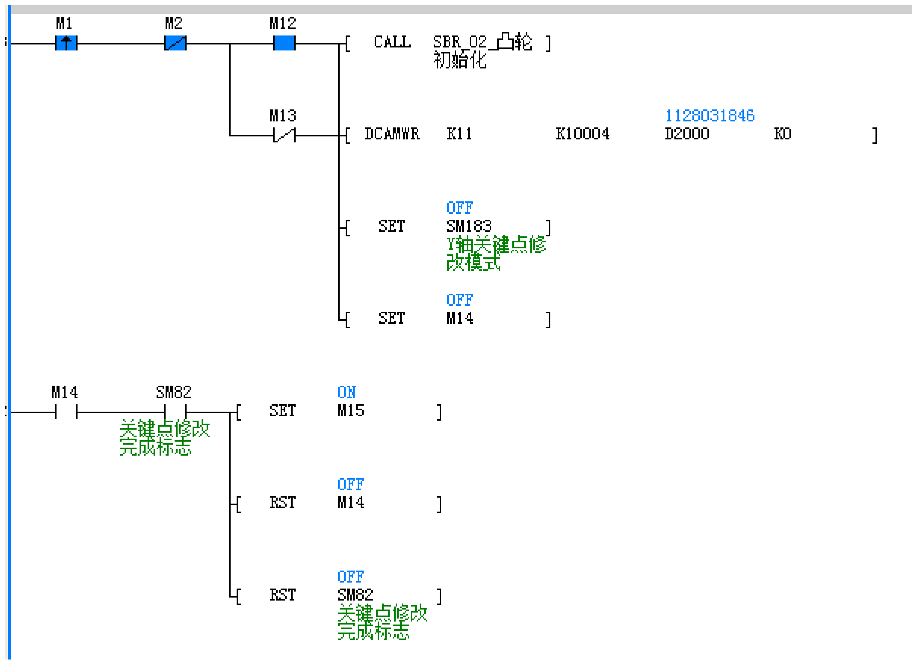
- Step 1: Control the slave axis to the origin with the ZRN command



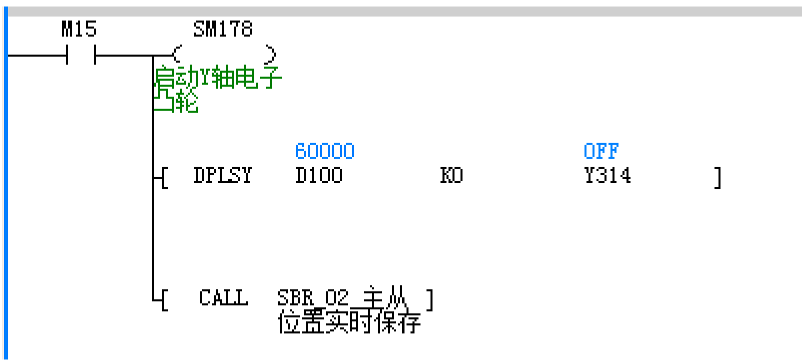
- Step 2: After returning to the origin, pre-set the slave axis .



- Step 3: E-CAM initialization, configure the parameters related to the E-CAM; write the chasing parameters through the DCAMWR command.



- Step 4: Engage the E-CAM, the controller controls the spindle movement, and the slave axis automatically moves according to the generated E-CAM curve



The overall operation results are shown in the figure:

a) The process of returning to the origin. The ZRN command is used to return to the origin.

b) The slave axis pre-set process. Pre-set value = (slave axis movement range - synchronization zone distance) / 2.

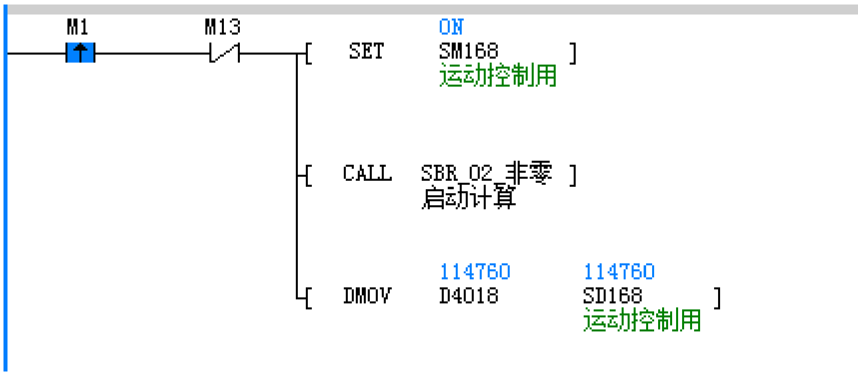
c) A chasing cycle. It can be seen from the figure that a chasing cycle starts from the beginning of the synchronization area and ends at the start of the synchronization area.

Note: Process ① and Process ② generally only need to be performed once. The spindle and slave position can be saved later, and the cam is directly engaged after power-on, so that there is no waste of material.

6.5.2.2 Power-on operation case

In the chasing process, if not finding the original point and directly engage after power-on, you can use the non-zero start function of the E-CAM to achieve. The specific implementation principles refer to the relavant chapters.

The following is based on the above example and add direct start function. The ladder diagram is as follows:



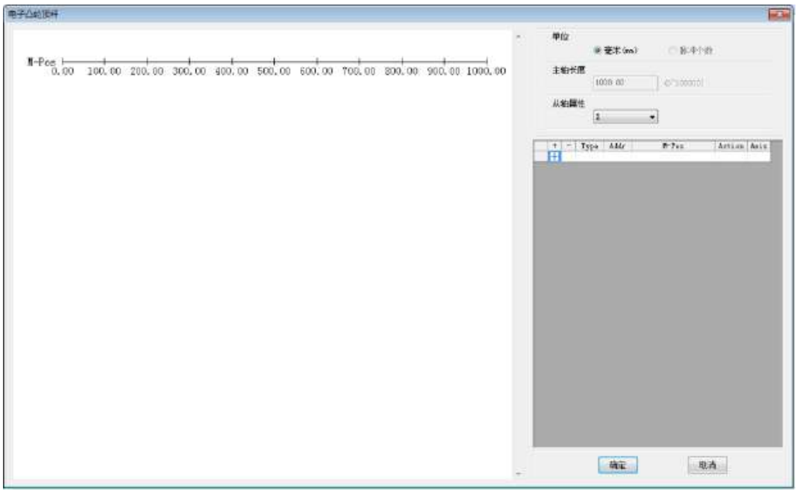
The operation results are shown below:

6.6 Special features

6.6.1 Ejector function

The ejector function can realize the cooperation of the bit component (M/Y) and the position of the electronic cam spindle. The control bit component changes with the position of the spindle. The settings of the ejector are as follows.

1. Create a new cam table and set the spindle range and unit in it. Right click ON the cam table to create a new ejector. Double click to open the electronic cam ejector editing interface:



2. Click ON the ejector data sheet to add or remove ejector data:

	+	−	Type	Addr	M-Pos	Action	Axis
			M	0			
					0.00	NA	X
					0.00	NA	X
			M	1			
					0.00	NA	X

3. Set the ejector data:

M-Pos: Set the position of the electronic cam spindle;

Type: Set the bit component type:M component and Y component;

Addr: Set the label of the component;

Action: When the spindle position is equal to the M-Pos setting value, the position component operates; NA means no action; ON means turn ON; OFF means turn OFF; INV means reversing;

6.6.2 Calculation of master-slave speed ratio

This function can calculate the speed ratio between the master and slave shafts. When the cam parameters are set, the system will calculate the maximum and minimum speed ratio of the master and slave shafts.

Users can calculate the speed ratio between the master and slave shafts through this function to verify whether the cam point setting is reasonable.

Users can also use the calculated speed ratio to limit the speed of the spindle to prevent slave shaft from over speed.

The relevant soft components of this function are shown in the table below:

	Shaft 1	Shaft 2	Shaft 3	Shaft 4	Y0	Y1	Instruction
Start calculating[1]	SM72	SM172	SM272	SM772	SM872	SM972	When the soft component is set to ON, the master-slave speed ratio is calculated. After the calculation, the system will automatically set the soft component to OFF.
Maximum speed ratio[2]	SD74	SD174	SD274	SD774	SD874	SD974	Maximum speed ratio of master and slave, 32-bit floating-point number
Minimum speed ratio[3]	SD76	SD176	SD276	SD776	SD876	SD976	Minimum speed ratio of master and slave, 32-bit floating-point number

Note [1]：When the SM soft component for calculating the master-slave ratio is set to ON, the system will start to calculate the master-slave maximum / minimum ratio. After the calculation, the system will set the soft component to OFF.

Note [2]：The calculated value is the maximum / minimum speed ratio. Note that the data type of the soft component is floating-point.

Note [3]：When the speed ratio curve is asymmetric, the calculation of speed ratio is not accurate。

For example:

① If the maximum speed ratio is 10.0 after the master-slave ratio calculation function is turned ON, and if the maximum pulse frequency that can be received from the shaft is 100kHz, it is recommended that the frequency of the spindle should not exceed 10kHz , otherwise it may cause excessive speed of the slave shaft

② If the master-slave ratio calculation function is turned ON, the calculated minimum speed ratio of the master-slave is - 1.0, but if the reverse direction of the slave shaft is not allowed in the processing, it means that the set cam point is unreasonable and the cam point needs to be adjusted.

6.6.3 Monitoring of master section position

Monitor the position of the spindle in a cam cycle.

It shows the range [1, len] of the current spindle, where 'len' is the length of spindle.

	Shaft 1	Shaft 2	Shaft 3	Shaft 4	Y0	Y1	Instruction
Monitoring of master section position	SD85, SD86	SD185, SD186	SD285, SD286	SD785, SD786	SD885, SD886	SD985, SD986	Unit：pls 32 bits integer

6.6.4 Monitoring of accumulated pulse of spindle

When the cam is running, the number of accumulated pulses received by the spindle can be obtained through the SD component.

	Shaft 1	Shaft 2	Shaft 3	Shaft 4	Y0	Y1	Instruction
Monitoring of accumulated pulse of spindle	SD48, SD49	SD148, SD149	SD248, SD249	SD748, SD749	SD848, SD849	SD948, SD949	Unit：pls 32 bits integer

Note 1: The pulse is not included in case of delay start.

Note 2: Internal 64 bits counting, take the lower 32 bits, such as count from 0x7fffffff up to 0x80000000, count from 0x80000000 down to 0x00000000.

6.6.5 Setting the direction of electronic cam

You can set the operation mode of the slave shaft in different spindle directions. The positive operation along the cam curve means that the slave shaft follows the spindle signal and runs along the set cam curve. The reverse operation along the cam curve means that the slave shaft follows the spindle signal and runs in the opposite direction of the cam curve.

When the spindle runs for a distance in one direction and then reverses, slave shaft runs reversely along the set curve, that is to say the end of the cam curve is taken as the starting of the slave shaft. Until the spindle returns to the cam engagement position, if the spindle continues to run in the direction at this time, the secondary shaft will not act and a fault will be reported.

	Shaft1	Shaft2	Shaft3	Shaft4	Y0	Y1	Instruction
Setting the direction of cam SD	SD84	SD184	SD284	SD784	SD884	SD984	0：When the spindle direction is positive, it runs along the cam curve positively. 1：When the spindle direction is negative, it runs along the cam curve positively. 2：When the spindle direction is positive, it runs reversely along the cam curve. 3：When the spindle direction is negative, it runs reversely along the cam curve

Note: if the servo parameters are not adjusted properly in the position of the electronic cam engagement, the feedback pulse will fluctuate and the fault will be reported.

6.6.6 Encoder frequency monitoring

Monitor encoder frequency. (Unit pls)

	Shaft1	Shaft2	Shaft3	Shaft4	Y0	Y1	Instruction
Encoder frequency monitoring	SD46, SD47	SD146, SD147	SD246, SD247	SD746, SD747	SD846, SD847	SD946, SD947	Unit：pls 32 bits integer

6.6.7 Calculation of spindle position (Developing)

1.Instruction form

Calculate the spindle position according to the position of the electronic cam spindle.

DCAMSP S1 S2 D		Calculation of spindle position	Instruction execution
S1	Slave shaft position		32-bit instruction（13setp） Continuous execution
S2	Cam slave shaft number +1		
D	D+0：Total points D+2：Point 1: Interval (interval starts from 1) D+4：Point 1: Spindle position D+6：Point 2: interval D+8：Point 2: Spindle position		

2. Operand

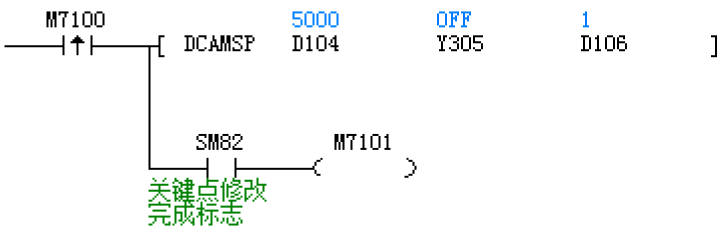
Operand	Bit soft component							Word soft component															
	System · user							System · user					Bits allocation					Indexing			Constant number		real number
s1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
s2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: Component with gray shading indicate that they are not supported.

Function instruction

- The cam must be engaged to obtain a valid slave shaft position.
- When the function is enabled, SM82 needs to be cleared and will be automatically set after calculating.
- Try to use edge triggering to prevent multiple calculations.
- Difference from the previous calculation: Cam shaft number + 1.

Example：



	元件名称	数据类型	显示格式	当前值
1	D106	32位整数	十进制	1
2	D108	32位整数	十进制	1
3	D110	32位整数	十进制	5000

6.6.8 Calculation of slave shaft position

1.Instruction form

Calculate the position of the slave shaft according to the position of the electronic cam spindle.

DCAMSP S1 S2 D		Calculation of slave shaft position	Instruction execution
S1	Spindle position		32-bit instruction（13setp）Continuous execution
S2	Cam slave shaft number		
D	Slave shaft position		

2.Operands

Operands	Bit soft component							Word component															
	System · user							System · user					Bits allocation					Indexing			Constant number		real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E
D	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V	Z	Modify	K	H	E

Note: Component with gray shading indicate that they are not supported.

Function and action instruction:

- S1: The unit of the camshaft position is pulses. The setting value of [S1] should be within the spindle range of one cycle of the cam
- S2: y304, y310 and y314 respectively correspond to shaft 1, shaft 2, shaft 3 and shaft 4.
- D: Calculate the slave shaft position corresponding to the cam table used in [S2] based ON the spindle position set in [S1],

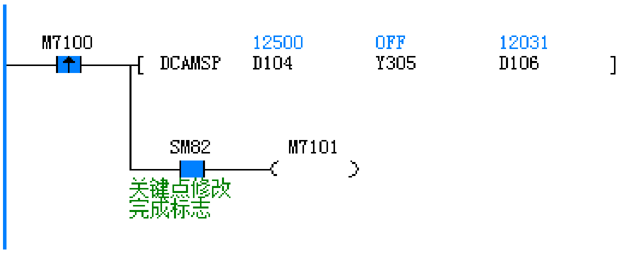
Note:

a. The cam must be engaged to obtain a valid slave shaft position.

b. Turn ON the function, you need to clear SM82, SM82 automatically set after calculating.

c. Try to use edge triggering to prevent multiple calculations.

example:



6.6.9 Probe

1. Function instruction

The function of probe is realized by hardware. It has the characteristics of real-time data locking. In many cases, it can significantly improve the accuracy of the system, and the calculation with phase can accurately calculate the deviation.

2.Operation register

IN/OUT	Instruction	Register	Length	Instruction
IN	Enable	M8150	BOOL	ON Enable；OFF Disable
	Port selection	D8150:bit[2:0]	3 bit	0：X0;1：X1 2：EZ[0];3：EZ [1] 4：EZ[2];5：EZ[3]
	Edge selection	D8150:Bit[3]	1bit	0: Rising edge 1:Falling edge
	Shaft setting	D8150:bit[6:4]	3bit	0~5：Corresponding shaft 1~6
	Logic and encoder selection	D8150:bit[7]	1bit	0：Logic position D8356 1：Encoder position SD56
OUT	Latch position	D8151：:32bit	32bit	Corresponding data
	Latch status	M8151	BOOL	ON:Latch；OFF NO latch

3.Usage method

A. Setting mode. (after enabling, the Modify mode is invalid)

B. Enable.

C.Read the latched status in real time. If it is ON, the latched signal will be read. (read continuously)

D. Disable.

Note 1: V5 has three probes, as shown in the register list below

Note 2: The status of the latch will also be OFF when disabled.

Note 3: If the ENABLE FUNCTION is always ON, the function of triggering probe will be kept. The latch position will be updated continuously and the status will be ON.

Note 4: For the corresponding axis, 20 needs to be added, for example, axis 1: d8356, axis 2: d8376

Note 5: Compared with D8340 (periodic refresh), D8356 is real-time data sent by hardware, which will change every time a pulse is sent.

The corresponding registers of the three probes are shown in the table below

NO	Instruction	Read/Write	Probe 1	Probe 2	Probe 3
1	Enable	W	M8150	M8153	M8156
2	Pattern	W	D8150	D8153	D8156
3	Latch position	R	D8151，D8152	D8154，D8155	D8157，D8158
4	Latch phase value	R	D8162，D8163	D8164，D8165	D8166，D8167
5	Latch status	R	M8151	M8154	M8157

Note: Latch phase value =Current latch position -Last lock position (The first latch does not practice)

6.6.10 Motion superposition

MOVIMPOS Motion superposition

1.Instruction form

Motion pulse superposition.

MOVIMPOS S1 S2 S3 D			Motion superposition	Instruction execution	
S1	Pulses	Set the total number of stacked pulses		16-bit instruction (9step) MOVIMPOS continuous execution	32-bit instruction (17step) DMOVIMPOS continuous execution
S2	Frequency	Set stacked pulse frequency			
S3	Pattern	Superposition mode			
D	Output port	Set the output port of superimposed pulse			

2.Operands

Operands	Bit soft component							Word soft component														
	System · user							System · user				Bits allocation					Indexing		Constant number		real number
S1	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	
S2	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	
S3	X	Y	M	T	C	S	SM	D	R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	
D	X	Y	M	T	C	S	SM		R	T	C	SD	KnX	KnY	KnM	KnS	KnSM	V,Z	Modify	K	H	

Note: Soft components with gray shading indicate that they can be supported.

Function Instruction

1. ON the basis of the current pulse output, stack a certain pulse according to the set stacking frequency.

2. S1: Total number of superposed pulses. 16 bits: - 32767 ~ 32767. 32 bits: - 2147483648 ~ 2147483647. Symbol indicates superposed direction. Positive time indicates pulse increase, Negative time indicates pulse decrease.

Note: Pulse increase or decrease can only be carried out in one direction. Pulse decrease can not change the direction of current pulse output instruction, and the maximum reduction is that the output pulse frequency is 0;

3. S2: Superimposed pulse frequency. 16 bits: 1-65535, 32 bits: 1-3m. Unit: Hz. The actual superimposed frequency may be different from the set frequency, and the actual maximum superimposed is 1K /ms. The output pulse frequency of the actual port is: positioning instruction frequency ± superposition frequency;

4. S3: Stacking mode. 0: acceleration / deceleration mode. Others: reserved;

5.D: Pulse output port. axis 1, axis 2, axis 3, axis 4, Y0, Y1.

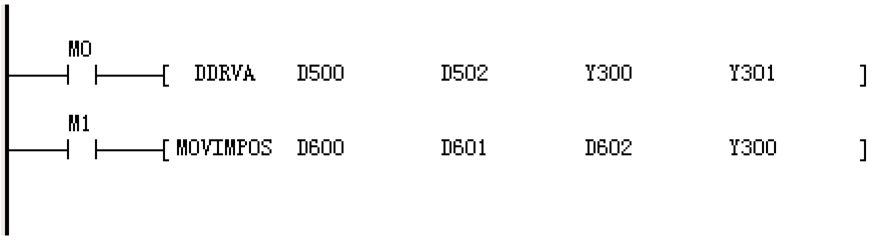
Important points:

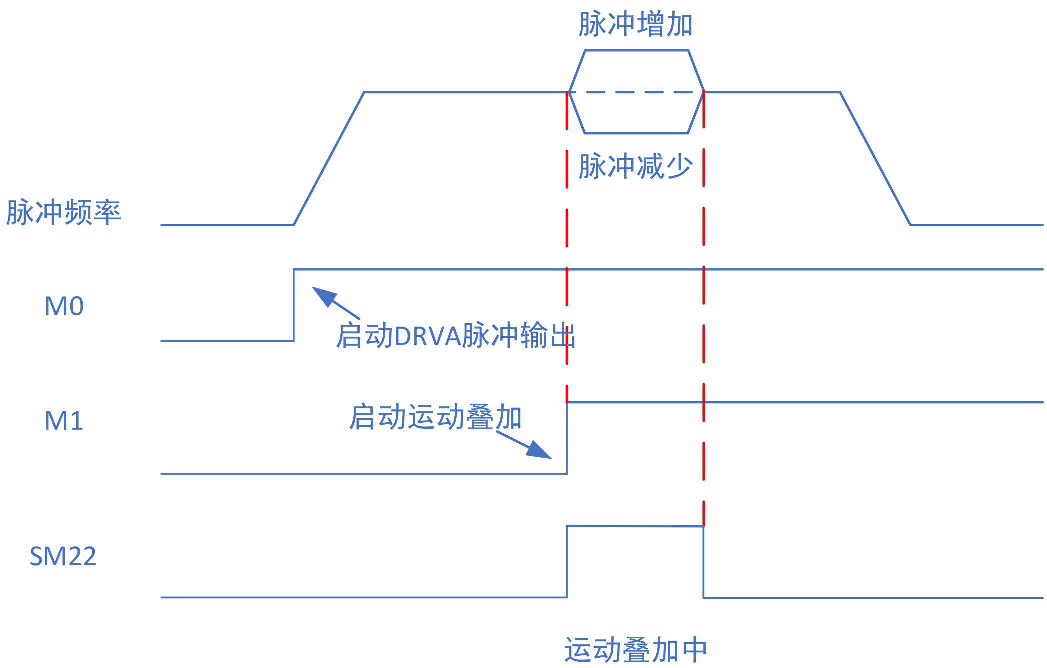
The special flag bits used can monitor the motion superposition state:

Shaft1	Shaft2	Shaft3	Shaft4	Y0	Y1	Function	
SM22	SM122	SM222	SM722	SM822	SM822	Motion superposition state	ON: Motion stack in progress
							OFF：Motion stack complete

After the motion superposition is started, the pulse output port changes the current output frequency according to the acceleration of 1kHz / ms based on the current output frequency, and increases or decreases the superposition frequency based on the current output frequency, so that increase the total number of output pulses or decrease the number of set pulses.

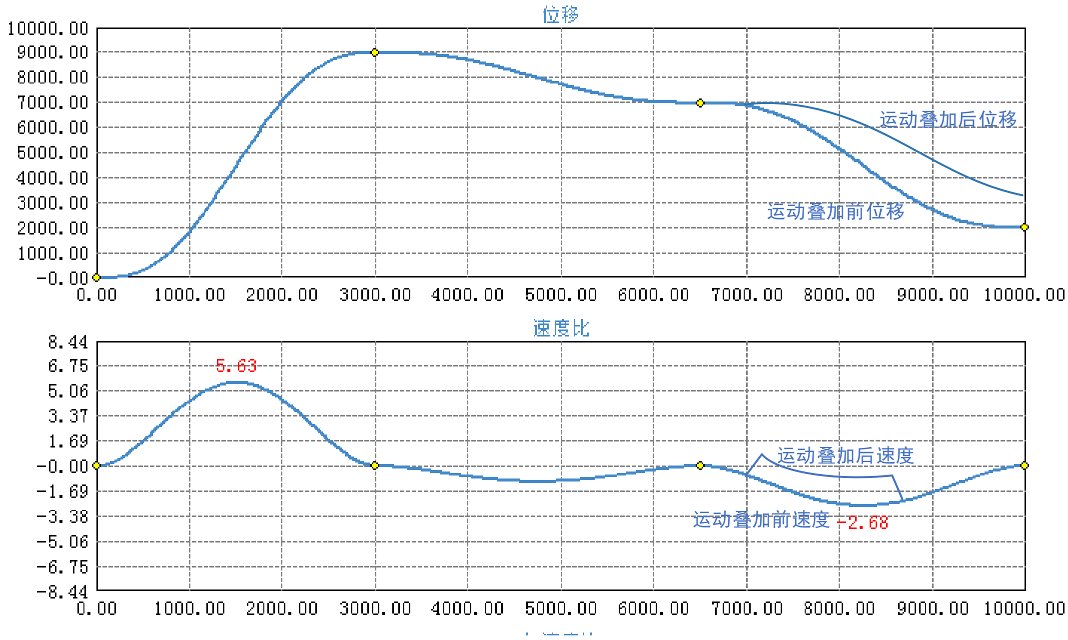
【For example】





After M0 starts the output pulse of positioning instruction DRVA, set M1 to start the motion superposition function, which can change the current pulse frequency and increase or decrease the number of positioning pulses.

In the application of electronic cam tracking, it can detect the time or distance between the two color marks. By detecting the time or distance between the two color marks, the motion superposition function can be started, and then the feeding axis can be compensated in real time to improve the tracking accuracy.



6.6.11 Control cycle

The control cycle is to calculate a data at a fixed time. The default is 500(unit *us)*. The value can affect the response time of the system, the parameter is SD87(16 bits).

The setting range is 125, 250, 500, 1000, corresponding to 125*us*, 250*us*, 500*us*, 1*ms* respectively. Among them, the setting can refer to the current cost and the maximum cost to adjust. It is recommended to set no more than 70%. For example, if the current SD88=100, then 100/0.7=142, then the set data is set to 250, 500, 1000.

No.	Parameter variable	Instruction	BIT
1	SD87	Set control cycle (R / W)	16bits
2	SD88	Real time sampling task cost(R)	16bits
3	SD89	Max sampling task cost(R)	16bits

Note: For occasions with high response requirements, it is recommended to set as small as possible.

6.7 list of electronic cam related soft components

SD soft components

Name	Shaft1	Shaft2	Shaft3	Shaft4	Y0	Y1	Meaning
	D8340 D8341	D8360 D8361	D8380 D8381	D8400 D8401	D8420 D8421	D8440 D8441	Internal logical position
	SD61	SD161	SD261	SD761	SD861	SD961	Pulse output setting
	SD56 SD57	SD156 SD157	SD256 SD257	SD756 SD757	SD856 SD857	SD956 SD957	External encoder position
	SD68 SD69	SD168 SD169	SD268 SD269	SD768 SD769	SD868 SD869	SD968 SD969	Non zero start spindle position
	SD70	SD170	SD270	SD770	SD870	SD970	Cam chart selection
	SD71	SD171	SD271	SD771	SD871	SD971	Cam spindle selection
	SD72	SD172	SD272	SD772	SD872	SD972	Number of cam set cycles
	SD73	SD173	SD273	SD773	SD873	SD973	Scale ratio of cam slave shaft
	SD74 SD75	SD174 SD175	SD274 SD275	SD774 SD775	SD874 SD875	SD974 SD975	Maximum speed ratio of cam (floating point number)
	SD76 SD77	SD176 SD177	SD276 SD277	SD776 SD777	SD876 SD877	SD976 SD977	Minimum speed ratio of cam (floating point number)
	SD78 SD79	SD178 SD179	SD278 SD279	SD778 SD779	SD878 SD879	SD978 SD979	Number of delayed start pulses
	SD80	SD180	SD280	SD780	SD880	SD980	Reserve
	SD81 SD82	SD181 SD182	SD281 SD282	SD781 SD782	SD881 SD882	SD981 SD982	Number of cam executed cycles
	SD83	SD183	SD283	SD783	SD883	SD983	Reserve
	SD84	SD184	SD284	SD784	SD884	SD984	Cam direction setting
	SD85 SD86	SD185 SD186	SD285 SD286	SD785 SD786	SD885 SD886	SD985 SD986	Length monitoring of spindle section of cam

SM soft components

Name	Shaft1	Shaft2	Shaft3	Shaft4	Y0	Y1	Meaning
	SM68	SM168	SM268	SM768	SM868	SM968	Cam non-zero start enable
	SM70	SM170	SM270	SM770	SM870	SM970	Cam trigger mode
	SM71	SM171	SM271	SM771	SM871	SM971	Cam spindle selection
	SM72	SM172	SM272	SM772	SM872	SM972	Cam speed ratio calculation enable
	SM73	SM173	SM273	SM773	SM873	SM973	Cam cycle mode enable
	SM74	SM174	SM274	SM774	SM874	SM974	Reserve
	SM75	SM175	SM275	SM775	SM875	SM975	Cam delay start enable
	SM76 SM77	SM176 SM177	SM276 SM277	SM776 SM777	SM876 SM877	SM976 SM977	Reserve
	SM78	SM178	SM278	SM778	SM878	SM978	Cam enable
	SM79	SM179	SM279	SM779	SM879	SM979	Cam cycle complete

	SM80	SM180	SM280	SM780	SM880	SM980	Cam stop status monitoring
	SM81	SM181	SM281	SM781	SM881	SM981	Cam stop mode
	SM82	SM182	SM282	SM782	SM882	SM982	Key Modify completion flag
	SM83	SM183	SM283	SM783	SM883	SM983	Cam key Modify mode
	SM84 SM88	SM184 SM188	SM284 SM288	SM784 SM788	SM884 SM888	SM984 SM988	Reserve
	SM89	SM189	SM289	SM789	SM889	SM989	Cam completion mark

Chapter 7 Interrupt

7.1 Overview

7.1.1 Overview

The interrupt function is not affected by the operation cycle of the main program. And you can take the interrupt function as the trigger signal to execute the function of the interrupt program (interrupt subprogram) immediately.

In general sequence control program processing, the delay and time deviation caused by operation cycle affect the mechanical action, which can be improved.

7.1.2 Interrupt type

		Overview	
I	Interrupt	External interrupt	X0-X2 input interrupt, No. I00 □, I10 □, I20 □, 3 points (□ means: 0 falling edge interrupt, 1 rising edge interrupt). After setting on the interrupt disable flag bit register, the corresponding input interrupt is disabled
		Timing interrupt	I6 □, i7 □, I8 □, 3 points (□□ = 1 ~ 99, time base = 1ms)
		Pulse completion interrupt	I502 ~ i504, 3 points

Host computer		V5 Controller
I001	XShaftPG0Rising edge	X0Rising edge
I101	YShaftPG0Rising edge	X1Rising edge
I201	ZShaftPG0Rising edge	X2Rising edge
I000	XShaftPG0 Falling edge	X0 Falling edge
I100	YShaftPG0 Falling edge	X1 Falling edge
I200	ZShaftPG0 Falling edge	X2 Falling edge
I6□□	Timer interrupt 0	Timer interrupt 0
I7□□	Timer interrupt 1	Timer interrupt 1
I8□□	Timer interrupt 2	Timer interrupt 2
I502	XShaft Output complete interrupt	Axis1 Output complete interrupt
I503	YShaft Output complete interrupt	Axis2 Output complete interrupt
I504	ZShaft Output complete interrupt	Axis3 Output complete interrupt

7.2 External interruption

7.2.1 Overview

Use the input signals from x000 to X002 to execute the interrupt subprogram.

Where x0, X1 are high-speed inputs and X2 are low-speed inputs.

Because the external input signal can be processed without the influence of the operation period of the PLC, it is suitable for performing high-speed control and acquiring short-time pulse.

7.2.2 External interrupt type

Input number	Interrupt number		No interruption
	Rising edge Interrupt	Falling edge Interrupt	
X0	I001	I000	M8050
X1	I101	I100	M8051
X2	I201	I200	M8052

Note: When M8050 ~ M8052 is ON, the interrupt events corresponding to their input numbers are prohibited.

Important points for external interruption:

Reuse of forbidden input: the number of input relay used as interrupt pointer shall not be repeated with application instructions such as "high speed counter", "pulse capture function", "pulse density" which use the same input range.

About the automatic adjustment of the input filter: after the input interrupt pointer I □ 0 □ is specified, the input filter of the input relay will be automatically changed to high-speed reading. Therefore, it is not necessary to use the REFF instruction and the special data register D8020 (adjustment of the input filter) to change the adjustment of the filter.

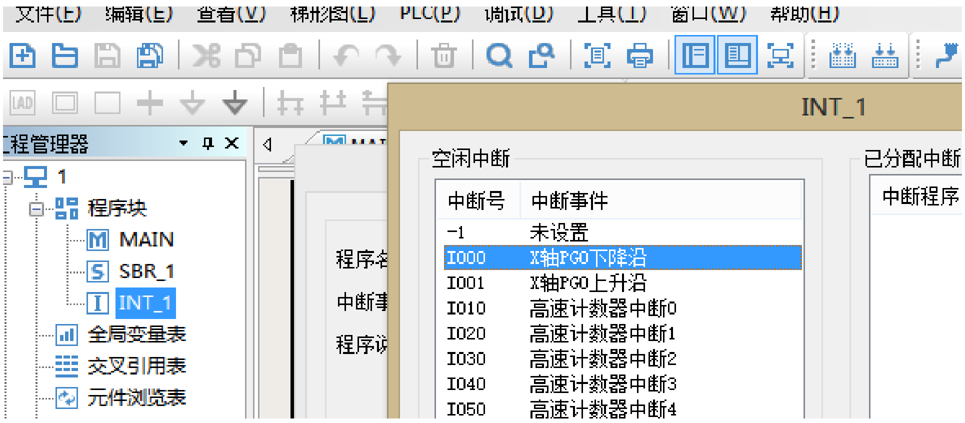
In addition, the input filter of the input relay not used as the input interrupt pointer operates in 10ms (initial value).

Reuse of pointer number: the rising edge interrupt and falling edge interrupt of the same input, such as I001 and I000, cannot be written at the same time.

7.2.3 Examples

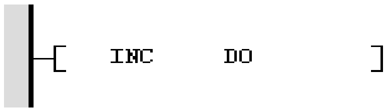
Program for counting external input interrupt X0 falling edge

(1) Establish the falling edge interrupt subroutine of X0,---I000

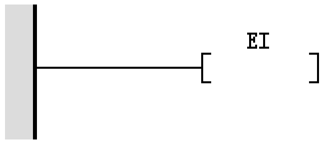


(2) interrupt subprogram content

When X0 has a falling edge, enter the interrupt subroutine D0 plus 1



(3) enable the corresponding x0 interrupt permission in the main program



7.3 Timer interrupt

7.3.1 Overview

The interrupt program is executed every 1ms ~ 99ms, which is not affected by the operation cycle of PLC.

It is suitable for high-speed processing of specific programs when the operation cycle of the main program is long, or when the program needs to be executed at a certain interval of sequential operation time.

7.3.2 Timer interrupt type

Pointer number	Interrupt period	No interruption
I6□□	In □ □ of the pointer name, enter an integer from 1 to 99. For example: i650 = timer interrupt every 50ms	M8056
I7□□		M8057
I8□□		M8058

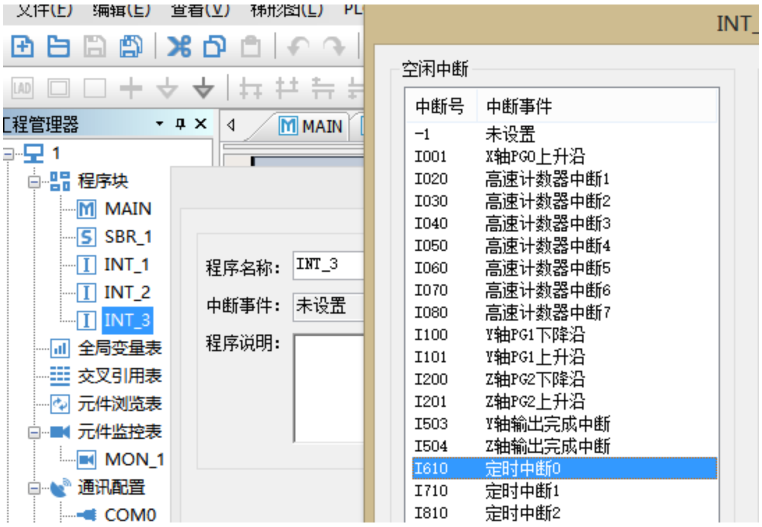
When M8056 ~ M8057 are ON, the interrupt events corresponding to their respective input numbers are prohibited;

Pointer numbers (I6, i7, I8) cannot be reused.

7.3.3 Examples

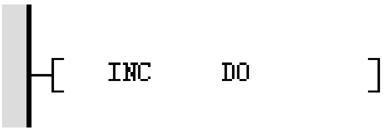
Add 1 to data d0 every 50ms

(1) Establish 50ms timing interrupt subroutine, No. i650.

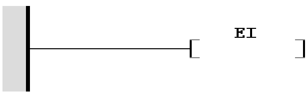


(2) Interrupt subprogram content

Enter timer interrupt every 50ms, and enter interrupt subroutine D0 plus 1



(3) EI of main program interruption



7.4 Pulse completion interrupt

7.4.1 Overview

When the special bits M8352, M8372 and M8392 (corresponding to axis1, Axis2 and axis3 respectively) are ON, pulse output complete interrupt can be realized in PISY, PLSR, DRVA, DRVI and other positioning instructions.

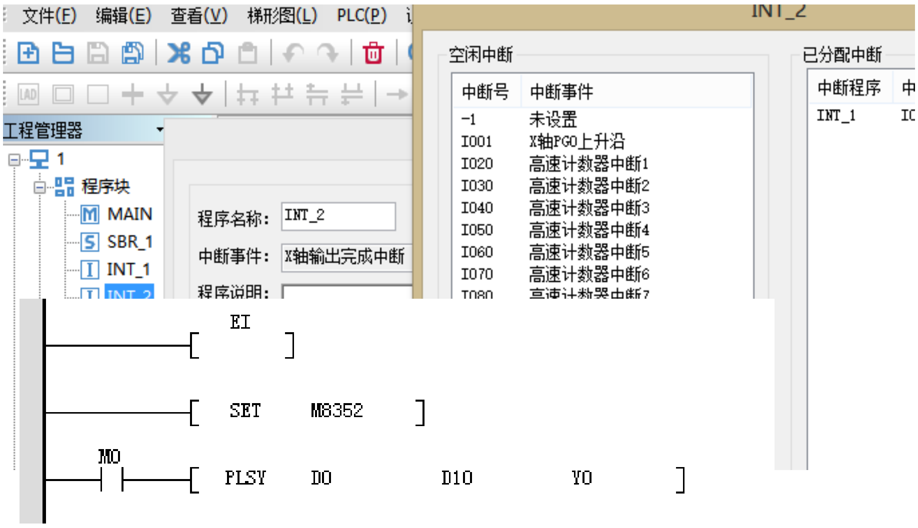
7.4.2 Pulse completion interrupt type

The pulse completion interrupt corresponding relationship is as follows:

	Interrupt object	Correspondence with V5
Axis1	Axis1 Output complete interrupt	Axis1 Pulse output interrupt I502
Axis2	Axis2 Output complete interrupt	Axis2 Pulse output interrupt I503
Axis3	Axis3 Output complete interrupt	Axis3 Pulse output interrupt I504

7.4.3 Example

(1) Set up Y0 high speed output port pulse completion interrupt, No. i502



(3) pulse completion interrupt subroutine

Chapter 8 communication

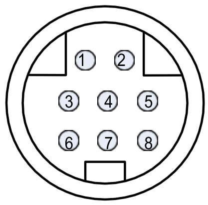
8.1 Outline

V5 motion control module, with CAN communication supports CANopen protocol and Modbus protocol.

It contains three independent physical serial communication ports, COM0, COM1 and COM 2. COM0 has programming and monitoring functions. COM1 includes RS485 communication functions (which are completely defined by users) and CAN communication functions. COM 2 is the network port and supports CAN / RS485 (232) communication.

8.2 Introduction

The COM0 hardware is the standard RS422, and the interface terminal is the 8-hole mouse head



base. The interface is defined as follows:

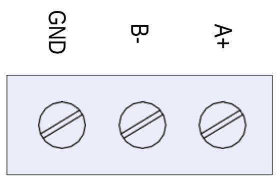
Pin number	Signal	Description
1	RXD-	Negative reception
2	RXD+	Reception is positive
3	GND	Ground wire, no electrical connection for 9 and 10
4	TXD-/RXD-	Negative external transmission ;negative reception(if it is RS485)
5	+5V	External power supply + 5V, same as internal logic + 5V
6	CCS	Communication direction control line, high level indicates transmitting, low level indicates receiving, The PLC controls whether the 4-pin and 7-pin receive or send. when RS485 is used. In case of RS422, 4-pin and 7-pin are always sending
7	TXD+/RXD+	Positive for external transmission and positive for reception if RS485
8	NC	Empty feet

There are two ways to connect V5 with computer or touch screen through COM0 port.

①Mode 1: RS422 ON V5 side and USB ON computer side. The computer is connected to the program download port of COM0 through a special USB download cable.

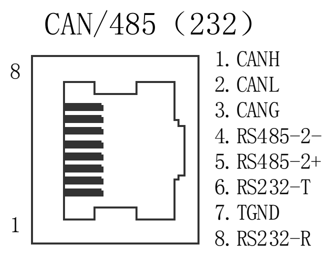
②Mode 2: RS422 ON V5 side, RS232 ON computer side. The computer is connected to the program download port of COM0 through a special serial port download cable.

COM1 is RS485, interface is terminal block, interface definition:



COM1 and other equipment communication connection mode. Through the terminal block, the user can wire on the spot.

Com2 is can / RS485 (232), and the interface is defined as follows:



RS232-R、RS232-T	RS232 communication signal (cannot be used together with RS485-2)
TGND	RS232 / RRS485-2 reference ground
RS485-2+、RS485-2-	The second group of RS485 communication signal lines
CANH、CANL、TGND	The reference level of CAN communication line is TGND. When using multiple sets, please make sure that each TGND is connected with each other

8.3 Communication protocol setup instruction

8.3.1 COM0 protocol configuration

Right click COM0 in "communication configuration" in project manager, and open it;

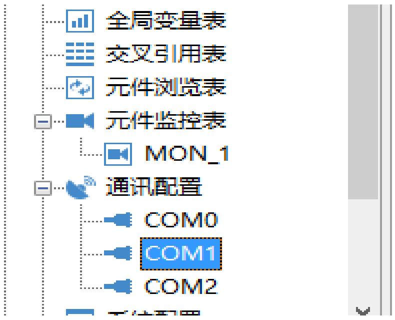
COM0's communication parameters are configured as follows:

COM0 protocol: Download / HMI monitoring protocol by default

COM0 protocol	D8116	Half duplex / full duplex mode	COM0Communication format
Download / HMI monitoring protocol	Not 02h	Unsupported	Fixed
MODBUS-RTU slave station	02h	Half duplex	Configured by COM0, viewed by D8110

COM1 Protocol configuration

COM1 Configuration diagram：



Protocol switching condition: COM1 can switch the protocol when the system is powered on or when the user program is in shutdown state, and the protocol will not be changed in the operation state.

Note: when COM1 is used as master station, M8260 flag bit should be reset.

M8260 = off, COM1 is effective as master station.

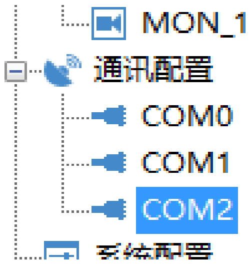
M8260 = ON, COM2 is effective as master station.

COM1 Protocol：

COM1 Protocol	D8126	Half duplex / full duplex mode	COM1Communication format
MODBUS-RTU /QLINK slave station	02h	Half duplex	Configured by COM1, viewed by D8120
MODBUS-RTU Master station	20h	Half duplex	

8.3.2 COM2 protocol configuration

COM2 configuration diagram:



Protocol switching condition:COM2 can switch the protocol when the system is powered on or when the user program is in shutdown state, and the protocol will not be changed in the operation state.

Note: when com2 is used as master station, m8260 mark position bit is required.

M8260 = off, COM1 is effective as master station

M8260 = ON, com2 is effective as master station.

Com2 protocol:

COM2 protocol	D8266	Half duplex / full duplex mode	COM2Communication format
MODBUS-RTU /QLINK slave station	02h	Half duplex	Configured by COM2, viewed by D8260
MODBUSRTUMaster station	20h	Half duplex	

8.3.3 Serial port communication format

After completing the communication configuration, the system will automatically generate the communication format, which is defined as follows:

Port	Corresponding D component
COM0	D8110
COM1	D8120
COM2	D8260

Comparison table of protocol and communication format

Protocol name	Baud rate (bit7-bit4)	Stop bit (bit3)	Check bit (bit2-bit1）	Data bit (bit0）
MODBUS-RTU Slave station MODBUS-RTU Master station	0011b:300Bits/s 0100b:600Bits/s 0101b:1200Bits/s 0110b:2400Bits/s0111b:4800Bits/s 1000b:9600Bits/s 1001b:19200Bits/s 1010b:38400Bits/s 1011b:57600Bits/s 1100b:115200Bits/s	0:1Bits 1:2Bits	00b：No check (N) 01b：Odd parity checks (O) 11b: even check	0b:7Bits 1b:8Bits Note: MODBUS-RTU slave protocol and master station only support 8-bit data bits, otherwise communication error will be caused

8.3.4 Serial port communication format soft component list

COM0:

M8110	Retain	D8110	Communication format
M8111	Retain	D8111	Communication station number
M8112	Modbus- Communication execution status	D8112	Download, HMI monitoring protocol -Communication format
M8113	Modbus- Communication error flag	D8113	Retain
M8114	Retain	D8114	Retain
M8115	Retain	D8115	Retain
M8116	Retain	D8116	Communication protocol
M8117	Retain	D8117	Retain
M8118	Retain	D8118	Modbus- Communication error station number
M8119	Timeout judgement	D8119	Communication timeout

COM1：

M8120	Retain	D8120	Communication format
M8121	Retain	D8121	Communication station number
M8122	Modbus- Communication execution status RS Instruction - send flag	D8122	Retain
M8123	Retain	D8123	Retain
M8124	Retain	D8124	Retain
M8125	Retain	D8125	Retain
M8126	Retain	D8126	Communication protocol
M8127	Retain	D8127	Retain
M8128	Retain	D8128	Retain
M8129	Retain	D8129	Communication timeout

COM2:

M8260	Switch master station M8260=0:COM1Master station M8260=1:COM2Master station	D8260	Communication format
M8261	Retain	D8261	Communication station number
M8262	Modbus- Communication execution status RS Instruction - sending flag	D8262	Retain
M8263	Retain	D8263	Retain
M8264	Retain	D8264	Retain
M8265	Retain	D8265	Retain
M8266	Retain	D8266	Communication protocol
M8267	Retain	D8267	Retain
M8268	Retain	D8268	Retain
M8269	Retain	D8269	Communication timeout

8.3.5 List of communication error codes

Parallel connection communication error M8063 (D8063) continues to run		
0000	No abnormalities	
6301	Parity error exceeds frame error	Check whether the power supply of programmable controllers of both sides is ON, whether the connection between adapter and controller or and between adapters is correct.
6302	Communication character error	
6303	Inconsistent sum of communication data	
6304	Wrong data format	
6305	Incorrect instruction	
6306	Watchdog timer overflow	
6307~6311	Nothing	
6312	Error in parallel connection character	

6313	Parallel connection and number error	
6314	Error in parallel connection format	
6330	Modbus slave address setting error	COM0 communication error Please check if COM0 communication cable is connected correctly.
6331	Data frame length error	
6332	Address error	
6333	CRC check error	
6334	Unsupported command code	
6335	Receiving timeout	
6336	Data error	
6337	Out of buffer	
6338	Framing error	
6339	Serial protocol error	
6340	Modbus slave address setting error	COM1 communication error, Please check if COM1 communication cable is connected correctly; Check whether the communication formats of the communication parties match;
6341	Data frame length error	
6342	Address error	
6343	CRC check error	
6344	Unsupported command code	
6345	Receiving timeout	
6346	Data error	
6347	Out of buffer	
6348	Framing error	
6349	Serial protocol error	

8.4 HMI Monitoring protocol

1.Hardware connection

Communication through COM0 . The connection mode is RS422, which can only be connected through the download port.

2. Software configuration

3. Agreement description

HMI monitoring protocol is the internal protocol of V5 motion controller. VCAutoDesignsoft software is used to communicate with V5. Through this protocol, VCAutoDesignsoft can erase, read and download user programs. It can implement telemetry, remote adjustment and remote control for V5. Specifically, it can monitor the status of any component in V5, change any component forcibly, and control the start and stop of V5.

8.5 MODBUS protocol

8.5.1 MODBUS protocol specification

The bottom layer of Modbus communication is RS485 signal, which can be connected by twisted pair. Therefore, the transmission distance is relatively long, up to 1000m, with good anti-interference performance and low cost. It is widely used in the communication of industrial control equipment. Now many manufacturers' inverters and controllers adopt this protocol.

There are two formats of data transmission: hex code data and ASCII code. They are respectively called MODBUS-RTU and modbus-asc protocol. The former is direct data transmission, while the latter needs to transform data into ASCII code for transmission. Therefore, the communication efficiency of MODBUS-RTU protocol is high, processing is simple and more used.

Modbus is a single active multi slave communication mode, which adopts the master-slave answer mode. Each communication is initiated by the master station first, and the slave station responds passively. Therefore, for the controlled equipment such as frequency converter, the built-in protocol is generally the slave station protocol, while for the control equipment such as PLC, the master station protocol and the slave station protocol are required.

8.5.2 MODBUS function code and data addressing

MODBUS protocol function codes 0x01, 0x03, 0x05, 0x06, 0x0f, 0x10; through these function codes, read-write coils have m, s, t, C, X (read-only), y and other variables; registers have D, t, C.

When the Modbus communication host accesses (reads or rewrites) the internal variables of the PLC slave, it must follow the following communication command frame definition and variable address index method to carry out normal communication operation. In case of communication error, see "(g) error response frame".

1.MODBUS frame format (taking MODBUS-RTU as an example)

◆ Function code 0x01 (01): read coil

Request frame format: slave address + 0x01 + coil start address + coil number + CRC inspection

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x01（function code）	1 bytes	Reda coil
3	Coil start address	2 bytes	High in front, low in back, see coil addressing
4	Number of coils	2 bytes	High in front, low in back (n)
5	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x01 + bytes + coil status + CRC inspection

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x01 (function code)	1 bytes	Reading coil
3	Bytes	1 bytes	Value: [(n + 7) / 8]
4	Coil status	[(N+7) /8] bytes	Every 8 coils are combined into one byte. If the last one is less than 8 bits, the undefined part is filled with 0. The first 8 coils are in the first byte, and the coil with the smallest address is in the lowest position. By analogy
5	CRC check	2 bytes	High in front, low in back

N maximum is 255.

◆ Function code 0x02(02): read coil

Request frame format: slave address + 0x02 + coil start address + coil number + CRC inspection

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x02 (function code)	1 bytes	Reading coil
3	Coil start address	2 bytes	High in front, low in back, see coil addressing

4	Number of coils	2 bytes	High in front, low in back (N)
5	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x02 + bytes + coil status + CRC inspection

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x02 (function code)	1 bytes	Read coil
3	Number of bytes	1 bytes	Value：[（N+7）/8]
4	Coil status	[（N+7）/8] bytes	Every 8 coils are combined into one byte. If the last one is less than 8 bits, the undefined part is filled with 0. The first 8 coils are in the first byte, and the coil with the smallest address is in the lowest position. By analogy
5	CRC check	2 bytes	High in front, low in back

N maximum is 255.

◆ Function code 0x03(03): read register

Request frame format: slave address + 0x03 + start address of register + number of registers + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x03（function code）	1 bytes	Read register
3	Register start address	1 bytes	High in front, low in back, see register addressing
4	Number of register	2 bytes	High in front, low in back（N）
5	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x03 + bytes + register value + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x03（function code）	1 bytes	Read register
3	Number of bytes	1 bytes	Value: N*2
4	Value of register	N*2 bytes	Every two bytes represents a register value, with the high bit first and the low bit last. Small register address in front
5	CRC check	2 bytes	High in front, low in back

N maximum is 125.

◆ Function code 0x03(04）：read register

Request frame format: slave address + 0x03 + start address of register + number of registers + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification

1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x04(function code)	1 bytes	Read register
3	Register start address	2 bytes	High in front, low in back, see register addressing
4	Number of register	2 bytes	High in front, low in back (N)
5	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x04 + bytes + register value + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x04(function code)	1 bytes	Read register
3	Number of bytes	1 bytes	Value：N*2
4	Value of register	N*2 bytes	Every two bytes represents a register value, with the high bit first and the low bit last. Small register address in front
5	CRC check	2 bytes	High in front, low in back

N maximum is 125.

- Function code 0x05(05) write coil

Request frame format: slave address + 0x05 + coil address + coil status + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x05(function code)	1 bytes	Write one coil
3	Coil address	2 bytes	High in front, low in back, see coil address
4	Coil staus	2 bytes	High in front, low in back, FF00 valid
5	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x05 + coil address + coil status + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x05(function code)	1 bytes	Write one coil
3	Coil address	2 bytes	High in front, low in back, see coil address
4	Coil staus	2 bytes	High in front, low in back, FF00 valid
5	CRC check	2 bytes	High in front, low in back

- Function code 0x06（06）：write one register

Request frame format: slave address + 0x06 + register address + register value + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x06(function code)	1 bytes	Write one register
3	Register address	2 bytes	High in front, low in back, see register address
4	Value of register	2 bytes	High in front, low in back, 0 is valid
5	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x06 + register address + register value + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x06(function code)	1 bytes	Write one register
3	Register address	2 bytes	High in front, low in back, see register address
4	Value of register	2 bytes	High in front, low in back, except for 0 is valid
5	CRC check	2 bytes	High in front, low in back

◆ Function code 0x0f（15）：write multiple coils

Request frame format: slave address + 0x0f + coil start address + coil number + bytes + coil status + CRC test.

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x0f(function code)	1 bytes	Write multiple coils
3	Coil start address	2 bytes	High in front, low in back, see coil address
4	Number of coil	2 bytes	High in front, low in back. N, maximum is 1968
5	Number of bytes	1 bytes	Value：[（N+7）/8]
6	Coil staus	[（N+7）/8] bytes	Every 8 coils are combined into one byte. If the last one is less than 8 bits, the undefined part is filled with 0. The first 8 coils are in the first byte, and the coil with the lowest address is in the lowest position. By analogy
7	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x0f + coil start address + coil number + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x0f(function code)	1 bytes	Write multiple coils

3	Coil start address	2 bytes	High in front, low in back, see coil address
4	Number of coil	2 bytes	High in front, low in back
5	CRC check	2 bytes	High in front, low in back

N maximum is 255.

◆ Function code 0x10（16）：write mutiple register

Request frame format: slave address + 0x10 + start address of register + number of registers + number of bytes + register value + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x10（function code）	1 bytes	Write mutiple register
3	Register start address	2 bytes	High in front, low in back, see register addresss
4	Number of register	2 bytes	High in front, low in back. N maximum is120
5	Number of bytes	1 bytes	value：N*2
6	Value of register	N*2（N*4）	
7	CRC check	2 bytes	High in front, low in back

Response frame format: slave address + 0x10 + start address of register + number of registers + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	0x10（function code）	1 bytes	Write mutiple register
3	Register start address	2 bytes	High in front, low in back, see register addresss
4	Number of register	2 bytes	High in front, low in back. N maximum is120
5	CRC check	2 bytes	High in front, low in back

N maximum is 120.

◆ Error response frame

Error response: slave address + (function code + 0x80) + error code + CRC test

Serial number	Data (byte) meaning	Number of bytes	Specification
1	Slave address	1 bytes	Value 1 ~ 247, set by D8121
2	Function code+0x80	1 bytes	Erro function code
3	Erro code	1 bytes	1~4
4	CRC check	2 bytes	High in front, low in back

2. Variable address

■ Coil address

Coil: refers to the position variable, with only two states 0 and 1. In V5, variables such as M, S, T, C, X, Y are included.

Variable name	Start address	Number of coil
M0~M7679	0x0000（0）	7680
M8000~M8511	0x1F40（8000）	512
SM0~SM1023	0x2400（9216）	1024
S0~S4095	0xE000（57344）	4096
T0~T511	0xF000（61440）	512
C0~C255	0xF400（62464）	256
X0~X377	0xF800（63488）	256
Y0~Y377	0xFC00（64512）	256

■ Register address

Register: refers to 16 bit or 32-bit variables. In this PLC, 16 bit variables include D, T, C0~ 199; 32-bit variables are C200 ~ 255.

Variable name	Start address	Number of register	Specification
D0~D8511	0（0）	8512	16 bit register
SD0~SD1023	0x2400	1024	16 bit register
R0~R32767	0x3000	32768	16 bit register
T0~T511	0xF000（61440）	512	16 bit register
C0~C199	0xF400（62464）	200	16 bit register
C200~C255	0xF700（63232）	56	32 bit register

Note: when accessing C200 ~ c255 32-bit register through modbus, one register is treated as two registers, and one 32-bit register occupies two 16 register spaces. For example, the user needs to read or write four registers c205 ~ c208, the Modbus address is 0xf70a (0xf700 + 10), and the number of registers is 8 (4 * 2).

32-bit registers do not support writing single register (0x06) function codes.

8.5.3 MODBUS Mailing address

When V5 motion controller is used as MODBUS slave station, the address of soft element is as follows:

1. Address of font variable register

Represents 16 bit (word) or 32-bit (doubleword) variables. The first address of these variable types is shown in the table below. The specific address (first address + variable serial number) of each register is as follows:

Variable name	Initial address	Number of registers	Description
D0~D8511	0x0000（0）	8512	16 bit register
SD0~SD1023	0x2400（9216）	1024	16 bit register

R0~R32767	0x3000（12288）	32768	16 bit register
T0~T511	0xF000（61440）	512	16 bit register
C0~C199	0xF400（62464）	200	16 bit register
C200~C255	0xF700（63232）	56	32 bit register

Note: when accessing 32-bit registers through modbus, one register is treated as two registers. To read or write the five registers c200-c205, the Modbus address is 0xf700, and the number of registers is 10 (5 * 2).

Coil address of bit variable

Bit variable is also called coil, such as M / S / T / C / X / Y and other variables, with only two states of 0 or 1. These are variable addresses as follows.

Specific address (first address + variable serial number).

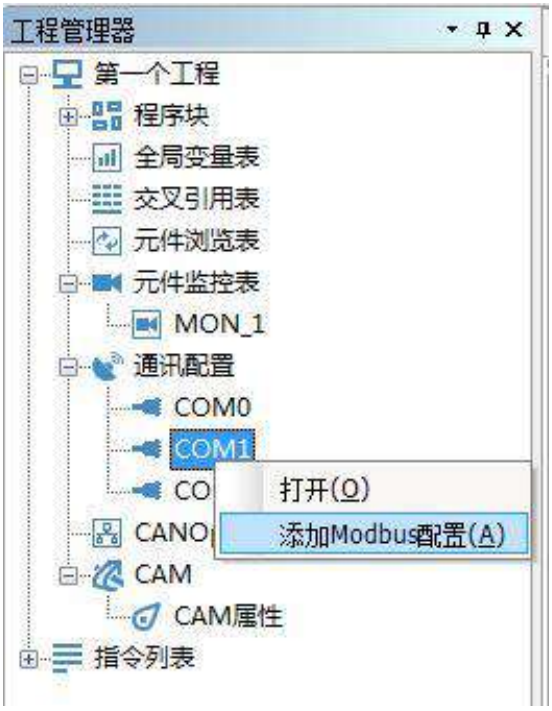
Variable name	Initial address	Number of coils
M0~M7679	0x0000（0）	7680
M8000~M8511	0x1F40（8000）	512
SM0~SM1023	0x2400（9216）	1024
S0~S4095	0xE000（57344）	4096
T0~T511	0xF000（61440）	512
C0~C255	0xF400（62464）	256
X0~X377	0xF800（63488）	256
Y0~Y377	0xFC00（64512）	256

8.5.4 MODBUS configuration instructions

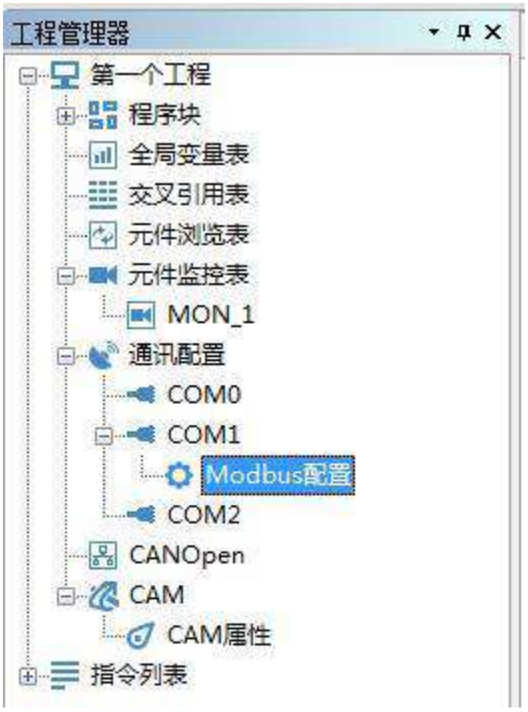
Modbus configuration can realize centralized configuration of Modbus communication data, download configuration data to MODBUS master station, PLC board software can realize communication with slave station equipment according to configuration information, so as to achieve the purpose of data exchange; using MODBUS configuration reduces the difficulty of using modbus, reduces workload, and improves the effect of user experience.

■ Create Modbus configuration

In the project manager, right-click COM1 or COM2 under the “Communication Configuration” node, and select Add MODBUS configuration in the pop-up menu to add the Modbus configuration node, as shown in the following figure:



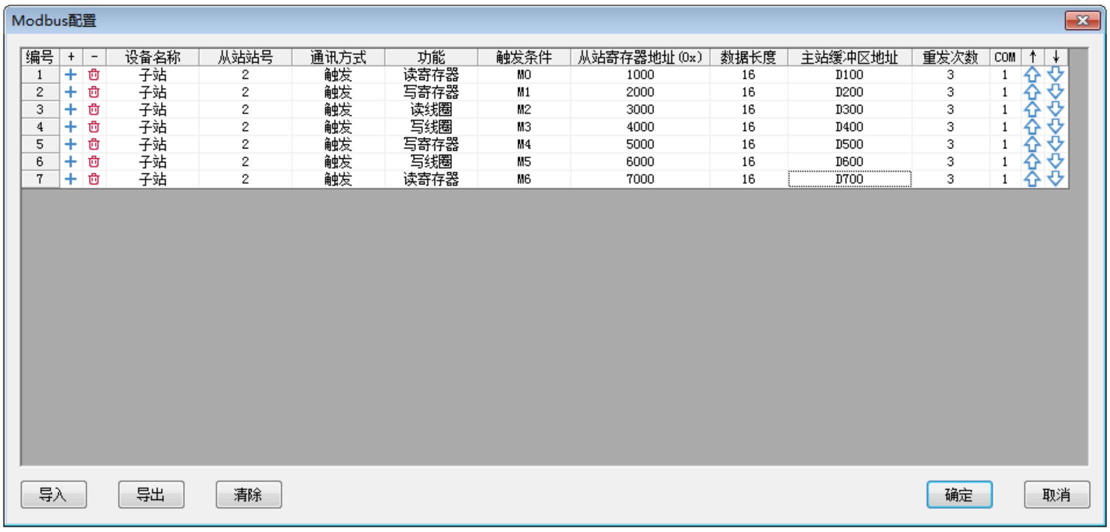
工程管理器
第一个工程
程序块
全局变量表
交叉引用表
元件浏览表
元件监控表
MON_1
通讯配置
COM0
COM1
CO
打开(O)
添加Modbus配置(A)
CANO
CAM
CAM属性
指令列表



工程管理器
第一个工程
程序块
全局变量表
交叉引用表
元件浏览表
元件监控表
MON_1
通讯配置
COM0
COM1
Modbus配置
COM2
CANOpen
CAM
CAM属性
指令列表

- Modbus configuration data addition

Double click the "MODBUS configuration" node with the left mouse button to open the Modbus configuration interface, and set the relevant parameters as follows:



Interface operation instructions:

Button	Icon	Function
Add	+	Add a new configuration record on the next line
Delete		Delete configuration record of current line
Up		Move the current row up one row, and the position remains the same when the first row is moved up
Down		Moves the current row down one row, and the position of the last row does not change when it moves down
Import		Import data from Excel to interface, convenient for data processing
Export		Export date from interface to excel, convenient for data processing
Eliminate		Clear all configuration information in the window

Configuration data description:

Configuration parameter	Parameter description
Device name	User defined text messages will not be downloaded to PLC, just for the convenience of distinguishing master and slave stations on the interface
Slave station number	Slave station number in decimal, range (1 to 247)
Communication mode	The communication mode is divided into "trigger" and "cycle". Trigger: "trigger condition" set "on" communication operation is triggered; Trigger element type: m, s; Cycle: "trigger condition" is "empty", and communication operation is executed in cycle;
Function	Four functions: read register, write register, read coil and write coil
Trigger condition	When the communication mode is "trigger", the trigger condition element can be input, and the element type is m and s.
Slave register address	Communication operation starts from the operation address of the station, which is expressed in hexadecimal.
Data length	The data length of read register and write register indicates the number of registers, it is n registers starting from the address of station register; the data length of read coil

	and write coil indicates the number of coils, it is n coils starting from the address of station register;
Master buffer address	When the master and slave stations exchange data, the buffer is the D element of PLC, and the number of elements is determined by the data length.
Retransmission times	When the communication mode is "trigger", the number of times of sending
COM	Master Modbus communication serial port

■ When Modbus is the master station, how to set the Protocol of the configuration table

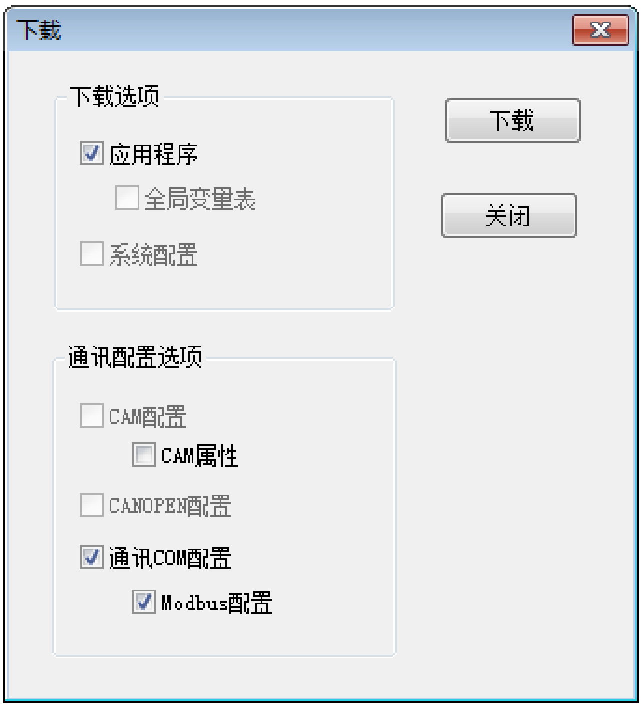
In addition to configuring MODBUS configuration parameters, the COM port corresponding to MODBUS configuration should also be configured. COM1 and com2 can be used as communication ports of Modbus master station. If the communication port of Modbus master station is COM1 port, in "project manager", double-click "COM1" node under "communication configuration" node or right-click "COM1" node and select "open" to enter "COM1 configuration" interface, as shown below:

Then select the "enable communication parameters" option, select the "MODBUS-RTU master station" as the protocol type, and modify the required "baud rate", "check bit", "stop bit", "communication timeout" and other parameters. The master station number uses "1" by default, and the modified interface is as shown in the figure below. You can click "read configuration" or "write configuration" to read and set the configuration. Set, click the "OK" button to complete the master station communication

settings.

Modbus master station configuration download:

Click the "download" menu in the "PLC" menu, or use the shortcut key "F8", or click the download button（）in the "PLC Toolbar" to open the download window, select "communication com configuration" and "MODBUS configuration", and click the "download" button to complete the download of Modbus master communication agreement configuration, as shown in the following figure:



Note: if the communication protocol has been set before, just download "MODBUS configuration".

■ MODBUS slave station configuration

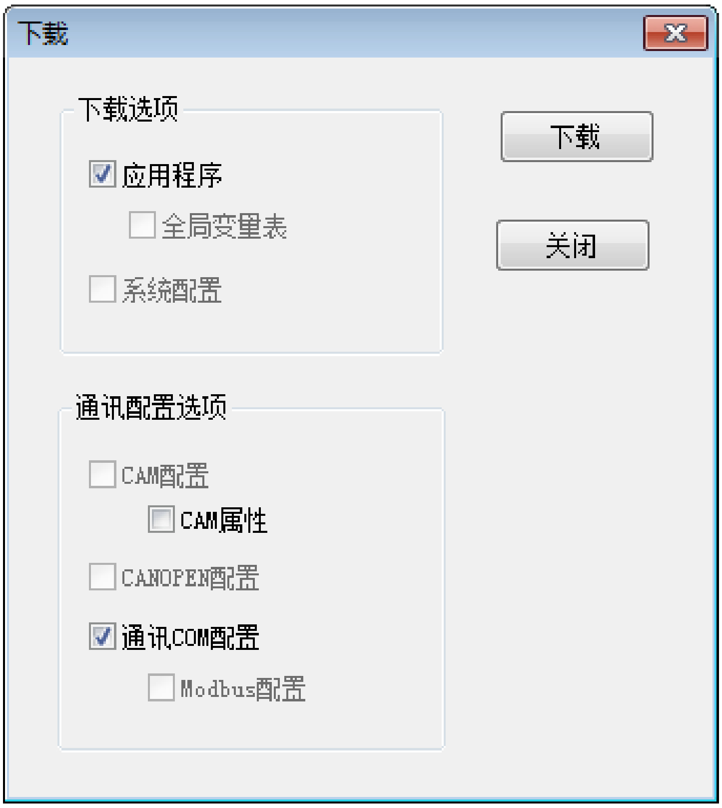
In addition to setting the communication configuration of the master station, it is also necessary to configure the communication parameters of the slave station. COM1 and com2 can be used as the communication ports of the Modbus slave station. If the communication port of Modbus slave station is COM1 port, in "project manager", double-click "COM1" node under "communication configuration" node or right-click "COM1" node and select

"open" to enter "COM1 configuration" interface, as shown below:

Then select the "enable communication parameters" option, select the "MODBUS-RTU / QLink slave" protocol type, and modify the required "baud rate", "check bit", "stop bit", "station number", "communication timeout" and other parameters. Here, set the slave station number as "2". The interface after Modify is as shown in the figure below. Click the "OK" button to complete the communication settings of the slave station:

Modbus slave station configuration download:

There is no "MODBUS configuration" data in the slave station, select "communication com configuration" in the "download" window, and click "download" button to complete the configuration download of Modbus slave communication protocol, as shown in the following figure:

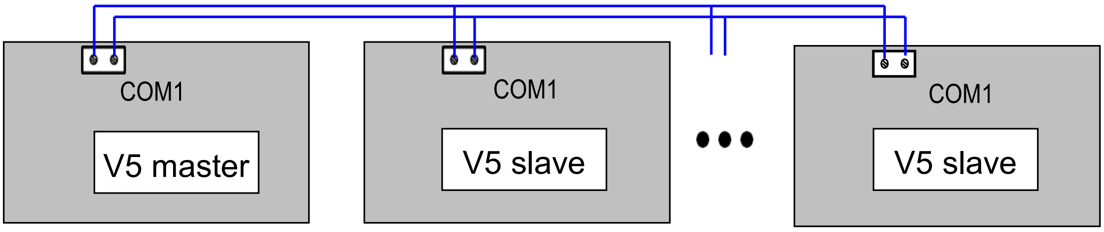


8.5.5 MODBUS command instructions

■ MODBUS master station communication application

COM1 communication port of V5 motion controller can be set as MODBUS-RTU or MODBUS-ASC master station.

（1）Hardware connection



（2）Software setting

Set as MODBUS-RTU master station, baud rate as 9600, check bit as none, stop bit as 2, station number as 1, communication timeout as 10, click OK after setting.

（3）Agreement description

The Modbus command is effective for the serial port COM1. The user can program through the Modbus command, and use V5 as the master station to communicate with the Modbus slave station equipment.

Multiple MODBUS commands can exist at the same time and all of them are driven. The system will coordinate the order of command execution. The Modbus protocol requires the slave station to have a response (except broadcast) no matter whether it is written or read. A MODBUS command may take a long time to execute, generally requiring multiple scan cycles. During a scan cycle, instructions are driven, but not necessarily executed.

If there are multiple MODBUS commands, the execution sequence is as follows: scan the first driven MODBUS command from the start of power on. If it is scanned, record the parameters of the Modbus and execute them in the background. After execution, return to the user program, scan the next driven MODBUS command from the position of the newly executed MODBUS command, and execute it again and again.

Command format：MODBUS(S1，S2，n，D)

①S1：The slave address and MODBUS function code. The high 8 bits indicate the slave address, that is, the target device address. The low 8 bits represent MODBUS function code, which is defined by standard Modbus protocol. Currently, the supported function codes are 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0f, 0x10. Please refer to standard Modbus protocol or target device Modbus protocol for specific meaning.

②S2：For the slave coil (1 bit) or register (16 bit) address to be read or written, please refer to the slave Modbus protocol. Can be element or constant.

③n：The number of slave coils or registers to be read and written can be components or constants.

④D：Only D components. This machine is used to store the starting register of data, that is, data buffer. Buffer length is related to regen, at least 1. If the Modbus command is read, the slave data will be read into the buffer after the command is successfully executed. If the Modbus command is write, the buffer will be sent to the slave. When designing programs, users need to calculate the buffer length and reserve enough registers as buffers.

Related status flag

①M8122：Modbus command execution status indication, when off, it means the execution is completed, when on, it means the execution is in progress. If m8122 is off and the instruction flow is valid in a scanning cycle, and m8122 is set to on, the system will record the instruction parameters and transfer to the background to execute the communication requirements of the instruction. After the communication is executed, when the command is run to the position of the command again, m8122 will be reset to off regardless of whether the command stream is valid or not, the next valid command of the energy stream will be scanned immediately, the command parameters will be recorded and the communication requirements of the command will be executed in the background.

②M8123：Command communication status indication, on indicates abnormal communication, off indicates normal communication.

③M8063：Instruction error indication, error code stored in D8063.

④D8063：Error code (see list of communication error codes).

Note：

When reading and writing registers, V5 meets the requirements of Modbus protocol standard, and supports reading 125 registers and writing 123 registers at a time. If the reading and writing exceeds the maximum value, a parameter error will be reported.

When reading and writing coils, V5 meets the requirements of Modbus protocol standard. It can read 255 coils and write 255 coils at a time. If the reading and writing exceeds the maximum value, a parameter error will be reported.

（4）Example 1:　Read the register with slave address 100, and store the data in D100.

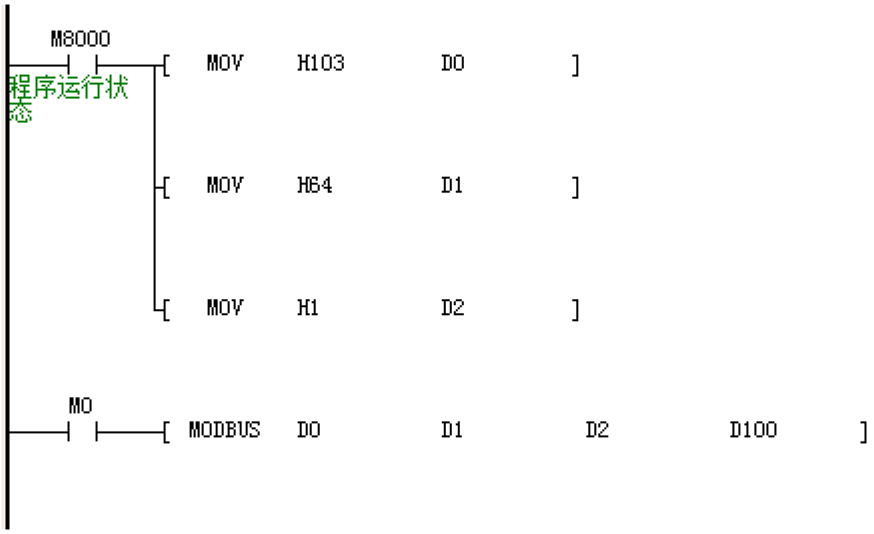
D0 = h0103 slave address is 01, function code is 03;

D1 = register address of the slave to be operated by h0064;

D2 = h0001 number of registers to be operated;

D100 data buffer, which stores the read data in D100.

The ladder diagram is as follows:



When M0 is ON, read the register with slave address 100 and store it in D100.

Send the following frame of data (hexadecimal) through COM1: 010300640001D5E5

01: represents the slave address, the top 8 bits of D0;

03: MODBUS command code, lower 8 bits of D0, meaning read slave register;

0064: read slave register address, D1 value;

0001: number of registers to be read, D2 value;

D5E5: CRC check code.

If the slave is also a V5 motion controller, it is set to ModbusRTU slave protocol, and the ladder diagram is as follows:

The configuration of slave station is as follows:

Correct corresponding data frame of slave (hexadecimal): 0203020520F2C3

The slave sends D100 (register address is H0064) to the host:

02: represents the slave address;

03: MODBUS command code;

02: it means to reply 2 bytes of valid data;

0520: register data, that is, the value of D100;

F2C3: CRC check code.

■ MODBUS slave station communication application

In some industrial applications, the V5 motion controller, as a part of the industrial automation system, should accept the monitoring of the automation control network. The typical upper computer, such as DCS, industrial PC running configuration software, etc., acts as the monitoring host, communicates with the V5 motion controller and other devices by MODBUS master station protocol. At this time, the communication port of V5 needs to communicate with the upper computer by MODBUS slave station protocol. The V5 motion controller has built-in MODBUS-RTU slave protocol, which can be run on both COM1 and com2 ports.

COM1 slave configuration

1. Agreement description

Modbus slave protocol includes ModbusRTU protocol (hereinafter referred to as RTU protocol) and modbusasc protocol (hereinafter referred to as ASC protocol). The difference between the two is that the data RTU protocol transmitted by communication is the real data, and the data transmitted by ASC protocol is the data converted into ASC code. In addition, there are also differences in frame structure between the two. RTU protocol distinguishes data frames by time. If there are 3.5 bytes of time in the communication and no data is received, it is considered that the data transmission of the other party is completed. ASC protocol takes ASC code ":" as the frame start character, and \ Cr \ LF (0d0ah) as the frame end character. From the perspective of communication efficiency, RTU protocol is higher than ASC protocol, probably RTU protocol is ASC protocol. Twice as much.

8.6 CANopen communication

8.6.1 Overview

1. CAN is the abbreviation of controller area network (CAN), which is a kind of serial communication mode. The baud rate of communication can reach 1Mbps. CANopen is an application layer protocol of can network. CANopen application layer communication protocol specification is called cia301, also known as ds301. Based on the definition of cia301 for individual equipment, such as cia401 for I / O module and cia402 for motion control.

2. V5 motion controller supports CANopen protocol and only supports master station mode. Set the baud rate and node number of the master station through the configuration interface of CANopen. The theory supports 126 slave nodes. In practical application, the maximum number of nodes depends on the performance of the can transceiver used. At present, there are 10 slave nodes.

Model	V5-MC104
Support CANopen Protocol	DS301V4.02
Maximum TPDO/RPDO	64
Number of slave nodes	10
Supported baud rate and corresponding communication distance	1Mbps/25m 800Kbps/50m 500Kbps/100m 250Kbps/250m 125Kbps/500m 100Kbps/600m 50Kbps/1000m 20Kbps/2500m 10Kbps/6700m
Slave PDO mapping register range	D0-D7999

3. Type and description of CANopen communication object

Object	CAN-ID	Description
NMT network management command	000h	The host manages the slave station through NMT message, and carries out "start node", "stop node" and other corresponding operations.
Sync synchronization message	080h	Through synchronous signal, all nodes can upload data or execute application instructions at the same time
Emergency Emergency message	080h + slave node number	When the CANopen node has an error, the node will send a frame of emergency message. When the master station receives this message, it will handle it accordingly.
PDO process data	See Table 1	Used for reading and writing from master

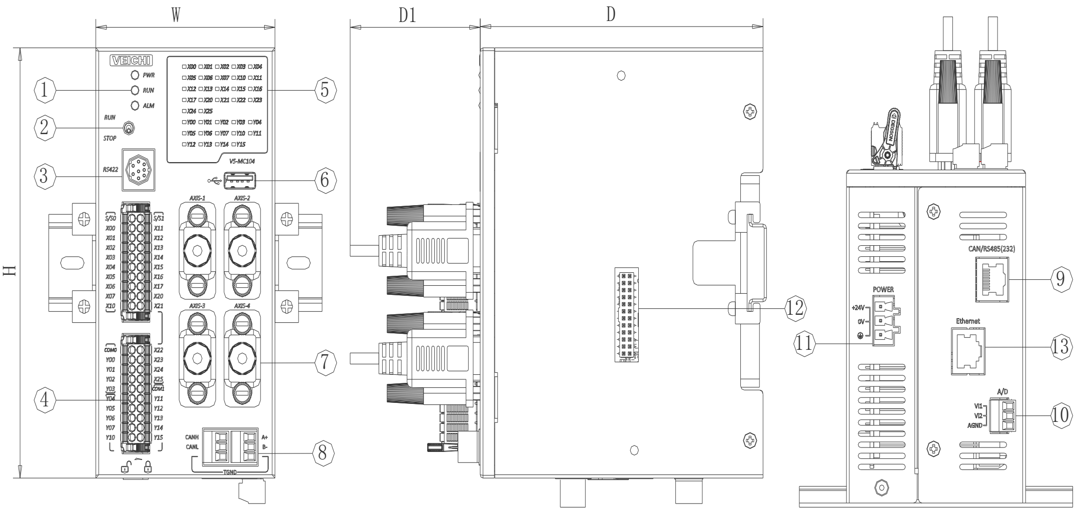
object		station to slave station
SDO service data object	"Answer": 580h + slave node number "Ask": 600H + slave node number	It is mainly used for parameter setting of slave station. When configuring the contents of the slave object dictionary, one question one answer form

Table 1：

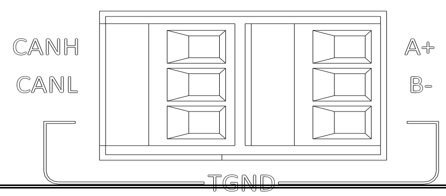
Object	CAN-ID
TPDO1	181h-1FFh（180h+node-ID）
RPDO1	201h-27Fh（200h+node-ID）
TPDO2	281h-2FFh（280h+node-ID）
RPDO2	301h-37Fh（300h+node-ID）
TPDO3	381h-3FFh（380h+node-ID）
RPDO3	401h-47Fh（400h+node-ID）
TPDO4	481h-4FFh（480h+node-ID）
RPDO4	501h-57Fh（500h+node-ID）

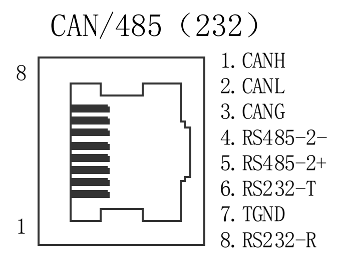
8.6.2 Hardware configuration

1. There are two forms of CAN communication interface of V5 controller: terminal (No. 8 in the figure below) and RJ45 (No. 9 in the figure below). The internal wiring of both cities is interlinked, and only one of them can be used. 120 ohm resistance has been connected inside.

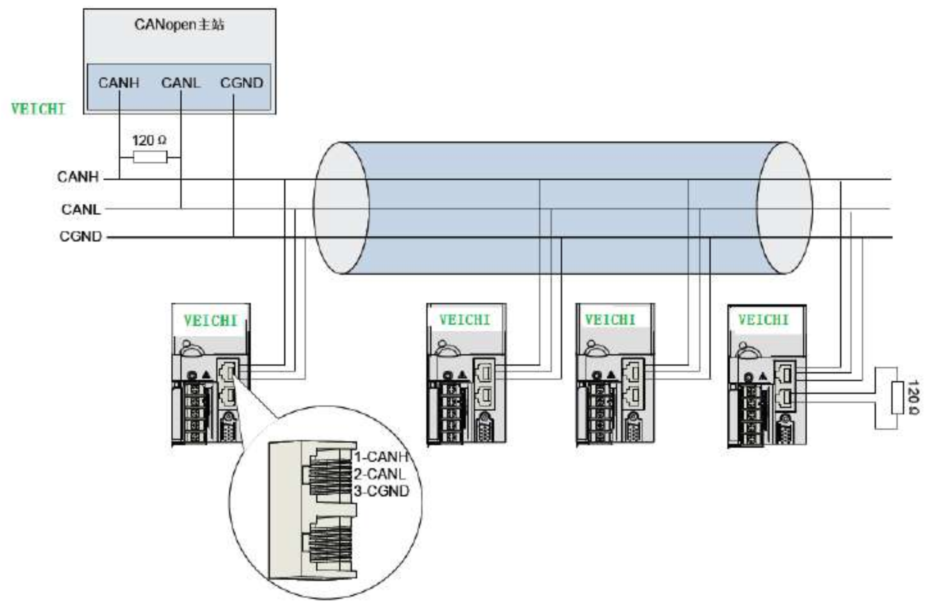


2. Terminal Description:

Name	Function description	Sketch Map
RS485+ RS485-	The first group of RS485 communication signal lines	
CANH CANL	CAN communication line	

TGND	RS485/CAN reference level tgnd, please make sure each tgnd is connected with	
RS232-R RS232-T	RS232 communication signal (cannot be used together with rs485-2)	
TGND	RS232/RS485-2 reference ground	
RS485-2+ RS485-2-	The second group of RS485 communication signal lines	
CANH CANL TGND	The reference level of CAN communication line is tgnd. When using multiple sets, please make sure that each tgnd is connected with each other.	

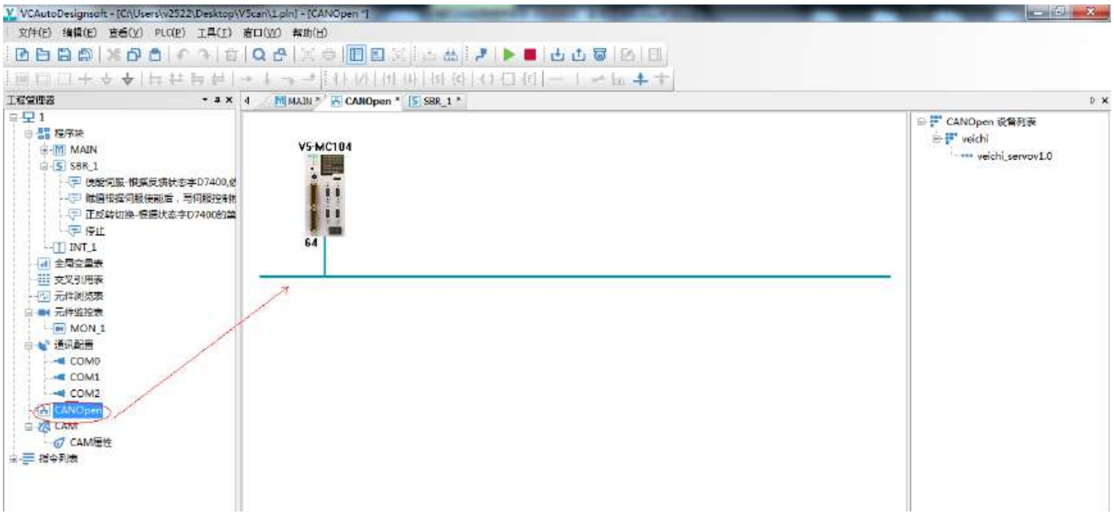
3. Wiring instructions (in order to improve the communication anti-interference and ensure the communication quality, the slave station terminal shall be connected in parallel with 120 ohm resistance):



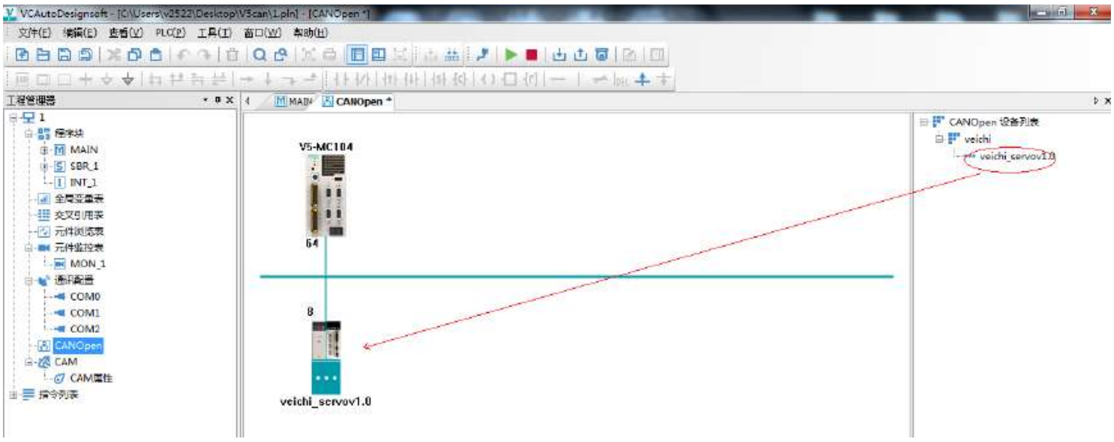
8.6.3 Creat CANopen configuration

The CANopen function is used to configure CANopen communication, including one CANopen master station, which is responsible for managing all slave stations in the network. The maximum number of slave stations is 126. Each device has an independent node address (node ID).

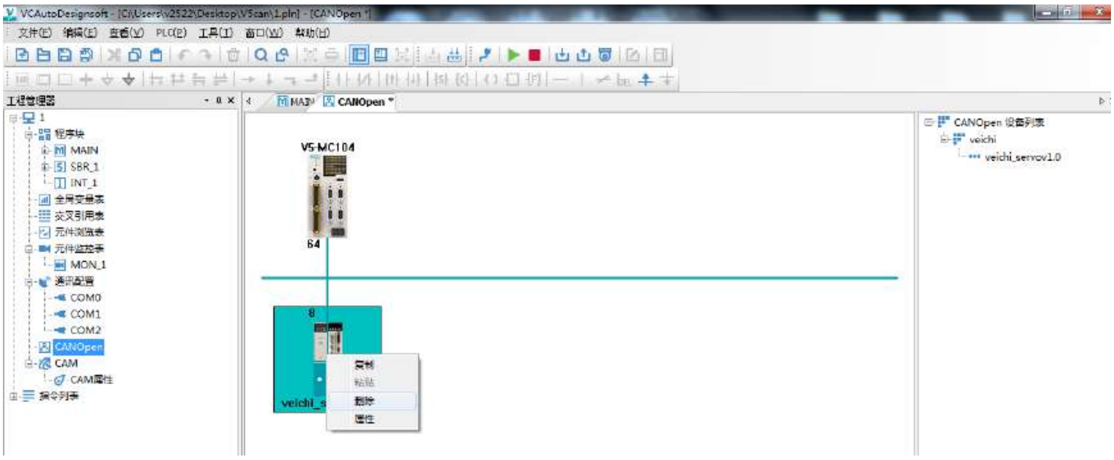
1. Double click “CANopen” in “Project Manager” with the left mouse button, or select open from the right mouse button to open the following configuration interface:



2. Add slave station: double click a device in the "CANopen device list" on the right side of the window to add a slave station, as shown in the following figure:



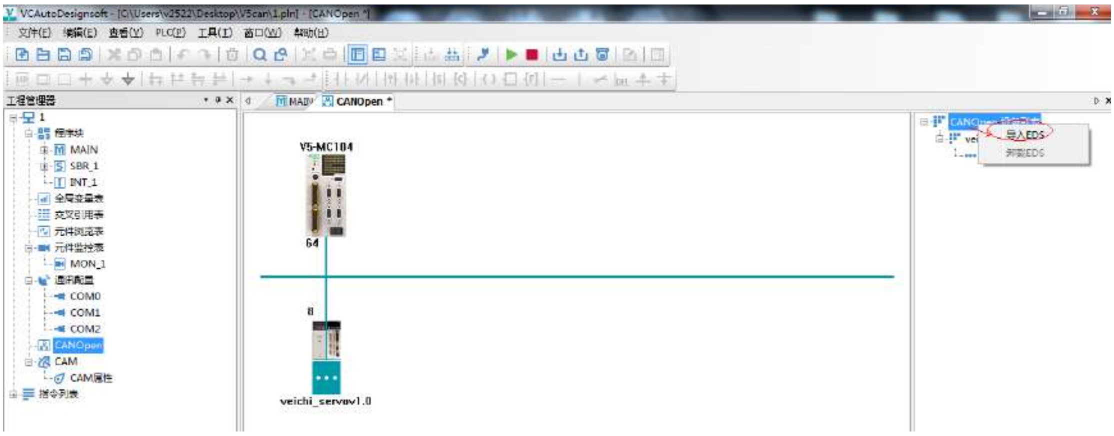
3. Delete slave station: select the slave station in the configuration window, and right-click the pop-up menu to execute "delete" to delete the selected slave station, as shown in the following figure:



4. Copy and paste

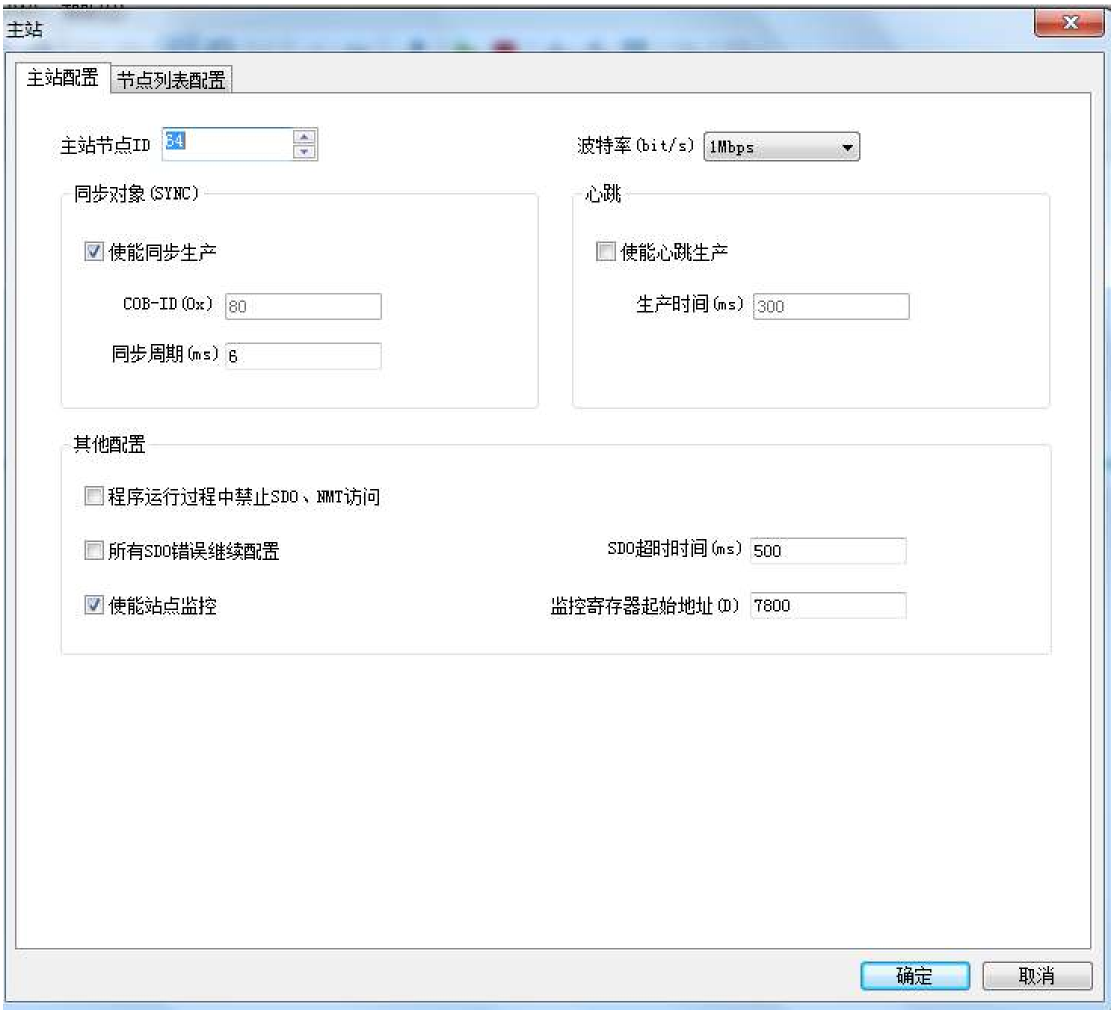
Select "slave station" in the configuration window, right-click to execute "copy" in the pop-up menu, and right-click in the blank to execute "paste" in the pop-up menu.

5. Add new EDS file



8.6.4 Master station configuration

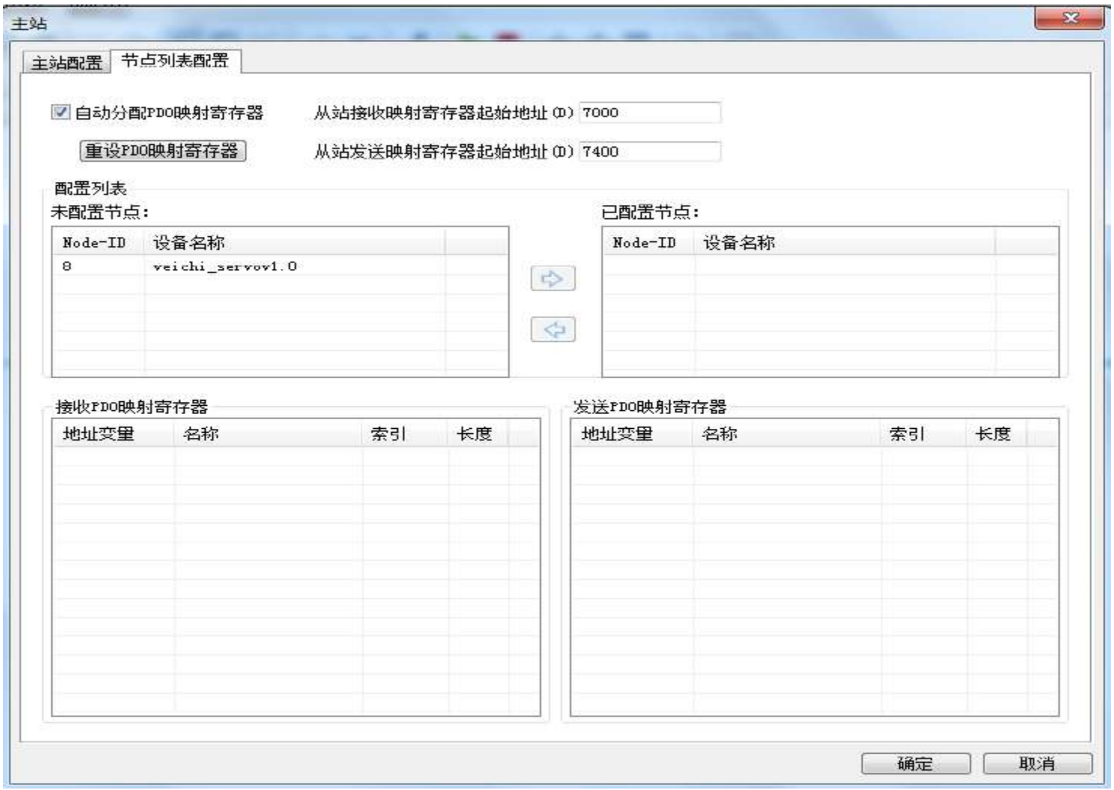
1. Select the "v5-mc104" master station picture in the window, double-click the left mouse button or right mouse button to pop up the menu to execute "properties" to enter the master station configuration interface, as shown in the following figure:



Description of master parameters	
Parameter	Function
Master node ID	Set the master station number. When the station number is the same as the PLC itself, the PLC is initialized to CANopen master station.
Baud rate	Baud rate of effective communication of master station.
Enable synchronous production	If this option is checked, the station will send synchronization frame according to the time cycle set in "synchronization period (ms)".
COB-ID	Synchronous frame sending ID, this item uses the default value 0x80, which is not allowed to be set.
Synchronization cycle(ms)	The cycle period during which synchronous frames are sent. The default is 200, unit ms.
Enable heartbeat production	If this item is checked, the station will send heartbeat frame according to the time cycle set by "production time (ms)".
Production time (ms)	The cycle that sends the heartbeat. The default is 300, unit ms.

Disable SDO and NMT access during program operation	If this function is checked, the online debugging function will not be available during operation. This feature is limited to background software only.
Continue configuration with all SDO errors	After checking this function, if there is an SDO configuration error (except for the verification error), the configuration will continue. This function is valid for all slave stations. If this function is not checked, the master station will reset the broadcast to the slave station in case of SDO error.
SDO timeout	The default is 500, units ms. SDO frame is mainly used as network configuration. SDO failed to receive the return frame on time after 3 retransmissions, and the master station determined that the configuration timed out. The waiting interval of each frame is this time.
Enable site monitoring	If this item is checked, the master station will write the slave station status to the corresponding set register. This item is checked by default.
Start address of monitoring register	The default is 7800. That is to say, d7800 is set as the starting address of equipment status monitoring. D7800 is the master station status, and D (7800 + slave station number) is the corresponding slave station status. The meaning of the status value is as follows: 0 is the initial state, 4 is the stop state, 5 is the operation state, 127 is the pre operation state, and 255 is the offline state. If the corresponding slave does not exist, the corresponding register will not be updated. For example, station 3 does not exist and d7803 data will not be updated. The slave station needs to set the heartbeat or node protection function, which is meaningful. Because this state is fed back by the heartbeat of the slave station or the protection frame of the node.

2. Node list configuration

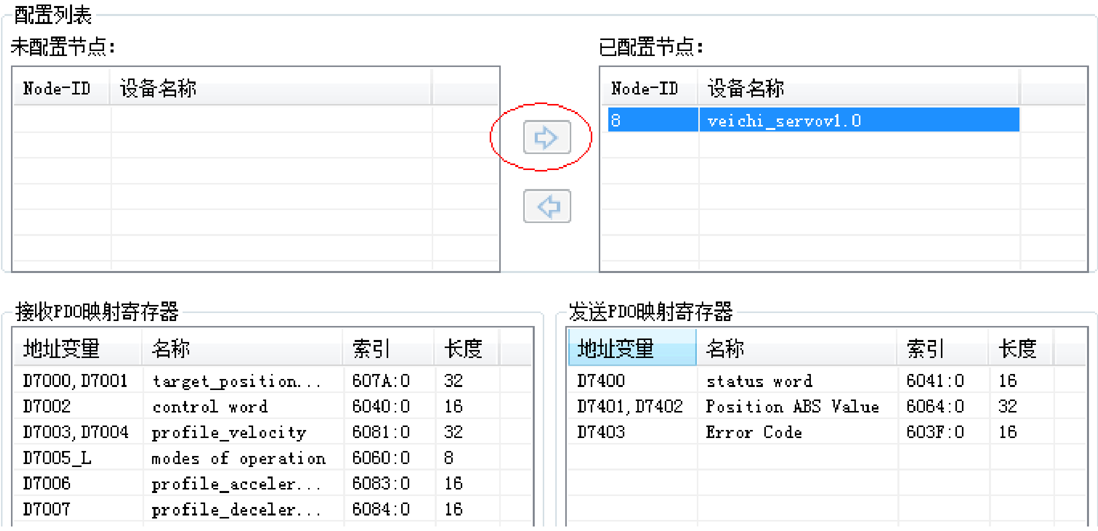


Automatic allocation: if this function is checked, the register address of data interaction between the master and slave station will be automatically allocated; if this function is not checked, the user needs to manually set the starting address of data interaction (set the starting address of each PDO separately), and this function is checked by default.

Starting address of receiving mapping register of slave station: automatically allocate the starting address of data sent by the master station.

Starting address of sending mapping register of slave station: automatically allocate the starting address of receiving data of master station.

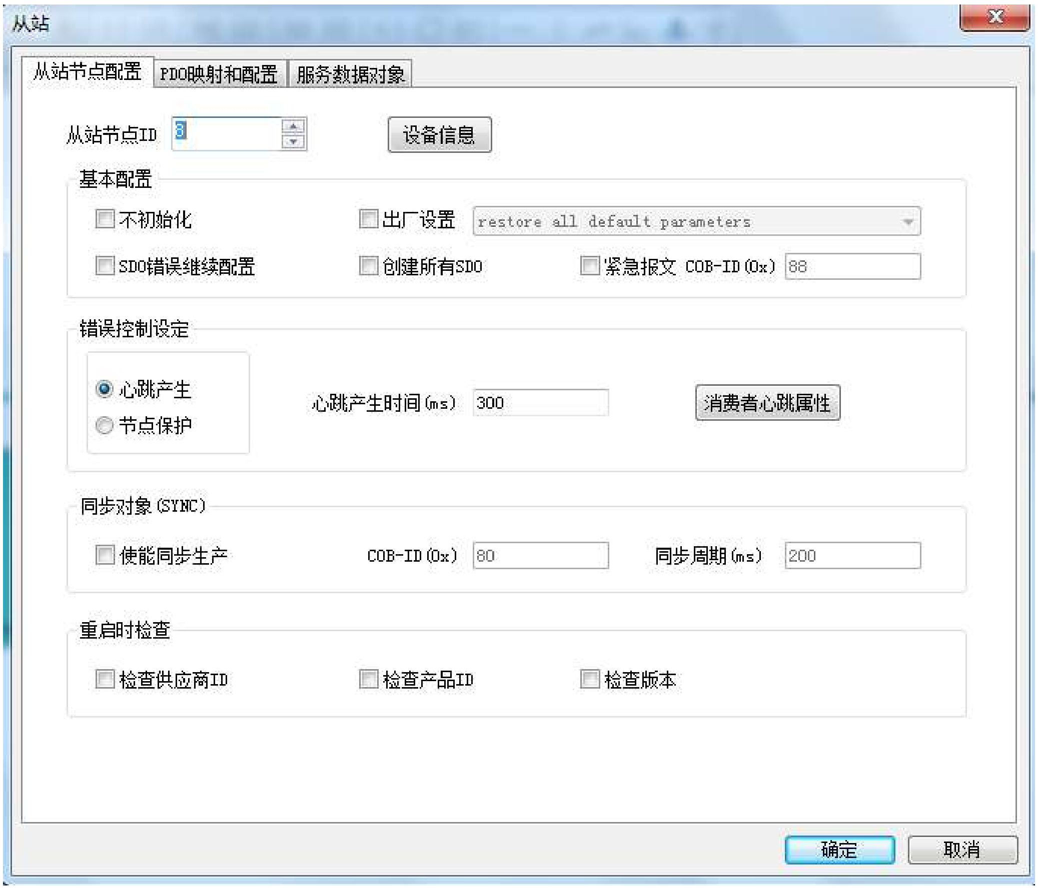
I. in automatic allocation, select the available node and click the right arrow to automatically allocate the address.



II、When canceling the automatic allocation, you need to enter the address manually. In the added node address table, double-click the D register to be allocated manually, and the mapping register setting interface will pop up. Enter the address to be allocated manually. Note that the address should not be repeated during the manual allocation.

8.6.5 Slave station configuration

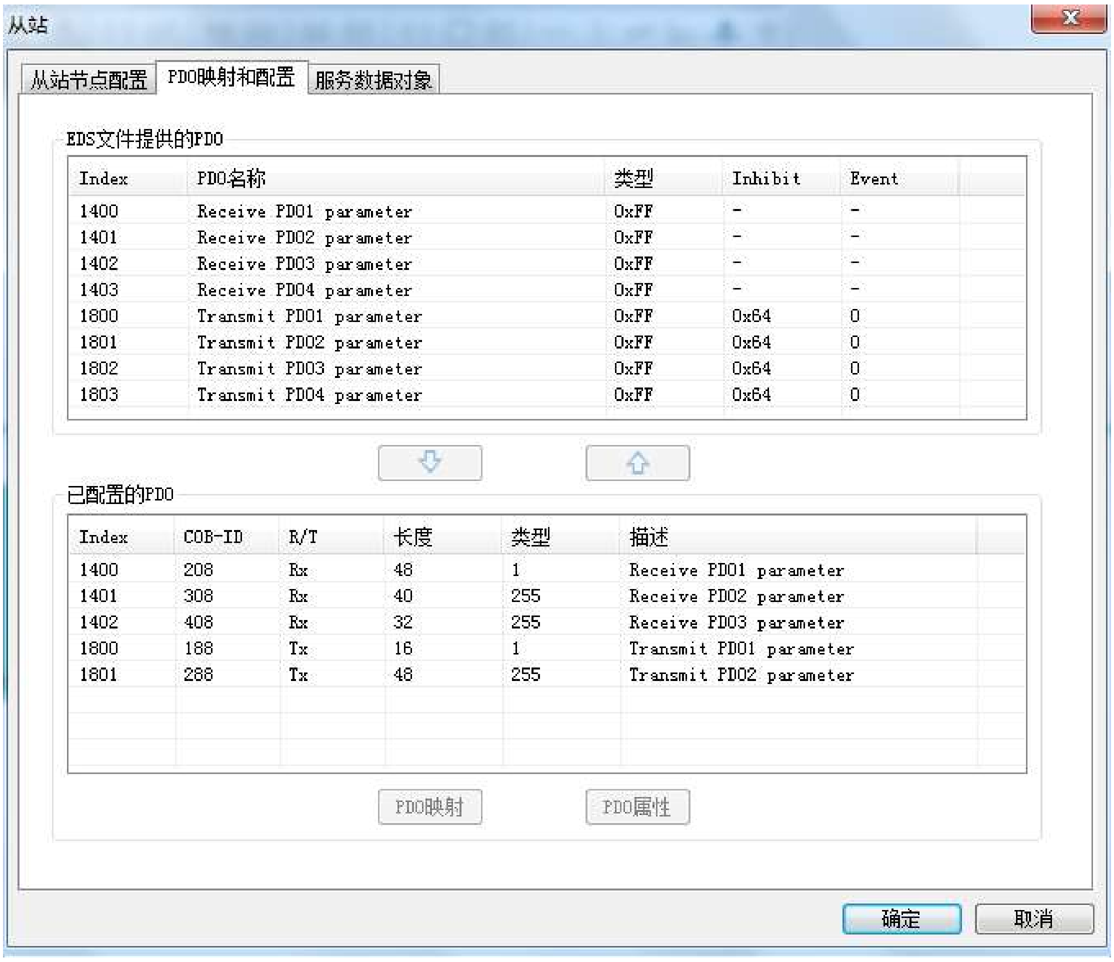
1、 Double click the slave icon in the window to enter the slave configuration interface, as shown in the figure below:



Description of slave parameters	
Parameter	Function
Slave node ID	Set the slave station number. When the station number is the same as the PLC itself, the PLC is initialized to CANopen slave station.
No initialization	When this function is selected, this slave station will not be initialized (it can only be selected if the default configuration is used). It is not checked by default
Factory settings	After checking this function, you can select the corresponding operation later. It is not checked by default (this function can only be checked if the selected slave station supports corresponding functions).
SDO error continue configuration	If there is a configuration error when it is valid, the next configuration will continue (except for the verification type error). If there is a configuration error when it is invalid, the master will not continue to configure, and the whole network will be stopped when the network is started. This option is unchecked by default.
Create all SDOs	When this function is selected, all writable object dictionaries in EDS will be added and initialized during configuration. It is not checked by default.
Emergency message	Check this function to set emergency message cob-id during configuration. It

	is not checked by default.
Heartbeat generation	After checking the function, the slave station will generate heartbeat. Check by default. When heartbeat is checked in the slave station, the master station will monitor the heartbeat status of the slave station by default.
Heartbeat generation time (ms)	The time when the heartbeat cycle is sent.
Consumer heartbeat attribute	This function is used to set the heartbeat of other sites that this slave station will monitor. This feature is not selected by default. This function also needs the slave station to support the heartbeat monitoring function.
Node protection	When this function is checked, the node protection function of the slave station will be set, which is not checked by default. Node protection is a kind of network evaluation function that the master station and slave station monitor each other with return frame. Only one of the heartbeat and node protection functions can be selected. Node protection timeout = protection time * life cycle factor.
Protection time (ms)	Node protection time, default 200ms.
Life cycle factors	Node protection factor, default 3.
Enable synchronous production	If this option is checked, the station will send synchronous frames according to the time cycle set in "synchronization period (MS)", and only one synchronous frame can be sent in a network. The premise is that the slave station should support the sending synchronization function.
COB-ID	Synchronous frame sending ID, this item uses the default value 0x80, which is not allowed to be set.
Synchronization cycle (ms)	The cycle period during which synchronous frames are sent. Default 200, units ms。
Check on restart	Test supplier ID, test product ID, test version: check the corresponding function, and the corresponding verification will be performed before the configuration of the starting station. If the verification fails, the network will not start.

2、PDO mapping and configuration interface



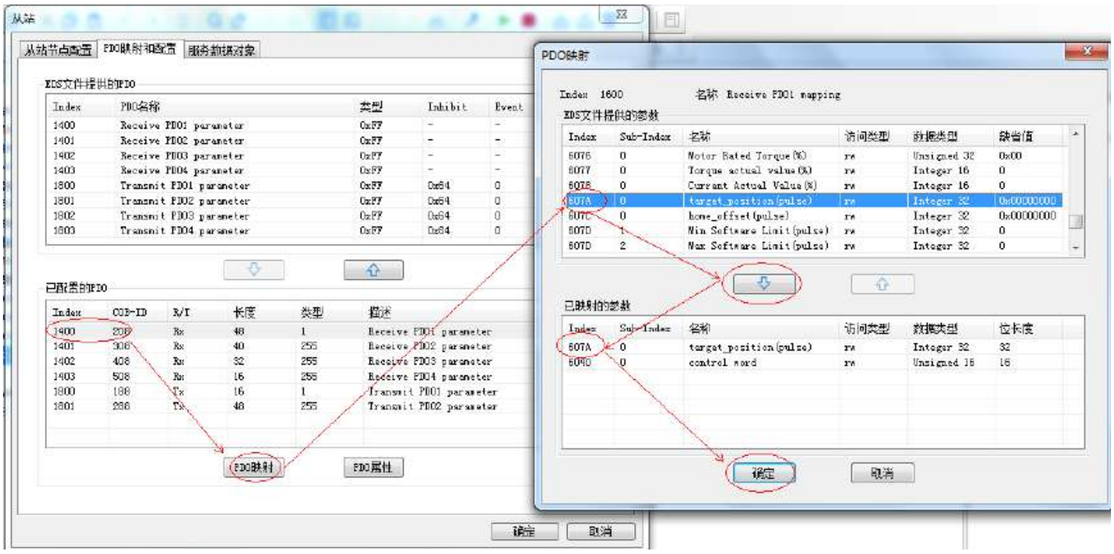
Ⅰ. Select PDO provided by EDS file to add PDO by clicking the down arrow.

Ⅱ. Select configured PDO and click the up arrow to delete the PDO.

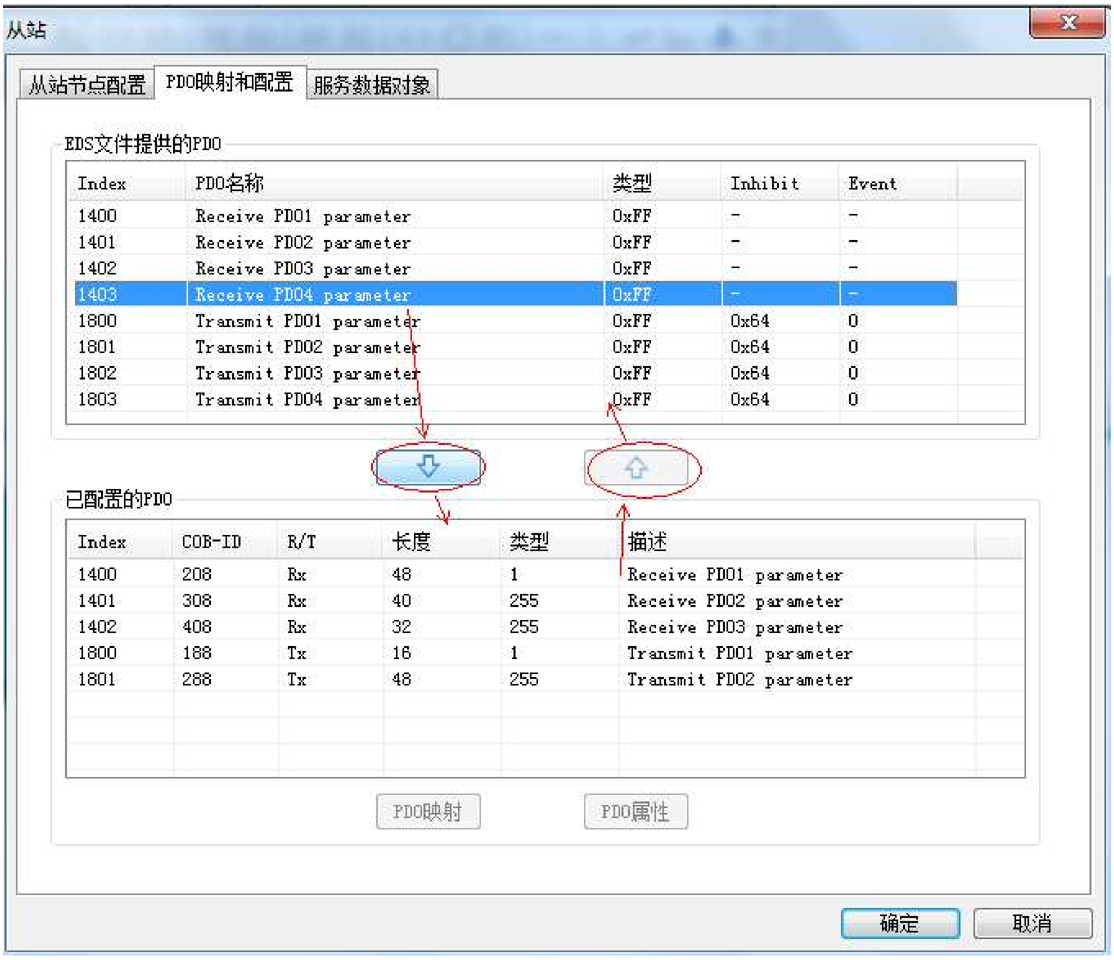
Ⅲ. Select "configured PDO" and click "PDO properties" to set its properties, as shown in the following figure:

<table>
<tr><th colspan="4">PDO attribute description</th></tr>
<tr><th>Parameter</th><th colspan="3">Function</th></tr>
<tr><td>COB-ID</td><td colspan="3">ID number used for PDO sending. According to canopends301 protocol, the first four PDOS have default cob-id initial values, and other PDOS need to be set by users themselves (if supported by the slave). The setting principle is that the whole network can not have duplicate cob-ids, and the setting range is 0x180-0x57f.</td></tr>
<tr><td rowspan="7">Transfer type</td><th>Type</th><th>Data sending conditions</th><th>Data validation conditions</th></tr>
<tr><td>0: Cycle - Synchronization</td><td>Data changes and a synchronous frame is received</td><td>It does not take effect immediately after receiving the data. It takes another frame synchronization to take effect.</td></tr>
<tr><td>1~240: Cycle - Synchronization</td><td>Data transmission after receiving the corresponding synchronization number frame synchronization</td><td>It does not take effect immediately after receiving the data. It takes another frame synchronization to take effect.</td></tr>
<tr><td>252: Asynchronous – Only RTR</td><td>Not support</td><td>Not support</td></tr>
<tr><td>253: Asynchronous – Only RTR</td><td>Not support</td><td>Not support</td></tr>
<tr><td>254: Asynchronous - manufacturer specified</td><td>Customized by each manufacturer</td><td>Customized by each manufacturer</td></tr>
<tr><td>255: Asynchronous - device profile specification</td><td>The data changes or satisfies the event time, and the change frequency is less than the suppression time.</td><td>Immediate effect</td></tr>
<tr><td>Suppression time</td><td colspan="3">When the inhibition time is less than 0.5ms or equal to 0, the inhibition interval is 0.5ms; when the inhibition time is greater than 0.5ms, the inhibition interval is 0.1ms.</td></tr>
<tr><td>Event time</td><td colspan="3">When it is 0, this function means to send by data change. When it is not 0, it means that it is sent according to the timing cycle. (this sending situation is also limited by the suppression time)</td></tr>
</table>

IV. select "configured PDO" and click "PDO mapping" to open the PDO mapping configuration interface. The maximum mapping length of each PDO is 64 bits. As shown in the figure below:

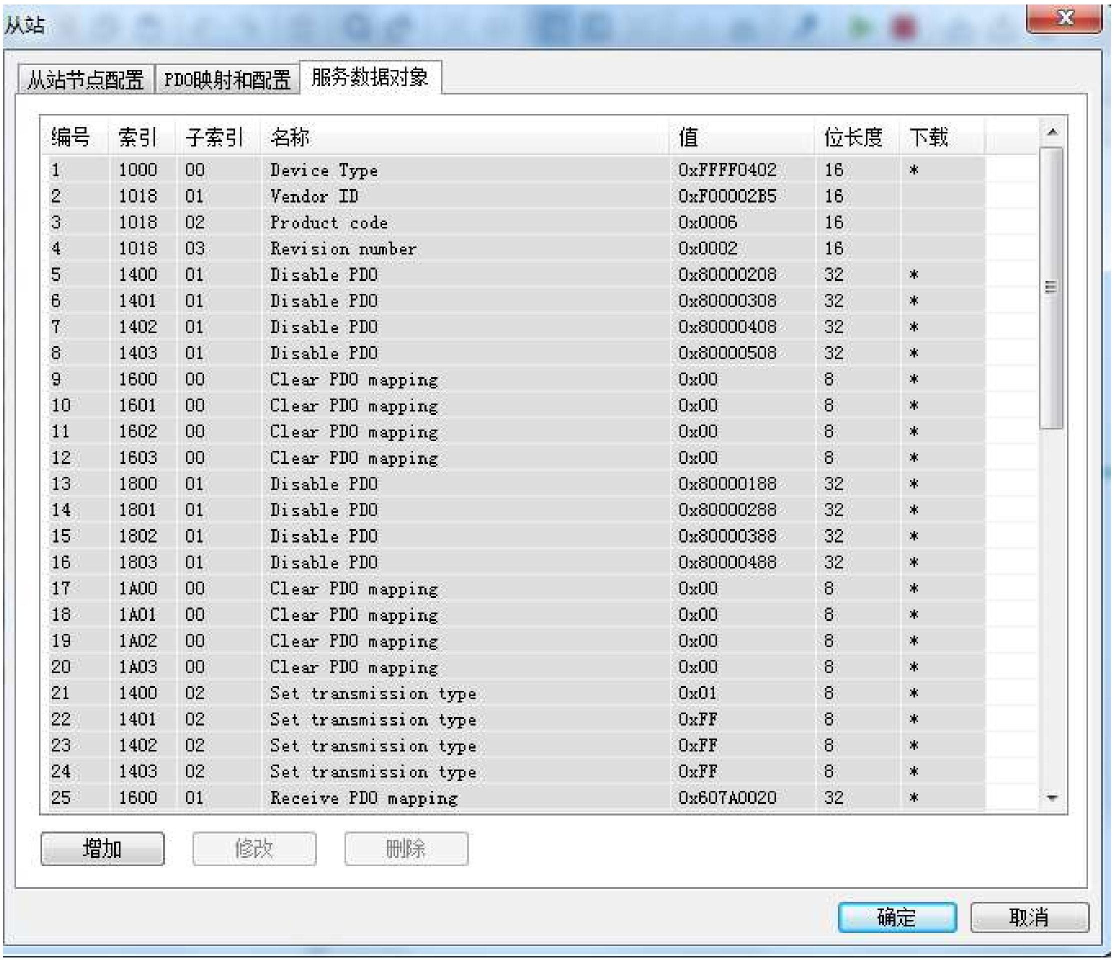


V. select the parameters to be mapped from "parameters provided by EDS file", click the down arrow to add the mapping; select the mapping to be deleted from "mapped parameters", and click the up arrow to remove the mapping, as shown in the following figure:



3、Service data object interface

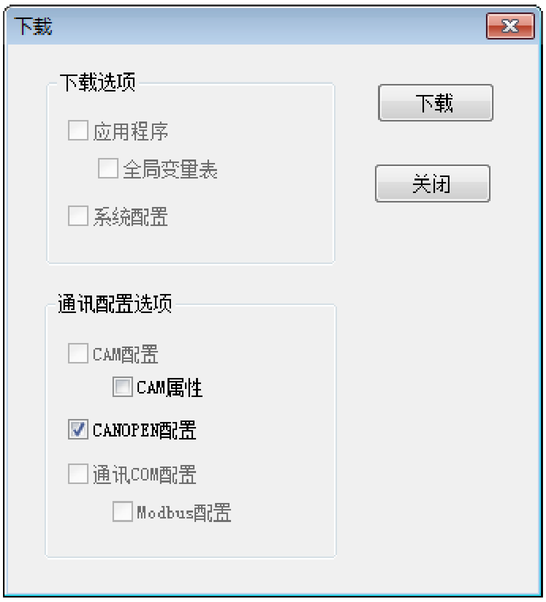
The information in this table is the SDO configuration data automatically generated according to the user's settings. You can also click Add to manually add a data dictionary.



编号	索引	子索引	名称	值	位长度	下载
1	1000	00	Device Type	0xFFFF0402	16	*
2	1018	01	Vendor ID	0xF00002B5	16	
3	1018	02	Product code	0x0006	16	
4	1018	03	Revision number	0x0002	16	
5	1400	01	Disable PDO	0x80000208	32	*
6	1401	01	Disable PDO	0x80000308	32	*
7	1402	01	Disable PDO	0x80000408	32	*
8	1403	01	Disable PDO	0x80000508	32	*
9	1600	00	Clear PDO mapping	0x00	8	*
10	1601	00	Clear PDO mapping	0x00	8	*
11	1602	00	Clear PDO mapping	0x00	8	*
12	1603	00	Clear PDO mapping	0x00	8	*
13	1800	01	Disable PDO	0x80000188	32	*
14	1801	01	Disable PDO	0x80000288	32	*
15	1802	01	Disable PDO	0x80000388	32	*
16	1803	01	Disable PDO	0x80000488	32	*
17	1A00	00	Clear PDO mapping	0x00	8	*
18	1A01	00	Clear PDO mapping	0x00	8	*
19	1A02	00	Clear PDO mapping	0x00	8	*
20	1A03	00	Clear PDO mapping	0x00	8	*
21	1400	02	Set transmission type	0x01	8	*
22	1401	02	Set transmission type	0xFF	8	*
23	1402	02	Set transmission type	0xFF	8	*
24	1403	02	Set transmission type	0xFF	8	*
25	1600	01	Receive PDO mapping	0x607A0020	32	*

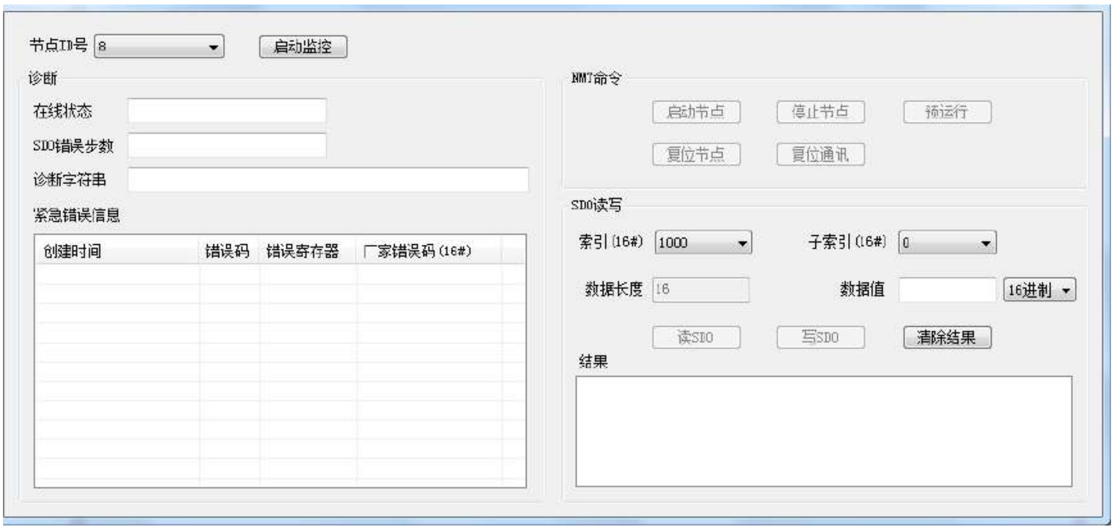
8.6.6 CANopen configuration download

Click the "download" menu in the "PLC" menu, or use the shortcut key "F8", or click the "PLC Toolbar" download button （ ） Open the download window and click the "download" button to complete the download of CANopen configuration, as shown in the figure below:



8.6.7 Online debugging

Click “SDO debugging” in the “Tools” menu to open the SDO debugging interface, select the corresponding node number, start monitoring, and execute the corresponding object dictionary read-write operation and NMT command, as shown in the following figure.



SDO Read & Write description	
Parameter	Description
Node ID	PLC site to read and write.
Index	Index of the PLC site object dictionary.
Sub index	Sub index of the PLC site object dictionary.
Data length	The bit length of the data in the index and sub index.
Data value	To send or return data, you can choose to display in hexadecimal and decimal.
Read SDO	Perform the corresponding object dictionary reading operation.
Write SDO	Perform the corresponding object dictionary write operation.
SDO reception	Data content display received from PLC site.

Diagnostic function	
Parameter	Description
Online status	Display the status of the selected node
SDO error step	Displays the location of SDO configuration errors in the service data object table
Diagnostic string	Abort code of SDO configuration error (see section 6.1.9 SDO abort code)
Emergency error message	Display emergency message generated by master station or slave station (see Chapter 6.1.9 error code of emergency message)

8.6.8 CAN bus monitoring

Click "CAN bus monitoring" in the "Tools" menu to open the CAN bus monitoring interface, as shown in the following figure.

Click start monitoring to monitor the status of the device. The network load is the percentage of the network bus transmission data in the bandwidth within 1s, which reflects the size of the whole can bus communication data and is the basis for judging whether the can configuration data communication mode is reasonable.

It can also monitor the slave station, service object table error location and error code corresponding to the emergency error information and service data object table configuration error.


CAN总线监控
站点状态
网络负载
4%
停止监控
站点
状态
64
运行
8
掉线
紧急错误信息
创建时间
站点
错误码
错误寄存器
厂家错误码(16#)
SDO配置错误
站号
错误步数
错误码(16#)

8.6.9 CANopen Communication fault code and elimination

1, SDO stop code (hexadecimal)

Abort code	Description
0x05040001	Control command is invalid (SDO only supports 0x40, 0x2F, 0x2B, 0x23 instructions)
0x06010002	Trying to write a read-only object
0x06020000	The object in the object dictionary does not exist
0x06040041	Object cannot be mapped to PDO
0x06040042	The number and length of mapped objects exceeds the PDO length
0x06070010	The write length does not match (the length defined by the object dictionary does not match the length of the write)
0x06070012	Data type does not match, service parameter length does not match
0x06090011	Subindex does not exist
0x06090031	Write parameter value is too large
0x06090032	Write parameter value is too small

2、Emergency message error code (hexadecimal)

Emergency error code	Description	Emergency error code	Description
0000	Error reset or no error	6300	Data setting
1000	General error	7000	Add-on module error
2000	Current error	8000	Monitoring error
2100	Device input current	8100	General error
2200	Device internal current	8110	CAN communication overload
2300	Device output current	8120	CAN passive error
3000	Voltage error	8130	Node protection or heartbeat error
3100	Power error	8140	Bus off recovery
3200	Device internal voltage	8150	CAN-ID conflict
3300	The output voltage	8200	Protocol error
4000	Temperature error	8210	PDO length error
4100	Ambient temperature	8220	PDO length exceeded
4200	Equipment temperature	8240	Cannot recognize sync data length
5000	Device hardware error	8250	RPDO timeout

6000	Device software error	9000	External error
6100	Internal software	F000	Additional function error
6200	User software	FF00	Special equipment error

3、Communication failure troubleshooting

(1) It is recommended to use shielded twisted pair connection. Two 120-ohm terminal matching resistors are connected at both ends of the bus to prevent signal reflection. The shielding layer is generally grounded with a single point.

(2) Using a multimeter to measure the resistance between CANH and CANL, you can confirm whether the field termination resistance is correct. The normal resistance should be about 60Ω (two resistors in parallel). If there is no resistance of 60Ω or so, if the bus is not connected, Measure whether the resistance between the head and tail device CANH and CANL is 120 ohms respectively. If there is, the intermediate CAN device may have a matching resistor and remove the matching resistance of the intermediate CAN device. The intermediate device cannot access the matching resistor during CAN bus communication, otherwise it will affect the normal communication of the device.

(3) The master-slave baud rate setting must be consistent, otherwise it will not be able to communicate with the slave.

(4) The station address cannot be repeated, otherwise the master station cannot correctly read and write the slave station with duplicate address.

(5) When long-distance communication of CAN equipment, the common ground CGND of different CAN circuits must be connected to each other to ensure that the reference potentials between different communication devices are equal, and the baud rate is appropriately reduced to ensure normal communication.

(6) When the interference is large and the above method has no effect, the appropriate bus reduces the baud rate.

(7) When the bus load rate is too high (greater than 90%), the bus is blocked, and communication failures and dropped calls may occur. The load rate is high, and the program planning data is often problematic. At this point, the amount of data exchanged should be reasonably planned, and different types of transmission methods should be selected for data exchange. Generally, in the synchronous mode, when the synchronization period is short, the number of synchronization is small, and the amount of data transmitted is large, communication failure occurs and the line is dropped. In this case, the synchronization period or the synchronization number is increased. In the asynchronous mode, if the suppression time is too short, the communication will fail and the line will be dropped. Increase the suppression time.

8.6.10 CANopen Communication variable

CANopen Communication variable table:

M variable	Description	D variable	Description
M8282	0: reset only the current node	D8240	Bus load rate
M8284	1: Current node and total network reset	D8241	Error - the main station number
M8285	Set the CAN address	D8242	Error--error code
		D8243	Error -0x11
		D8244	Error -- auxiliary byte L
		D8245	Error -- auxiliary byte L
		D8246	Online--node
		D8247	Online--index
		D8248	Online--byte length+subindex
		D8249	Online - Data 1
		D8250	Online - Data 2
		D8251	Online - Data 3
		D8252	Online - Data 4
		D8253	Online--command 0: Write SDO 1: read SDO 2: Start node 3: Stop the node 4: Pre-run 5: Reset node 6: Reset communication 0xff: waiting
		D8254	SDO error abort code L
		D8255	SDO error abort code H
		D8284	Set the CAN address
		D8285	Set baud rate
		D8287	CANOpen configuration error station number
		D8288	CANOpen configuration error number
		D8289	CAN bus error
		D8290	CAN receive error count
		D8291	CAN total number of frames sent and received per second
		D7800	Status--master station
		D7801	Status - slave 1
		D78xx	The last one is filled with 255

8.6.11 V5-CANopen Control SD700 Servo Drive

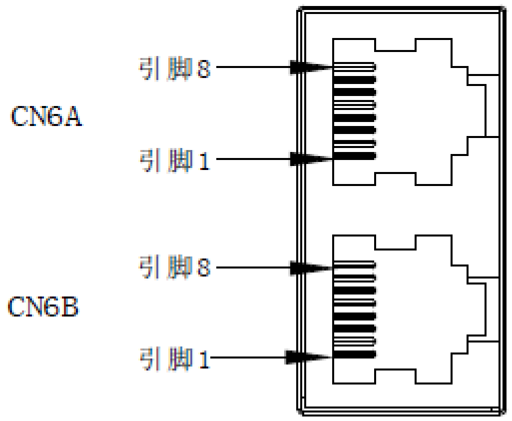
1、SD700 Servo naming (determine whether the servo drive has CAN communication function):

SD700-3R3A-PA*

A B C D E F G

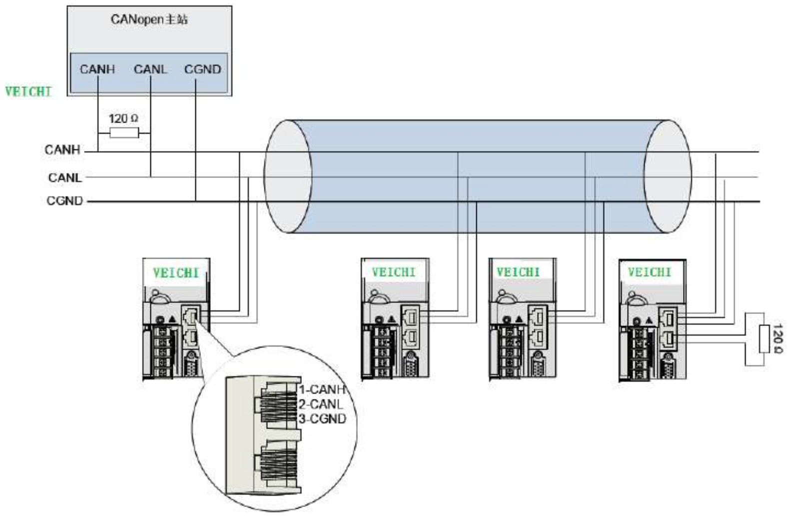
<table>
<tr><th>Field identifier</th><th colspan="2">Field details</th></tr>
<tr><td>A</td><td colspan="2">SD: Servo product code</td></tr>
<tr><td>B</td><td colspan="2">700: Product line</td></tr>
<tr><td rowspan="2">C</td><td colspan="2">Current rating:</td></tr>
<tr><td colspan="2">1R1：1.1A 1R8：1.8A 3R3：3.3A 5R5：5.5A 7R6：7.6A 9R5：9.5A
2R5：2.5A 3R8：3.8A 6R0：6A 8R4：8.4A 110：11A 170：17A
240：24A 300：30A</td></tr>
<tr><td rowspan="2">D</td><td colspan="2">Input voltage level:</td></tr>
<tr><td colspan="2">A: 220VAC; D: 400VAC</td></tr>
<tr><td rowspan="2">E</td><td colspan="2">Machine type:</td></tr>
<tr><td colspan="2">P: pulse type
S: Standard type
C: CANopen bus type
N: EtherCAT bus type
M: MECHATROLINK-II bus type
L: MECHATROLINK-III bus type</td></tr>
<tr><td rowspan="4">F</td><td colspan="2">Supported encoder types:</td></tr>
<tr><td>A</td><td>Absolute value</td></tr>
<tr><td>B</td><td>Incremental</td></tr>
<tr><td>T</td><td>Resolver type</td></tr>
<tr><td>G</td><td colspan="2">Product management number, standard product default</td></tr>
</table>

2、SD700 communication terminals CN6A and CN6B are defined (you can use the network cable to connect the CAN communication ports of V5 and S700):



CN6A/CN6B interface definition					
Pin number	Signal name	Features	Pin number	Signal name	Features
1	CANH	CAN data +	6	-	
2	CANL	CAN data -	7	GND	485 signal ground
3	CANG	CAN signal ground	8	-	-

3、Schematic diagram of CANopen wiring between V5 and SD700:



4、Servo CANopen communication parameters (station address, baud rate, internal position command selection 4-Canopen) settings:

	功能码	参数名称	当前值	单位	范围	默认值
✔	Pn030	保留参数	0x1706	-	0~65535	0x6AA
✔	Pn080	本机通信地址	0x8	-	0~127	0x1
✔	Pn083	CANopen通讯波特率选择	[6]1Mbps		0~6	[4]250...
✔	Pn085	通讯数据存储EEPROM选择	0x0	-	0~65535	0x1
✔	Pn204	电子齿轮比分子	0	-	0~1073741824	64
✔	Pn206	电子齿轮比分母	10000	-	1~1073741824	1
✔	Pn208	内部位置指令选择	[4]CanOpen		0~4	[0]位置...
✔	Pn791	编码器类型	[9]23位单圈绝...	-	1~18	[7]24...
✔	Pn880	Pr加减速时间0	50	ms	0~60000	100
✔	Pn898	Pr指令通讯触发	20000		0~65535	10000

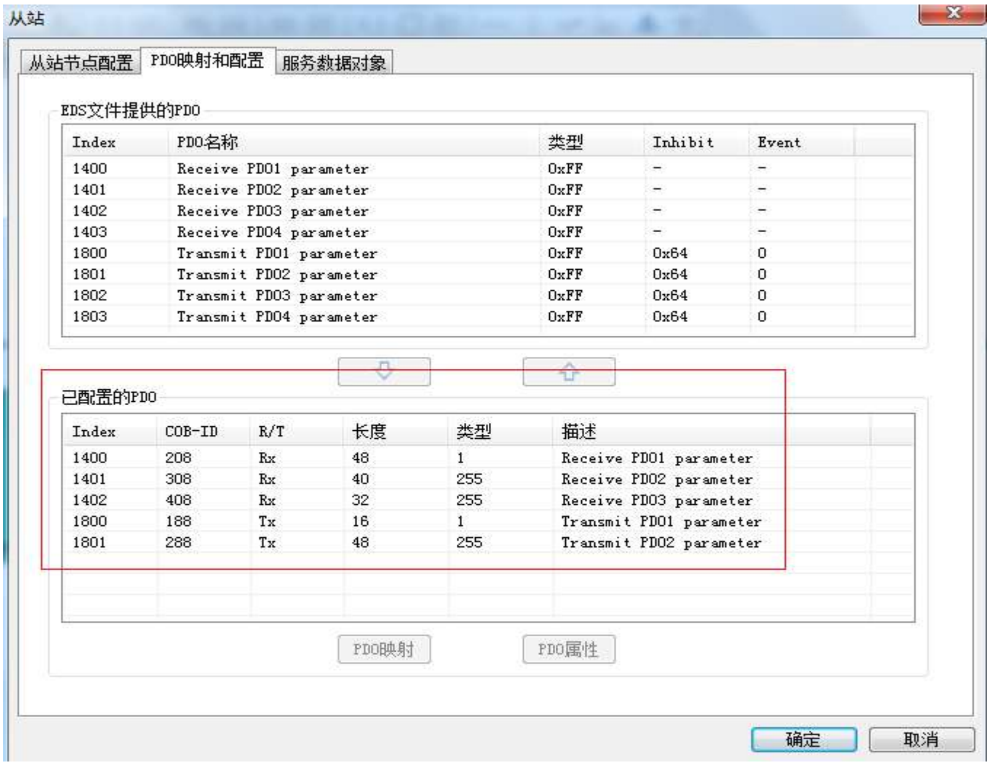
5、Write V5 control servo forward and reverse program:

(1) PDO property settings:

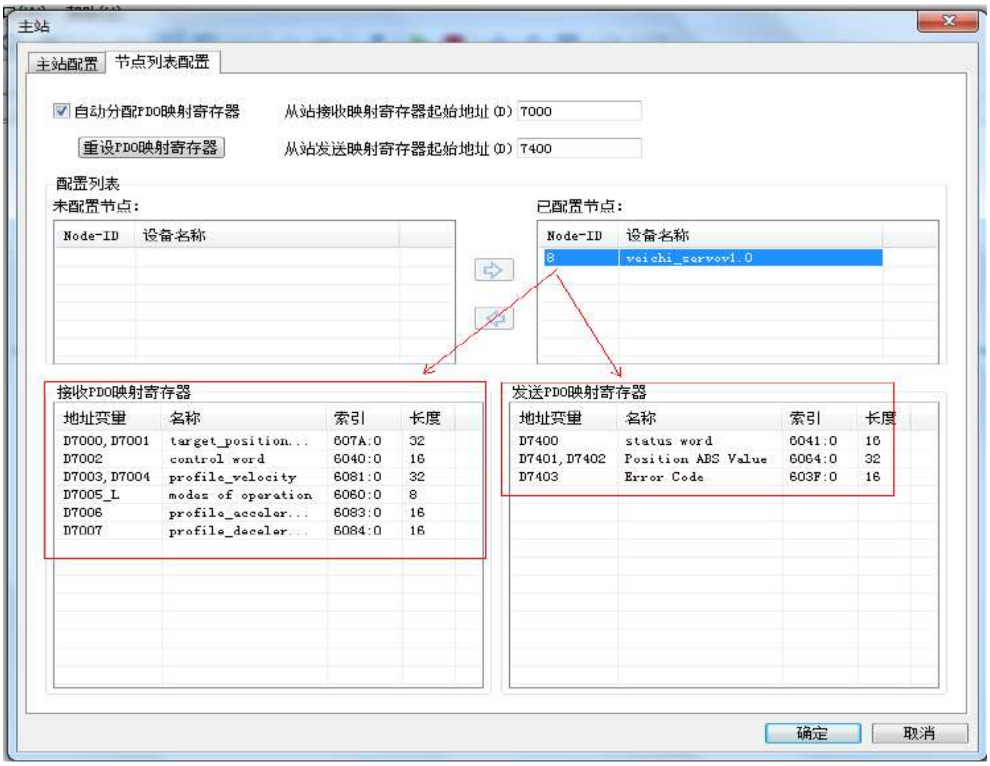
TPDO1 and RPDO1 are set to synchronous mode with a synchronization period of 4ms (the synchronization period or synchronization number can be set according to the number of slaves and the amount of communication data). The number of synchronizations is 1.

TPDO2 is set to synchronous mode with a sync number of 5. RPDO2 and RPDO3 are set to asynchronous mode, the suppression time and event time are both set to 0, and are sent as data changes.

TPDO2 is set to asynchronous mode with an event time of 10ms and a suppression time of 5ms.



（2）IO Mapping table:

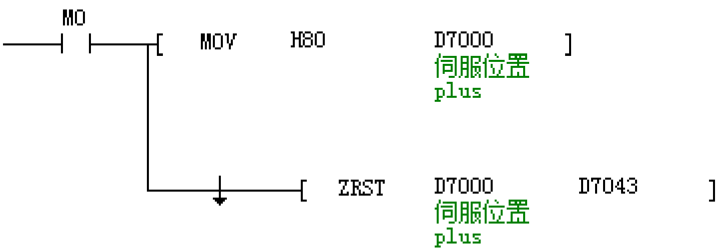


(3) Main program:

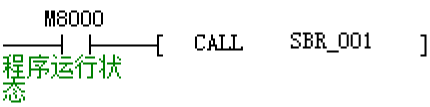
初始化伺服映射D寄存器

（因IO映射表中的D寄存器为掉电保持类，为防止上电误动作，上电将IO映射区D寄存器清零。）

复位伺服 对控制字D7000赋值H80。

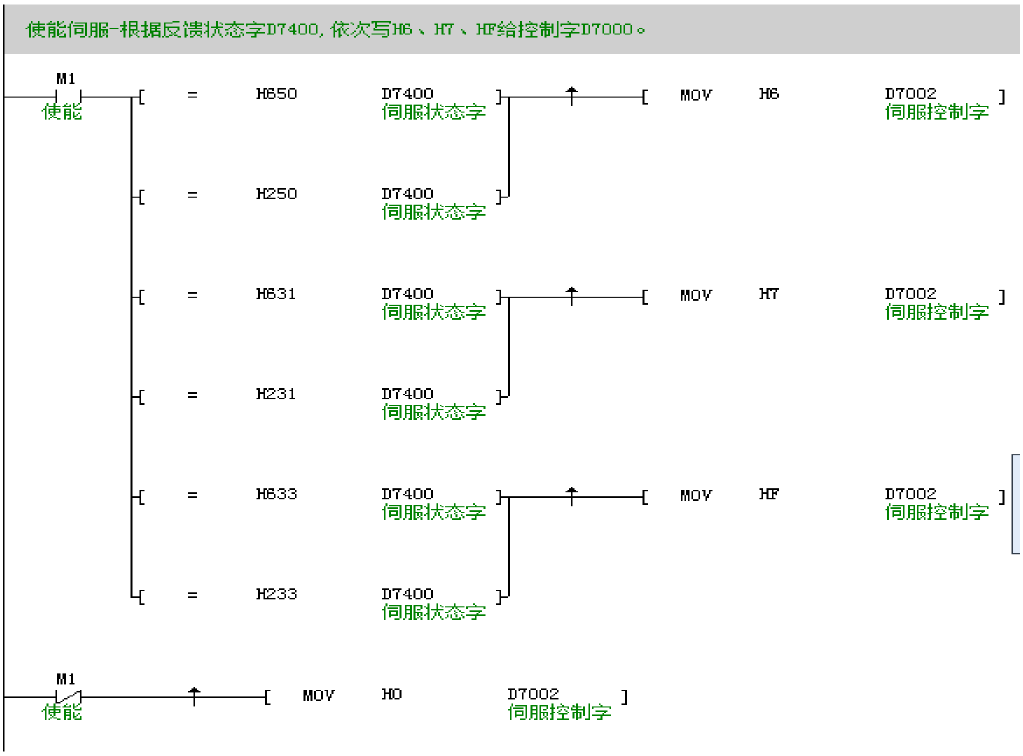


调用控制从站1子程序

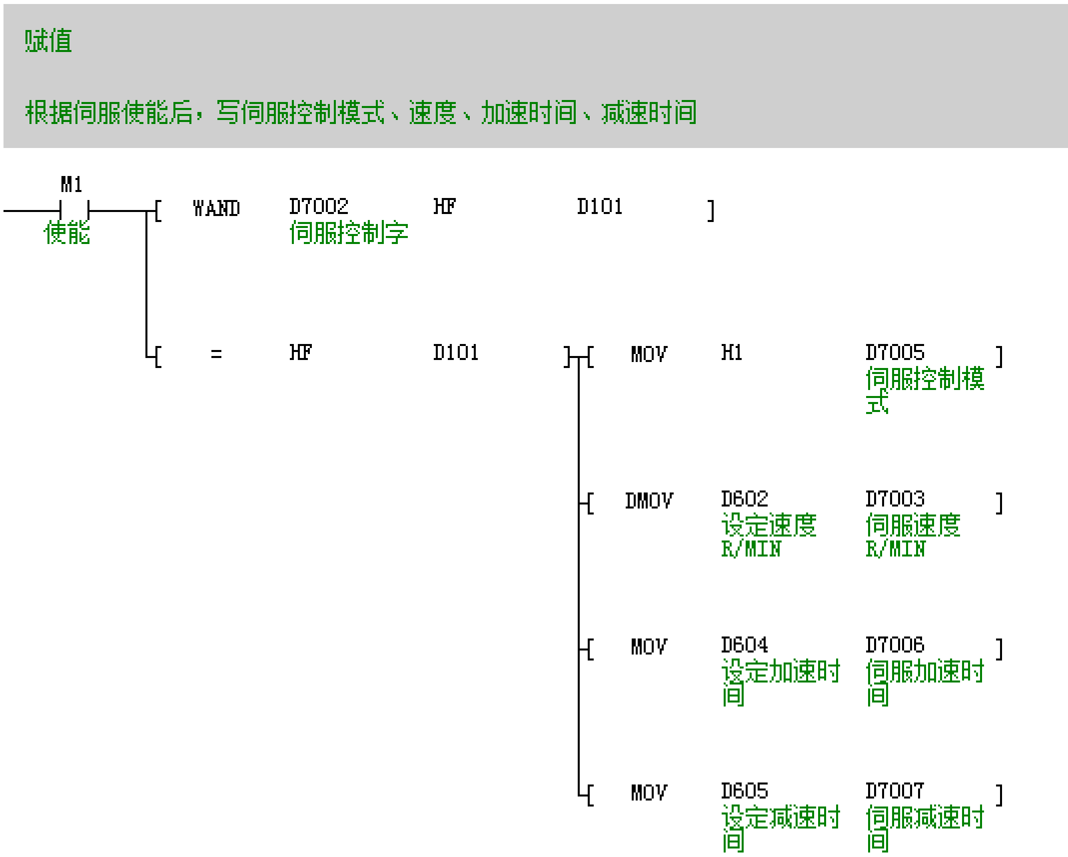


（4）子程序：

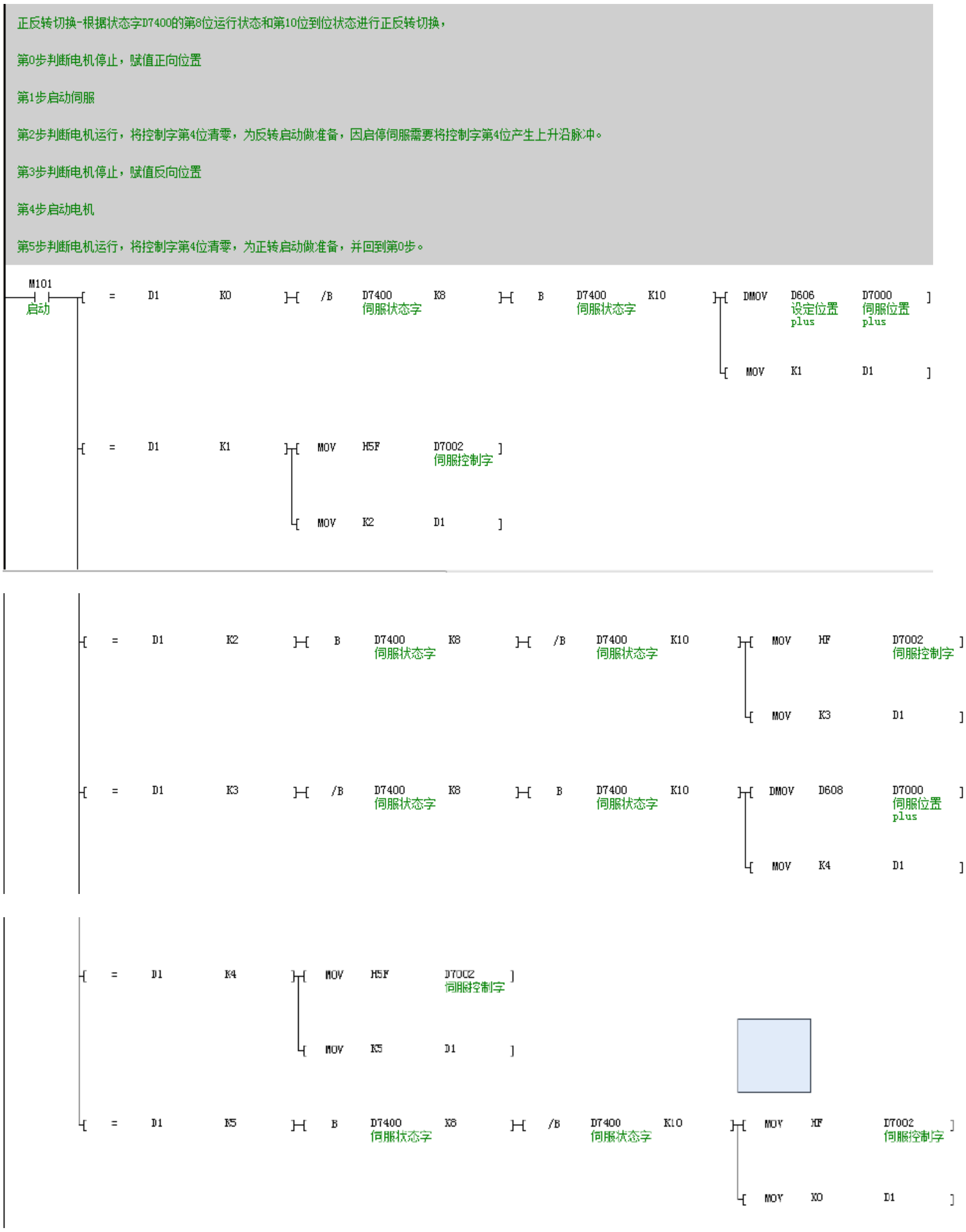
网络 1：



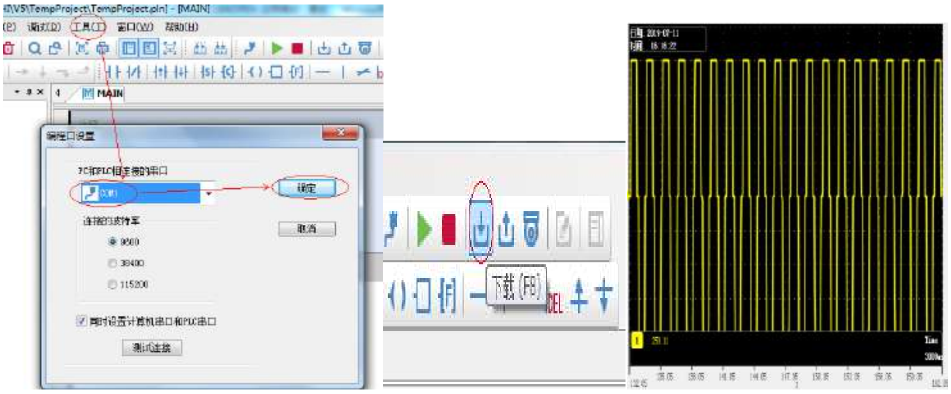
Network 2:



Network 3:



(5) Download the program and monitor the motor parameter command speed with the SD700 servo host computer software oscilloscope function:


MAIN
下载(F8)

Chapter 9 Subprogram

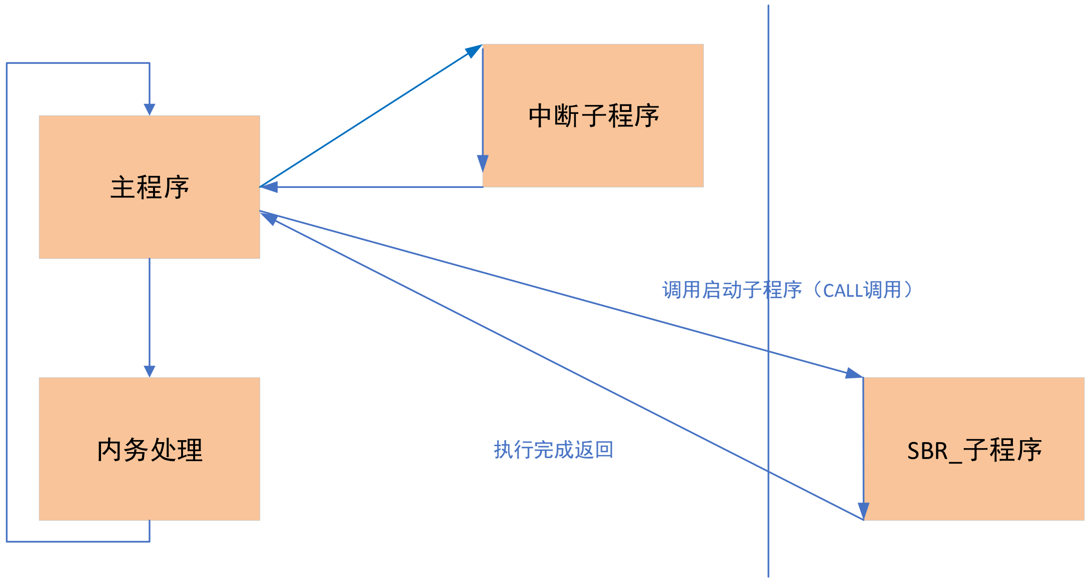
9.1 Summary

9.1.1 V5 subroutine overview

		Overview	
SBR	CALL instruction	There can be more than 64 subroutines in a project. The subprograms are mainly composed of main programs or other Subroutine calls. They mainly perform functions that are commonly used or need to be reused. Subroutines can only be written by ladder diagrams or instruction lists.	
P	CJ instruction	512 points, used with the LBL instruction	
I	Interrupt subroutine	External Interrupt	PG0-PG2 input interrupt, number I000, I001, I100, I101, I200, I201 (I000 indicates X0 falling edge interrupt, I001 indicates X0 rising edge interrupt, I100 indicates X1 falling edge interrupt, I101 indicates X1 rising edge interrupt, I200 indicates X2 Falling edge interrupt, I201 indicates X2 rising edge interrupt). After the interrupt disable flag register is turned ON, the corresponding input interrupt is disabled.
		Pulse completion interrupt	I502, I502, I504.
		Timed interrupt	I600 (I6 means timer interrupt 0, 00 means time base, time base range is 1 to 99), I700 (timed interrupt 1), I800 (timed interrupt 2).
		Count completion interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points (for DHSCS instructions).

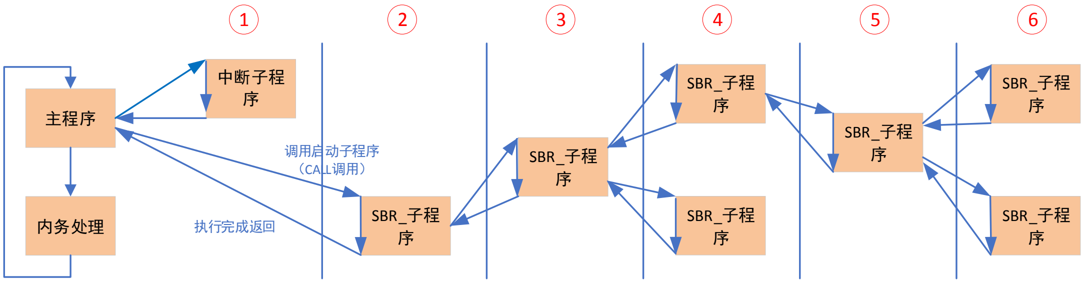
9.1.2 V5 Subroutine execution mechanism

Main program, subroutine execution logic, cyclic scan mode.



Subprogram nesting level

The subroutine is nested at most 6 layers. The main program calls the subroutine as the first layer. If it is not called once, it adds a nesting. If the nesting returns, the nesting level is not increased, as shown in the following figure.



9.2 General Subroutines Application

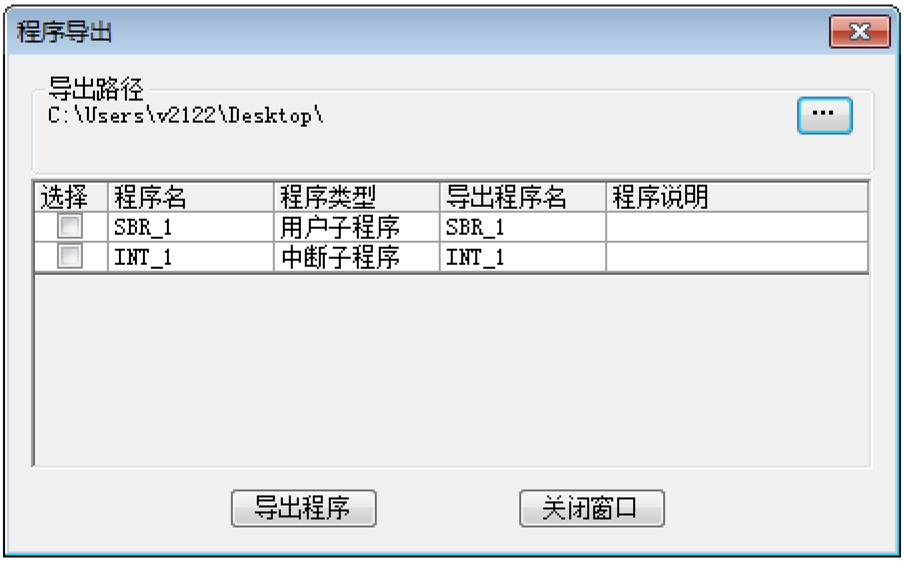
9.2.1 Creating a subroutine

Right click on the "Program Block" node of the "Project Manager" window, select "Insert Subprogram" or "Insert Interrupt Subprogram" to insert a new subroutine. The default name of the new subprogram name is SBR_* , new The default name of the interrupt subroutine name is INT_* (where * is a number automatically calculated by the software). After the build is complete, the default program name can be changed to a more meaningful name via the subroutine properties dialog. When a new program node is inserted in the project tree, the program is also opened and can be edited immediately.

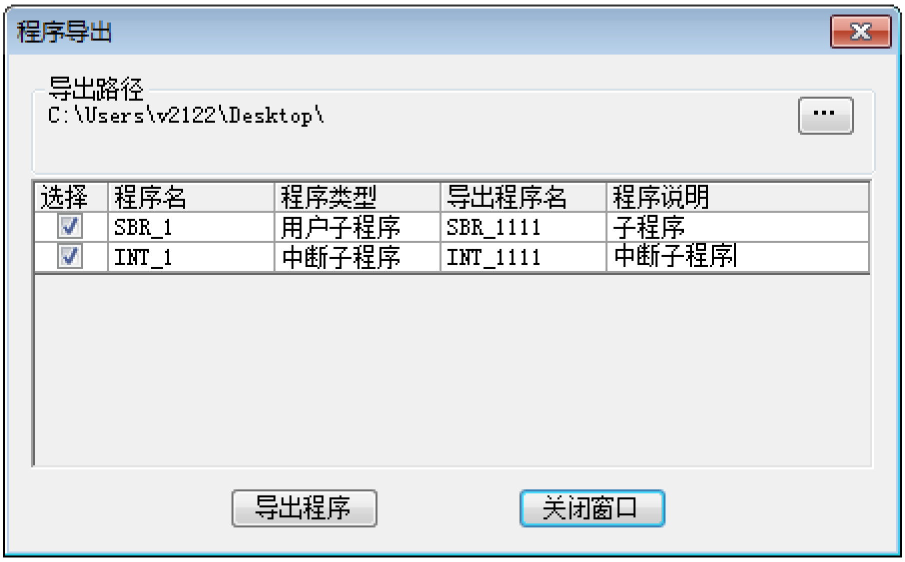
9.2.2 Export subroutine

The software provides the import and export functions of the sub-library. For a common function, it can be used again in different projects through the import / export function after writing in the sub- program, avoiding repeated code writing of the same function. VCAutoDesignsoft manages these programs by using the library files. The general subroutines can first be exported to a directory to form reusable library files. In the project that needs to use these subprograms, the import function can be used to transfer the library files. Imported into the current project as a standard subroutine.

Select the "Program File Guide" menu item under the "File" menu to open the program export dialog box. The dialog box automatically lists the programs that can be exported in the project. The first time the program export path is the software installation path. Under the \Lib\ directory, the user can click the "..." button to replace the path. The path selected by the user automatically becomes the default export / import path of the program.

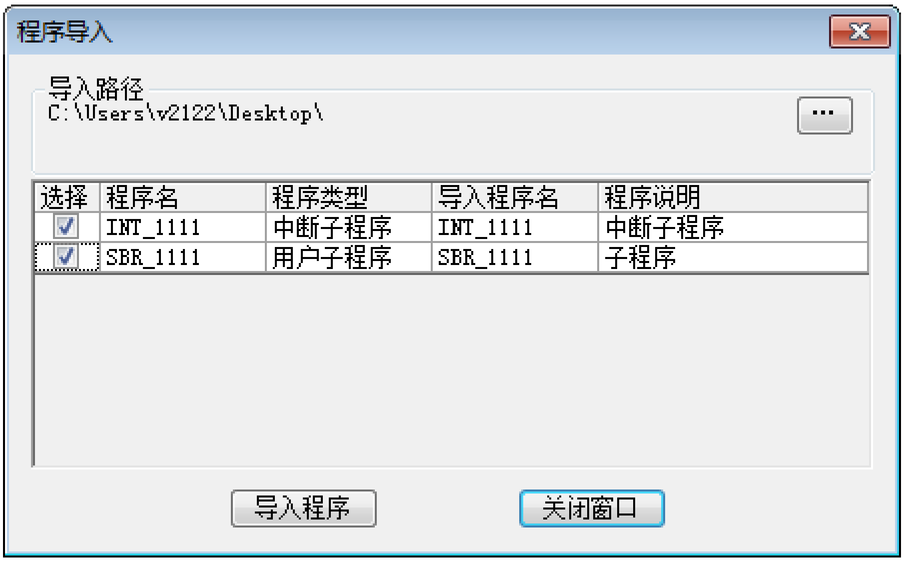


You can edit the program's export program name and program description , and finally tick the program you want to export, and then click the "Export Program" button. After the program is correctly exported, the hook before the program will disappear automatically, otherwise it will still be in the state before export.



9.2.3 Import subroutine

Select the "Program File Guide" menu item under the menu "File" to open the program import dialog: the dialog box automatically lists the program information that can be imported under the default import and export path. You can click the "..." button to change the path. Before importing, you can edit the name of the program after importing the project. When the import program name is invalid, the red color is underlined in the list to prompt. If the specified program name is the same as the existing program in the project, it will prompt whether to cover. After selecting the program, click the " Import Program " button and the program will be imported into the project.



9.2.4 Subroutine property

Select the subroutine node, then click the right mouse button to pop up the menu, select "Properties" to open the subroutine properties dialog box, in the subroutine properties dialog box you can modify the subroutine's program name, author , program description and encryption, as shown below Show:

Click "OK" to save, the subprogram name will be renamed with the newly entered "program name", as shown below:

9.2.5 Subroutine call

Subprogram call rules:

1. In the main program, you can nest subroutines (place subroutine call instructions in subroutines) with a maximum nesting depth of 5, and subprograms are not allowed in the interrupt program;

2. The explicit cyclic call is prohibited between the user programs, for example, the subprograms A and B call each other;

3. The user program prohibits recursive calls. For example, user program A calls program B, program B calls program C, program C calls program A, and forms a ring. In addition, the subprogram does not allow itself to be called.

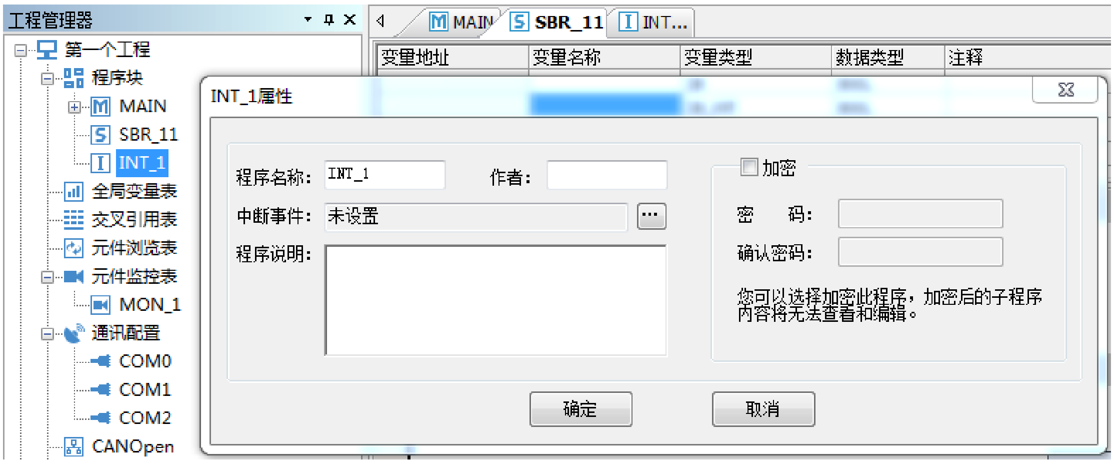
Other program units call subprograms: Subprograms can be called by the main program and other subprograms using the Call and Callp instructions.

Other program blocks are called in the subroutine: other subprograms, electronic cams, control interrupt subroutines, etc. can be called in the subroutine.

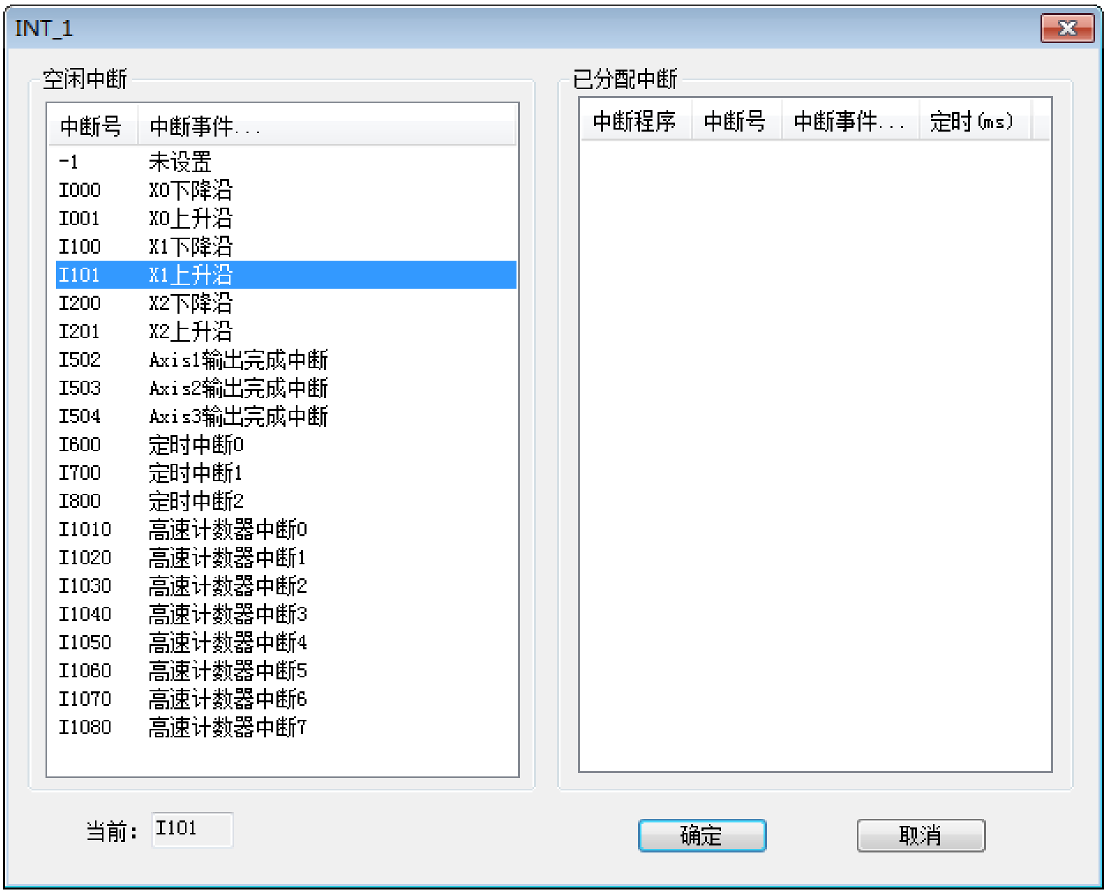
9.3interrupt subroutine application

9.3.1 Interrupt subroutine attribute

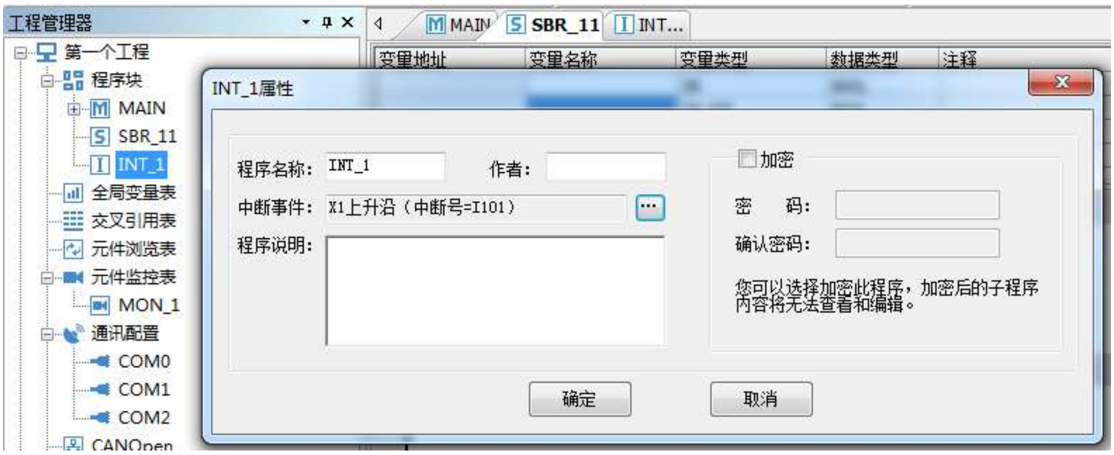
Select the interrupt subroutine node, then click the right mouse button to pop up the menu, select "Properties" to open the interrupt subroutine properties dialog box. For the interrupt subroutine, you can specify the interrupt number for it (the new interrupt subroutine default interrupt number is -1, indicating that it is not set), and the interrupt subroutine properties dialog box is as shown below:



Click the "..." button, the interrupt allocation dialog box will pop up, where the currently available idle interrupts and the interrupts that have been used by other interrupt subroutines are displayed. You can select an interrupt number in the idle interrupt to assign to the current interrupt subroutine, as follows The figure shows:



After selecting the interrupt number in the interrupt setting window, click "OK" to return to the interrupt property window. The newly set interrupt is as shown below:



Interrupt Properties Window Click the "OK" button to complete the interrupt settings. The Interrupt Subprogram Properties window allows you to set the program name, author, program description, and encryption.

9.3.2 Interrupt Subroutine call

1, interrupt description

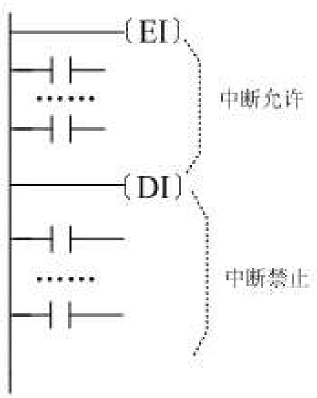
The interrupt subroutine is executed immediately after being triggered by the interrupt function, and is not affected by the user program scan cycle. In the general sequence program processing, the delay caused by the operation cycle and the time deviation affect the mechanical action, in the interrupt subroutine. Can be improved.

The types of interrupts supported by the V5-MC104 are shown in the following table:

Interrupt type	Interrupt number
External Interrupt	PG0-PG2 input interrupt, number I000, I001, I100, I101, I200, I201 (I000 indicates X0 falling edge interrupt, I001 indicates X0 rising edge interrupt, I100 indicates X1 falling edge interrupt, I101 indicates X1 rising edge interrupt, I200 indicates X2 Falling edge interrupt, I201 indicates X2 rising edge interrupt). After the interrupt disable flag register is turned ON, the corresponding input interrupt is disabled.
Pulse completion interrupt	I502, I502, I504.
Timed interrupt	I600 (I6 means timer interrupt 0, 00 means time base, time base range is 1 to 99), I700 (timed interrupt 1), I800 (timed interrupt 2).
Count completion interrupt	I010, I020, I030, I040, I050, I060, I070, I080, 8 points (for DHSCS instructions).

2, open, disable interrupt

When the PLC program starts running, the default is the interrupt disable state; after the EI statement is executed, the interrupt function is allowed; when the interrupt is enabled, after the DI statement is executed, the interrupt disable state is entered. If there is no interrupt in the program to insert the prohibited interval, you can not use the DI command.



Appendix I Special Device Assignment Instructions

SM flag bit allocation

Axis 1	Axis 2	Axis 3	Axis 4	Y0	Y1	Description
SM0	SM100	SM200	SM700	SM800	SM900	Motion control
SM1	SM101	SM201	SM701	SM801	SM901	Motion control
SM2	SM102	SM202	SM702	SM802	SM902	Motion control
SM3	SM103	SM203	SM703	SM803	SM903	Motion control
SM4	SM104	SM204	SM704	SM804	SM904	Motion control
SM5	SM105	SM205	SM705	SM805	SM905	Motion control
SM6	SM106	SM206	SM706	SM806	SM906	Motion control
SM7	SM107	SM207	SM707	SM807	SM907	Motion control
SM8	SM108	SM208	SM708	SM808	SM908	Motion control
SM9	SM109	SM209	SM709	SM809	SM909	Motion control
SM10	SM110	SM210	SM710	SM810	SM910	Motion control
SM11	SM111	SM211	SM711	SM811	SM911	Motion control
SM12	SM112	SM212	SM712	SM812	SM912	DRVZ origin return direction
SM13	SM113	SM213	SM713	SM813	SM913	ZRN origin signal specification
SM14	SM114	SM214	SM714	SM814	SM914	Motion control
SM15	SM115	SM215	SM715	SM815	SM915	Motion control
SM16	SM116	SM216	SM716	SM816	SM916	Motion control
SM17	SM117	SM217	SM717	SM817	SM917	G01S curve enable
SM18	SM118	SM218	SM718	SM818	SM918	DRVZ instruction enable
SM19	SM119	SM219	SM719	SM819	SM919	Reverse compensation enable flag
SM20	SM120	SM220	SM720	SM820	SM920	Continuous interpolation enable flag
SM21	SM121	SM221	SM721	SM821	SM921	Motion control
SM22	SM122	SM222	SM722	SM822	SM922	Motion overlay status flag
SM23	SM123	SM223	SM723	SM823	SM923	Motion control
SM24	SM124	SM224	SM724	SM824	SM924	Motion control
SM25	SM125	SM225	SM725	SM825	SM925	Motion control
SM26	SM126	SM226	SM726	SM826	SM926	Motion control
SM27	SM127	SM227	SM727	SM827	SM927	Motion control
SM28	SM128	SM228	SM728	SM828	SM928	Motion control
SM29	SM129	SM229	SM729	SM829	SM929	Motion control
SM30	SM130	SM230	SM730	SM830	SM930	Motion control
SM31	SM131	SM231	SM731	SM831	SM931	Motion control
SM32	SM132	SM232	SM732	SM832	SM932	Motion control
SM33	SM133	SM233	SM733	SM833	SM933	Motion control
SM34	SM134	SM234	SM734	SM834	SM934	Motion control
SM35	SM135	SM235	SM735	SM835	SM935	Motion control

SM36	SM136	SM236	SM736	SM836	SM936	Motion control
SM37	SM137	SM237	SM737	SM837	SM937	Motion control
SM38	SM138	SM238	SM738	SM838	SM938	Motion control
SM39	SM139	SM239	SM739	SM839	SM939	Motion control
SM40	SM140	SM240	SM740	SM840	SM940	Motion control
SM41	SM141	SM241	SM741	SM841	SM941	Motion control
SM42	SM142	SM242	SM742	SM842	SM942	Motion control
SM43	SM143	SM243	SM743	SM843	SM943	Motion control
SM44	SM144	SM244	SM744	SM844	SM944	Motion control
SM45	SM145	SM245	SM745	SM845	SM945	Motion control
SM46	SM146	SM246	SM746	SM846	SM946	Motion control
SM47	SM147	SM247	SM747	SM847	SM947	Motion control
SM48	SM148	SM248	SM748	SM848	SM948	Motion control
SM49	SM149	SM249	SM749	SM849	SM949	Motion control
SM50	SM150	SM250	SM750	SM850	SM950	Motion control
SM51	SM151	SM251	SM751	SM851	SM951	Motion control
SM52	SM152	SM252	SM752	SM852	SM952	Motion control
SM53	SM153	SM253	SM753	SM853	SM953	Motion control
SM54	SM154	SM254	SM754	SM854	SM954	Motion control
SM55	SM155	SM255	SM755	SM855	SM955	Motion control
SM56	SM156	SM256	SM756	SM856	SM956	Motion control
SM57	SM157	SM257	SM757	SM857	SM957	Motion control
SM58	SM158	SM258	SM758	SM858	SM958	Motion control
SM59	SM159	SM259	SM759	SM859	SM959	Motion control
SM60	SM160	SM260	SM760	SM860	SM960	Electronic cam terminal start
SM61	SM161	SM261	SM761	SM861	SM961	Motion control
SM62	SM162	SM262	SM762	SM862	SM962	Motion control
SM63	SM163	SM263	SM763	SM863	SM963	Motion control
SM64	SM164	SM264	SM764	SM864	SM964	Motion control
SM65	SM165	SM265	SM765	SM865	SM965	Motion control
SM66	SM166	SM266	SM766	SM866	SM966	Motion control
SM67	SM167	SM267	SM767	SM867	SM967	Motion control
SM68	SM168	SM268	SM768	SM868	SM968	Electronic cam enable starting position
SM69	SM169	SM269	SM769	SM869	SM969	Motion control
SM70	SM170	SM270	SM770	SM870	SM970	Electronic cam trigger mode selection: OFF: Software trigger ON: hardware trigger
SM71	SM171	SM271	SM771	SM871	SM971	Electronic cam input source selection OFF: Internal virtual connection ON: external input
SM72	SM172	SM272	SM772	SM872	SM972	Speed ratio calculation enable
SM73	SM173	SM273	SM773	SM873	SM973	Cycle enable
SM74	SM174	SM274	SM774	SM874	SM974	Motion control
SM75	SM175	SM275	SM775	SM875	SM975	Electronic cam delay start
SM76	SM176	SM276	SM776	SM876	SM976	Motion control

SM77	SM177	SM277	SM777	SM877	SM977	Motion control
SM78	SM178	SM278	SM778	SM878	SM978	Start electronic cam
SM79	SM179	SM279	SM779	SM879	SM979	Cam cycle completion flag
SM80	SM180	SM280	SM780	SM880	SM980	Stop sign
SM81	SM181	SM281	SM781	SM881	SM981	Stop mode selection
SM82	SM182	SM282	SM782	SM882	SM982	Key point Modify completion flag
SM83	SM183	SM283	SM783	SM883	SM983	Key point Modify mode
SM84	SM184	SM284	SM784	SM884	SM984	Motion control
SM85	SM185	SM285	SM785	SM885	SM985	Motion control
SM86	SM186	SM286	SM786	SM886	SM986	Motion control
SM87	SM187	SM287	SM787	SM887	SM987	Motion control
SM88	SM188	SM288	SM788	SM888	SM988	Motion control
SM89	SM189	SM289	SM789	SM889	SM989	Cam running flag / forced stop
SM90	SM190	SM290	SM790	SM890	SM990	Motion control subroutine enable
SM91	SM191	SM291	SM791	SM891	SM991	Motion control subroutine execution completed
SM92	SM192	SM292	SM792	SM892	SM992	Motion control
SM93	SM193	SM293	SM793	SM893	SM993	Motion control
SM94	SM194	SM294	SM794	SM894	SM994	Motion control
SM95	SM195	SM295	SM795	SM895	SM995	Motion control
SM96	SM196	SM296	SM796	SM896	SM996	Motion control
SM97	SM197	SM297	SM797	SM897	SM997	Motion control
SM98	SM198	SM298	SM798	SM898	SM998	Motion control
SM99	SM199	SM299	SM799	SM899	SM999	Motion control

SD register allocation

Axis 1	Axis 2	Axis 3	Axis 4	Y0	Y1	Description
SD0	SD100	SD200	SD700	SD800	SD900	Positive limit port number
SD1	SD101	SD201	SD701	SD801	SD901	Negative limit port number
SD2	SD102	SD202	SD702	SD802	SD902	Motion control
SD3	SD103	SD203	SD703	SD803	SD903	Motion control
SD4	SD104	SD204	SD704	SD804	SD904	Motion control
SD5	SD105	SD205	SD705	SD805	SD905	Motion control
SD6	SD106	SD206	SD706	SD806	SD906	Number of pulses required to rotate the motor one revolution
SD7	SD107	SD207	SD707	SD807	SD907	Number of pulses required to rotate the motor one revolution
SD8	SD108	SD208	SD708	SD808	SD908	Motor rotation one revolution distance
SD9	SD109	SD209	SD709	SD809	SD909	Motor rotation one revolution distance
SD10	SD110	SD210	SD710	SD810	SD910	G00 maximum speed
SD11	SD111	SD211	SD711	SD811	SD911	G00 maximum speed
SD12	SD112	SD212	SD712	SD812	SD912	G00 base speed
SD13	SD113	SD213	SD713	SD813	SD913	G00 axis base speed
SD14	SD114	SD214	SD714	SD814	SD914	Motion control
SD15	SD115	SD215	SD715	SD815	SD915	Motion control
SD16	SD116	SD216	SD716	SD816	SD916	Motion control

SD17	SD117	SD217	SD717	SD817	SD917	Motion control
SD18	SD118	SD218	SD718	SD818	SD918	Motion control
SD19	SD119	SD219	SD719	SD819	SD919	Motion control
SD20	SD120	SD220	SD720	SD820	SD920	G00 acceleration time
SD21	SD121	SD221	SD721	SD821	SD921	G00 deceleration time
SD22	SD122	SD222	SD722	SD822	SD922	Motion control
SD23	SD123	SD223	SD723	SD823	SD923	Motion control
SD24	SD124	SD224	SD724	SD824	SD924	Motion control
SD25	SD125	SD225	SD725	SD825	SD925	Motion control
SD26	SD126	SD226	SD726	SD826	SD926	Electrical zero position
SD27	SD127	SD227	SD727	SD827	SD927	Electrical zero position
SD28	SD128	SD228	SD728	SD828	SD928	Target position I
SD29	SD129	SD229	SD729	SD829	SD929	Target position I
SD30	SD130	SD230	SD730	SD830	SD930	Operating speed I
SD31	SD131	SD231	SD731	SD831	SD931	Operating speed I
SD32	SD132	SD232	SD732	SD832	SD932	Target position II
SD33	SD133	SD233	SD733	SD833	SD933	Target position II
SD34	SD134	SD234	SD734	SD834	SD934	Operating speed II
SD35	SD135	SD235	SD735	SD835	SD935	Operating speed II
SD36	SD136	SD236	SD736	SD836	SD936	Current position (PLS)
SD37	SD137	SD237	SD737	SD837	SD937	Current position (PLS)
SD38	SD138	SD238	SD738	SD838	SD938	Current speed (PPS)
SD39	SD139	SD239	SD739	SD839	SD939	Current speed (PPS)
SD40	SD140	SD240	SD740	SD840	SD940	Current position (mm)
SD41	SD141	SD241	SD741	SD841	SD941	Current position (mm)
SD42	SD142	SD242	SD742	SD842	SD942	Current speed (mm/min)
SD43	SD143	SD243	SD743	SD843	SD943	Current speed (mm/min)
SD44	SD144	SD244	SD744	SD844	SD944	Electronic gear molecule
SD45	SD145	SD245	SD745	SD845	SD945	Electronic gear denominator
SD46	SD146	SD246	SD746	SD846	SD946	Current input frequency
SD47	SD147	SD247	SD747	SD847	SD947	Current input frequency
SD48	SD148	SD248	SD748	SD848	SD948	Electronic cam cumulative input pulse number
SD49	SD149	SD249	SD749	SD849	SD949	Electronic cam cumulative input pulse number
SD50	SD150	SD250	SD750	SD850	SD950	Offset compensation value
SD51	SD151	SD251	SD751	SD851	SD951	Offset compensation value
SD52	SD152	SD252	SD752	SD852	SD952	Center coordinate offset compensation value
SD53	SD153	SD253	SD753	SD853	SD953	Center coordinate offset compensation value
SD54	SD154	SD254	SD754	SD854	SD954	Motion control
SD55	SD155	SD255	SD755	SD855	SD955	Motion control
SD56	SD156	SD256	SD756	SD856	SD956	Pulse input value count
SD57	SD157	SD257	SD757	SD857	SD957	Pulse input value count
SD58	SD158	SD258	SD758	SD858	SD958	Motion control
SD59	SD159	SD259	SD759	SD859	SD959	Motion control

SD60	SD160	SD260	SD760	SD860	SD960	Motion control
SD61	SD161	SD261	SD761	SD861	SD961	Pulse output setting
SD62	SD162	SD262	SD762	SD862	SD962	Motion control
SD63	SD163	SD263	SD763	SD863	SD963	Motion control
SD64	SD164	SD264	SD764	SD864	SD964	Motion control
SD65	SD165	SD265	SD765	SD865	SD965	Zero return waiting time, default 1000 (ms)
SD66	SD166	SD266	SD766	SD866	SD966	Motion control
SD67	SD167	SD267	SD767	SD867	SD967	Motion control
SD68	SD168	SD268	SD768	SD868	SD968	Electronic cam enable starting position
SD69	SD169	SD269	SD769	SD869	SD969	Electronic cam enable starting position
SD70	SD170	SD270	SD770	SD870	SD970	Cam table setting
SD71	SD171	SD271	SD771	SD871	SD971	Cam input shaft number setting
SD72	SD172	SD272	SD772	SD872	SD972	Number of aperiodic executions
SD73	SD173	SD273	SD773	SD873	SD973	Cam scaling from the axis
SD74	SD174	SD274	SD774	SD874	SD974	Maximum speed ratio (floating point)
SD75	SD175	SD275	SD775	SD875	SD975	Maximum speed ratio (floating point)
SD76	SD176	SD276	SD776	SD876	SD976	Minimum speed ratio (floating point number)
SD77	SD177	SD277	SD777	SD877	SD977	Minimum speed ratio (floating point number)
SD78	SD178	SD278	SD778	SD878	SD978	Delay start pulse number
SD79	SD179	SD279	SD779	SD879	SD979	Delay start pulse number
SD80	SD180	SD280	SD780	SD880	SD980	Motion control
SD81	SD181	SD281	SD781	SD881	SD981	Number of cycles the cam has executed
SD82	SD182	SD282	SD782	SD882	SD982	Number of cycles the cam has executed
SD83	SD183	SD283	SD783	SD883	SD983	Motion control
SD84	SD184	SD284	SD784	SD884	SD984	Electronic cam direction selection setting
SD85	SD185	SD285	SD785	SD885	SD985	Spindle cycle length
SD86	SD186	SD286	SD786	SD886	SD986	Spindle cycle length
SD87	SD187	SD287	SD787	SD887	SD987	Control cycle (default 500) (us)
SD88	SD188	SD288	SD788	SD888	SD988	Real-time control cycle time monitoring s (us)
SD89	SD189	SD289	SD789	SD889	SD989	Maximum control cycle time monitoring (us)
SD90	SD190	SD290	SD790	SD890	SD990	Motion control subroutine label setting
SD91	SD191	SD291	SD791	SD891	SD991	(internal use)
SD92	SD192	SD292	SD792	SD892	SD992	(internal use)
SD93	SD193	SD293	SD793	SD893	SD993	(internal use)
SD94	SD194	SD294	SD794	SD894	SD994	Motion control
SD95	SD195	SD295	SD795	SD895	SD995	Motion control
SD96	SD196	SD296	SD796	SD896	SD996	Motion control
SD97	SD197	SD297	SD797	SD897	SD997	Motion control
SD98	SD198	SD298	SD798	SD898	SD998	Motion control
SD99	SD199	SD299	SD799	SD899	SD999	Motion control

M8000 flag bit, D8000 register allocation

M component	Component function	D component	Component function
M8000	Program running status	D8000	Program watchdog timer
M8001	M8000 status is reversed	D8001	ARM version
M8002	The program starts running	D8002	Maximum user program capacity
M8003	M8002 status is reversed	D8003	Maximum user data capacity
M8004	Monitor PLC system errors	D8004	Wrong BCD value of M8060~M8067
M8005	System reservation	D8005	Current BCD battery voltage
M8006	System reservation	D8006	Detection value of battery voltage is too low
M8007	System reservation	D8007	Save M8007 actions
M8008	System reservation	D8008	AC power failure detection time (ms)
M8009	System reservation	D8009	Error display axis number
M8010	System reservation	D8010	Current scan time
M8011	10ms oscillating clock	D8011	Minimum scan time (0.1ms)
M8012	100ms oscillating clock	D8012	Maximum scan time (0.1ms)
M8013	1s oscillating clock	D8013	Real time clock seconds
M8014	1 minute oscillating clock	D8014	Real time clock
M8015	Clock stop and preset	D8015	Real time clock
M8016	Clock reading display stops	D8016	Real time clock day
M8017	±30 second correction	D8017	Real time clock month
M8018	System reservation	D8018	Real time clock year
M8019	Real-time clock RTC error	D8019	Real time clock week
M8020	Operation zero mark	D8020	X0~X7 universal input filter constant
M8021	Operation borrowing sign	D8021	X0~X7 high speed input filter constant
M8022	Operation carry flag	D8022	FPGA version number (low 16 bits)
M8023	System reservation	D8023	FPGA version number (high 16 bits)
M8024	Direction of the BMOV Directive	D8024	Library file version
M8025	System reservation	D8025	hardware version
M8026	System reservation	D8026	System reservation
M8027	System reservation	D8027	ARM temperature
M8028	System reservation	D8028	System reservation
M8029	Multi-cycle instruction execution completed	D8029	System reservation
M8030	System reservation	D8030	ADC0 read value
M8031	System reservation	D8031	ADC1 read value
M8032	System reservation	D8032	Low number of ladder executions

M8033	Device status unchanged	D8033	High number of ladder executions
M8034	All PLC outputs are OFF	D8034	ADC0 filter value
M8035	System reservation	D8035	Filter value of ADC1
M8036	System reservation	D8036	System reservation
M8037	System reservation	D8037	System reservation
M8038	System reservation	D8038	System reservation
M8039	System reservation	D8039	System reservation
M8040	System reservation	D8040	System reservation
M8041	System reservation	D8041	System reservation
M8042	System reservation	D8042	System reservation
M8043	System reservation	D8043	System reservation
M8044	System reservation	D8044	System reservation
M8045	System reservation	D8045	System reservation
M8046	System reservation	D8046	System reservation
M8047	System reservation	D8047	System reservation
M8048	System reservation	D8048	System reservation
M8049	Signal alarm is valid	D8049	System reservation
M8050	I00□ (X0) interrupt is prohibited	D8050	CPU unique id
M8051	I10□ (X1) interrupt prohibition	D8051	CPU unique id
M8052	I20□ (X2) interrupt prohibition	D8052	CPU unique id
M8053	I30□ (X3) interrupt prohibition	D8053	CPU unique id
M8054	I40□ (X4) interrupt prohibition	D8054	CPU unique id
M8055	System reservation	D8055	CPU unique id
M8056	Drive I6□□ (Timer 0) interrupt disable	D8056	CPU flashsize
M8057	Drive I7□□ (Timing 1) interrupt disable	D8057	CPU flashsize
M8058	Drive I8□□ (Timer 2) interrupt disable	D8058	CPU unique id check code
M8059	System reservation	D8059	AD sample value at power-on
M8060	I/O constitutes an error	D8060	I/O error start address number
M8061	PLC hardware error	D8061	PLC hardware error code serial number
M8062	PLC communication error	D8062	PLC communication error code
M8063	Online/communication error	D8063	Parallel online error code
M8064	Parameter error	D8064	Parameter error code
M8065	Grammatical errors	D8065	Syntax error code
M8066	Loop error	D8066	Loop error code
M8067	Operation error	D8067	Operation error code
M8068	Operation error latch	D8068	Latch the wrong step number of the program
M8069	System error flag	D8069	Wrong step number of M8065~M8067
M8070	System reservation	D8070	System reservation

M8071	System reservation	D8071	System reservation
M8072	System reservation	D8072	System reservation
M8073	System reservation	D8073	System reservation
M8074	System reservation	D8074	System reservation
M8075	System reservation	D8075	System reservation
M8076	System reservation	D8076	System reservation
M8077	System reservation	D8077	System reservation
M8078	System reservation	D8078	System reservation
M8079	System reservation	D8079	System reservation
M8080	System reservation	D8080	System reservation
M8081	System reservation	D8081	System reservation
M8082	System reservation	D8082	System reservation
M8083	System reservation	D8083	System reservation
M8084	System reservation	D8084	System reservation
M8085	System reservation	D8085	System reservation
M8086	System reservation	D8086	System reservation
M8087	System reservation	D8087	System reservation
M8088	System reservation	D8088	System reservation
M8089	System reservation	D8089	System reservation
M8090	System reservation	D8090	System reservation
M8091	System reservation	D8091	System reservation
M8092	System reservation	D8092	System reservation
M8093	System reservation	D8093	System reservation
M8094	System reservation	D8094	System reservation
M8095	System reservation	D8095	System reservation
M8096	System reservation	D8096	System reservation
M8097	System reservation	D8097	System reservation
M8098	System reservation	D8098	System reservation
M8099	System reservation	D8099	System reservation
M8100	System reservation	D8100	System reservation
M8101	System reservation	D8101	System reservation
M8102	System reservation	D8102	System reservation
M8103	System reservation	D8103	System reservation
M8104	System reservation	D8104	System reservation
M8105	System reservation	D8105	System reservation
M8106	System reservation	D8106	Internal occupancy
M8107	System reservation	D8107	Internal occupancy
M8108	System reservation	D8108	Internal occupancy
M8109	System reservation	D8109	Internal occupancy
M8110	System reservation	D8110	COM0 communication format setting
M8111	System reservation	D8111	COM0 communication station number setting

M8112	System reservation	D8112	COM0 communication format setting
M8113	System reservation	D8113	System reservation
M8114	System reservation	D8114	System reservation
M8115	System reservation	D8115	System reservation
M8116	System reservation	D8116	COM0 communication protocol setting
M8117	System reservation	D8117	System reservation
M8118	System reservation	D8118	Modbus communication error station number
M8119	System reservation	D8119	Communication timeout judgment (100ms)
M8120	System reservation	D8120	COM1 communication format setting
M8121	System reservation	D8121	COM1 communication station number setting
M8122	Modbus execution status / RS send flag	D8122	COM1 communication format setting / RS transmission remaining amount
M8123	System reservation	D8123	System reservation
M8124	System reservation	D8124	System reservation
M8125	System reservation	D8125	System reservation
M8126	System reservation	D8126	COM1 communication protocol setting
M8127	System reservation	D8127	System reservation
M8128	System reservation	D8128	System reservation
M8129	System reservation	D8129	COM1 communication timeout judgment (100ms)
M8130	System reservation	D8130	System reservation
M8131	System reservation	D8131	System reservation
M8132	System reservation	D8132	System reservation
M8133	System reservation	D8133	System reservation
M8134	System reservation	D8134	System reservation
M8135	System reservation	D8135	System reservation
M8136	System reservation	D8136	System reservation
M8137	System reservation	D8137	System reservation
M8138	System reservation	D8138	System reservation
M8139	System reservation	D8139	System reservation
M8140	System reservation	D8140	System reservation
M8141	System reservation	D8141	System reservation
M8142	System reservation	D8142	System reservation
M8143	System reservation	D8143	System reservation
M8144	System reservation	D8144	System reservation
M8145	System reservation	D8145	System reservation

M8146	System reservation	D8146	System reservation
M8147	System reservation	D8147	System reservation
M8148	System reservation	D8148	System reservation
M8149	System reservation	D8149	System reservation
M8150	Probe 1, enable	D8150	Probe 1, mode setting
M8151	Probe 1, state	D8151	Probe 1, low latch position
M8152	System reservation	D8152	Probe 1, high latch position
M8153	Probe 2, enable	D8153	Probe 2, mode setting
M8154	Probe 2, state	D8154	Probe 2, low latch position
M8155	System reservation	D8155	Probe 2, high latch position
M8156	Probe 3, enable	D8156	Probe 3, mode setting
M8157	Probe 3, state	D8157	Probe 3, low latch position
M8158	System reservation	D8158	Probe 3, high latch position
M8159	System reservation	D8159	System reservation
M8160	(XCH) SWAP function	D8160	System reservation
M8161	8/16 bit processing mode	D8161	System reservation
M8162	System reservation	D8162	Probe 1, latch phase deviation low
M8163	BINDA output character switching flag	D8163	Probe 1, latch phase deviation high
M8164	System reservation	D8164	Probe 2, latch phase deviation low
M8165	System reservation	D8165	Probe 2, latch phase deviation high
M8166	System reservation	D8166	Probe 3, latch phase deviation low
M8167	System reservation	D8167	Probe 3, latching phase deviation high
M8168	System reservation	D8168	System reservation
M8169	System reservation	D8169	System reservation
M8170	System reservation	D8170	System reservation
M8171	System reservation	D8171	System reservation
M8172	System reservation	D8172	System reservation
M8173	System reservation	D8173	System reservation
M8174	System reservation	D8174	System reservation
M8175	System reservation	D8175	System reservation
M8176	System reservation	D8176	System reservation
M8177	System reservation	D8177	System reservation
M8178	System reservation	D8178	System reservation
M8179	System reservation	D8179	System reservation
M8180	System reservation	D8180	System reservation
M8181	System reservation	D8181	System reservation
M8182	System reservation	D8182	Z1 register
M8183	System reservation	D8183	V1 register
M8184	System reservation	D8184	Z2 register
M8185	System reservation	D8185	V2 register
M8186	System reservation	D8186	Z3 register

M8187	System reservation	D8187	V3 register
M8188	System reservation	D8188	Z4 register
M8189	System reservation	D8189	V4 register
M8190	System reservation	D8190	Z5 register
M8191	System reservation	D8191	V5 register
M8192	System reservation	D8192	Z6 register
M8193	System reservation	D8193	V6 register
M8194	System reservation	D8194	Z7 register
M8195	System reservation	D8195	V7 register
M8196	System reservation	D8196	System reservation
M8197	System reservation	D8197	System reservation
M8198	System reservation	D8198	System reservation
M8199	System reservation	D8199	System reservation
M8200	System reservation	D8200	ARM secondary version
M8201	System reservation	D8201	System reservation
M8202	System reservation	D8202	System reservation
M8203	System reservation	D8203	System reservation
M8204	System reservation	D8204	System reservation
M8205	System reservation	D8205	System reservation
M8206	System reservation	D8206	System reservation
M8207	System reservation	D8207	System reservation
M8208	System reservation	D8208	System reservation
M8209	System reservation	D8209	System reservation
M8210	System reservation	D8210	System reservation
M8211	System reservation	D8211	System reservation
M8212	System reservation	D8212	System reservation
M8213	System reservation	D8213	System reservation
M8214	System reservation	D8214	System reservation
M8215	System reservation	D8215	System reservation
M8216	System reservation	D8216	System reservation
M8217	System reservation	D8217	System reservation
M8218	System reservation	D8218	System reservation
M8219	System reservation	D8219	System reservation
M8220	System reservation	D8220	System reservation
M8221	System reservation	D8221	System reservation
M8222	System reservation	D8222	System reservation
M8223	System reservation	D8223	System reservation
M8224	System reservation	D8224	System reservation
M8225	System reservation	D8225	System reservation
M8226	System reservation	D8226	System reservation
M8227	System reservation	D8227	System reservation
M8228	System reservation	D8228	System reservation
M8229	System reservation	D8229	System reservation

M8230	System reservation	D8230	System reservation
M8231	System reservation	D8231	System reservation
M8232	System reservation	D8232	System reservation
M8233	System reservation	D8233	System reservation
M8234	System reservation	D8234	System reservation
M8235	C235 high speed counter direction control	D8235	System reservation
M8236	C236 high speed counter direction control	D8236	System reservation
M8237	C237 high speed counter direction control	D8237	System reservation
M8238	System reservation	D8238	System reservation
M8239	System reservation	D8239	System reservation
M8240	System reservation	D8240	CAN function is occupied, not available
M8241	System reservation	D8241	CAN function is occupied, not available
M8242	System reservation	D8242	CAN function is occupied, not available
M8243	System reservation	D8243	CAN function is occupied, not available
M8244	System reservation	D8244	CAN function is occupied, not available
M8245	System reservation	D8245	CAN function is occupied, not available
M8246	System reservation	D8246	CAN function is occupied, not available
M8247	System reservation	D8247	CAN function is occupied, not available
M8248	System reservation	D8248	CAN function is occupied, not available
M8249	System reservation	D8249	CAN function is occupied, not available
M8250	System reservation	D8250	CAN function is occupied, not available
M8251	System reservation	D8251	CAN function is occupied, not available
M8252	System reservation	D8252	CAN function is occupied, not available
M8253	System reservation	D8253	CAN function is occupied, not available
M8254	System reservation	D8254	CAN function is occupied, not available
M8255	System reservation	D8255	CAN function is occupied, not

			available
M8256	System reservation	D8256	System reservation
M8257	System reservation	D8257	System reservation
M8258	System reservation	D8258	System reservation
M8259	System reservation	D8259	System reservation
M8260	Modbus master switch	D8260	COM2 communication format setting
M8261	System reservation	D8261	COM2 communication station number setting
M8262	com2-Modbus execution status/RS transmission flag	D8262	COM2 communication format setting / RS transmission remaining amount
M8263	System reservation	D8263	System reservation
M8264	System reservation	D8264	System reservation
M8265	System reservation	D8265	System reservation
M8266	System reservation	D8266	COM2 communication protocol setting
M8267	System reservation	D8267	System reservation
M8268	System reservation	D8268	System reservation
M8269	System reservation	D8269	COM2 communication timeout judgment (100ms)
M8270	System reservation	D8270	System reservation
M8271	System reservation	D8271	System reservation
M8272	System reservation	D8272	System reservation
M8273	System reservation	D8273	System reservation
M8274	System reservation	D8274	System reservation
M8275	System reservation	D8275	System reservation
M8276	System reservation	D8276	System reservation
M8277	System reservation	D8277	System reservation
M8278	System reservation	D8278	System reservation
M8279	System reservation	D8279	System reservation
M8280	CAN protocol switching flag	D8280	CAN effective protocol display
M8281	System reservation	D8281	System reservation
M8282	System reservation	D8282	CanLink heartbeat
M8283	Effective CAN online monitoring address	D8283	CAN online monitoring start address
M8284	Set the CAN address	D8284	CAN address setting / display address
M8285	Set baud rate	D8285	Baud rate display
M8286	System reservation	D8286	CANlink baud rate setting
M8287	System reservation	D8287	CANOpen configuration error station number
M8288	System reservation	D8288	CANOpen configuration error number

M8289	System reservation	D8289	CAN bus error
M8290	System reservation	D8290	CAN reception error
M8291	System reservation	D8291	System reservation
M8292	System reservation	D8292	System reservation
M8293	System reservation	D8293	System reservation
M8294	System reservation	D8294	System reservation
M8295	System reservation	D8295	System reservation
M8296	Device address error	D8296	System reservation
M8297	System reservation	D8297	System reservation
M8298	System reservation	D8298	System reservation
M8299	System reservation	D8299	System reservation
M8300	System reservation	D8300	System reservation
M8301	System reservation	D8301	System reservation
M8302	System reservation	D8302	System reservation
M8303	System reservation	D8303	System reservation
M8304	Zero mark	D8304	System reservation
M8305	System reservation	D8305	System reservation
M8306	Carry flag	D8306	System reservation
M8307	System reservation	D8307	System reservation
M8308	System reservation	D8308	System reservation
M8309	Carry flag	D8309	System reservation
M8310	System reservation	D8310	(RND) random number lower 16 bits
M8311	System reservation	D8311	(RND) random number is 16 bits high
M8312	Carry flag	D8312	System reservation
M8313	System reservation	D8313	System reservation
M8314	System reservation	D8314	System reservation
M8315	Carry flag	D8315	System reservation
M8316	System reservation	D8316	System reservation
M8317	System reservation	D8317	System reservation
M8318	Carry flag	D8318	System reservation
M8319	System reservation	D8319	System reservation
M8320	Carry flag	D8320	System reservation
M8321	System reservation	D8321	System reservation
M8322	System reservation	D8322	System reservation
M8323	Carry flag	D8323	System reservation
M8324	System reservation	D8324	System reservation
M8325	System reservation	D8325	System reservation
M8326	Carry flag	D8326	System reservation
M8327	System reservation	D8327	System reservation
M8328	System reservation	D8328	System reservation
M8329	Carry flag	D8329	System reservation

M8330	Carry flag	D8330	System reservation
M8331	System reservation	D8331	System reservation
M8332	System reservation	D8332	System reservation
M8333	Carry flag	D8333	System reservation
M8334	System reservation	D8334	System reservation
M8335	System reservation	D8335	System reservation
M8336	Carry flag	D8336	System reservation
M8337	System reservation	D8337	System reservation
M8338	System reservation	D8338	System reservation
M8339	Carry flag	D8339	System reservation
M8340	Monitoring in Y300 pulse output	D8340	Y300 current value register (PLS, lower 16 bits)
M8341	Y300 clear signal output is valid	D8341	Y300 current value register (PLS, high 16 bits)
M8342	Y300 origin return direction designation	D8342	Y300 maximum speed (Hz, low 16 bits)
M8343	Y300 forward limit	D8343	Y300 maximum speed (Hz, high 16 bits)
M8344	Y300 reverse limit	D8344	Y300 origin return speed (Hz, low 16 bits)
M8345	Y300 near-point signal logic inversion	D8345	Y300 origin return speed (Hz, high 16 bits)
M8346	Y300 zero signal logic inversion	D8346	Y300 crawling speed (Hz)
M8347	Y300S curve acceleration and deceleration enable	D8347	Y300 substrate speed (Hz)
M8348	Y300 keeps the current position after returning to zero	D8348	Y300 acceleration time (ms)
M8349	Y300 pulse output stop sign	D8349	Y300 deceleration time (ms)
M8350	Y300 acceleration/deceleration time setting and pulse change are valid	D8350	Y300 clear device number
M8351	System reservation	D8351	Y300S curve filter value
M8352	Y300 output completion interrupt enable	D8352	System reservation
M8353	Y300 acceleration time Modify enable	D8353	System reservation
M8354	Y300 abnormal end flag	D8354	System reservation
M8355	Y300PLSV2 command acceleration	D8355	System reservation
M8356	Y300PLSV2 command deceleration	D8356	Y300 current value real-time register (PLS, low 16 bits)
M8357	Y300 allows deceleration inversion	D8357	Y300 current value real-time register (PLS, high 16 bits)
M8358	System reservation	D8358	System reservation

M8359	System reservation	D8359	System reservation
M8360	Monitoring in Y304 pulse output	D8360	Y304 current value register (PLS, low 16 bits)
M8361	Y304 clear signal output is valid	D8361	Y304 current value register (PLS, high 16 bits)
M8362	Y304 origin return direction designation	D8362	Y304 maximum speed (Hz, low 16 bits)
M8363	Y304 forward limit	D8363	Y304 maximum speed (Hz, high 16 bits)
M8364	Y304 reverse limit	D8364	Y304 origin return speed (Hz, low 16 bits)
M8365	Y304 near-point signal logic inversion	D8365	Y304 origin return speed (Hz, high 16 bits)
M8366	Y304 zero signal logic inversion	D8366	Y304 crawling speed (Hz)
M8367	Y304S curve acceleration and deceleration enable	D8367	Y304 substrate speed (Hz)
M8368	Y304 retains the current position after returning to zero	D8368	Y304 acceleration time (ms)
M8369	Y304 pulse output stop sign	D8369	Y304 deceleration time (ms)
M8370	Y304 acceleration/deceleration time setting and pulse change are valid	D8370	Y304 clear device number
M8371	System reservation	D8371	Y304S curve filter value
M8372	Y304 output completion interrupt enable	D8372	System reservation
M8373	Y304 acceleration time Modify enable	D8373	System reservation
M8374	Y304 abnormal end flag	D8374	System reservation
M8375	Y304PLSV2 command acceleration	D8375	System reservation
M8376	Y304PLSV2 command deceleration	D8376	Y304 current value real-time register (PLS, low 16 bits)
M8377	Y304 allows deceleration inversion	D8377	Y304 current value real-time register (PLS, high 16 bits)
M8378	System reservation	D8378	System reservation
M8379	System reservation	D8379	System reservation
M8380	Y310 pulse output monitoring	D8380	Y310 current value register (PLS, lower 16 bits)
M8381	Y310 clear signal output valid flag	D8381	Y310 current value register (PLS, high 16 bits)
M8382	Y310 origin return direction designation	D8382	Y310 maximum speed (Hz, low 16 bits)
M8383	Y310 forward limit	D8383	Y310 maximum speed (Hz, high 16 bits)

M8384	Y310 reverse limit	D8384	Y310 origin return speed (Hz, low 16 bits)
M8385	Y310 near-point signal logic inversion	D8385	Y310 origin return speed (Hz, high 16 bits)
M8386	Y310 zero signal logic inversion	D8386	Y310 crawling speed (Hz)
M8387	Y310S curve acceleration and deceleration enable	D8387	Y310 substrate speed (Hz)
M8388	Y310 keeps the current position after returning to zero	D8388	Y310 acceleration time (ms)
M8389	Y310 pulse output stop sign	D8389	Y310 deceleration time (ms)
M8390	Y310 acceleration/deceleration time setting and pulse change are valid	D8390	Y310 clear device number
M8391	System reservation	D8391	Y310S curve filter value
M8392	Y310 output completion interrupt enable	D8392	System reservation
M8393	Y310 acceleration time Modify enable	D8393	System reservation
M8394	Y310 abnormal end flag	D8394	System reservation
M8395	Y310PLSV2 instruction acceleration	D8395	System reservation
M8396	Y310PLSV2 command deceleration	D8396	Y310 current value real-time register (PLS, low 16 bits)
M8397	Y310 allows deceleration inversion	D8397	Y310 current value real-time register (PLS, high 16 bits)
M8398	System reservation	D8398	System reservation
M8399	System reservation	D8399	System reservation
M8400	Monitoring in Y314 pulse output	D8400	Y314 current value register (PLS, low 16 bits)
M8401	Y314 clear signal number output valid flag	D8401	Y314 current value register (PLS, high 16 bits)
M8402	Y314 origin return direction designation	D8402	Y314 maximum speed (Hz, low 16 bits)
M8403	Y314 forward limit	D8403	Y314 maximum speed (Hz, high 16 bits)
M8404	Y314 reverse limit	D8404	Y314 origin return speed (Hz, low 16 bits)
M8405	Y314 near-point signal logic inversion	D8405	Y314 origin return speed (Hz, high 16 bits)
M8406	Y314 zero signal logic inversion	D8406	Y314 crawling speed (Hz)
M8407	Y314S curve acceleration and deceleration enable	D8407	Y314 substrate speed (Hz)
M8408	Y314 keeps the mark of the current position after returning to	D8408	Y314 acceleration time (ms)

	zero		
M8409	Y314 pulse output stop sign	D8409	Y314 deceleration time (ms)
M8410	Y314 acceleration/deceleration time setting and pulse change are valid	D8410	Y314 clear device number
M8411	System reservation	D8411	Y314S curve filter value
M8412	Y314 output completion interrupt enable	D8412	System reservation
M8413	Y314 acceleration time Modify enable	D8413	System reservation
M8414	Y314 abnormal end flag	D8414	System reservation
M8415	Y314PLSV2 instruction acceleration	D8415	System reservation
M8416	Y314PLSV2 command deceleration	D8416	Y314 current value real-time register (PLS, low 16 bits)
M8417	Y314 allows deceleration inversion	D8417	Y314 current value real-time register (PLS, high 16 bits)
M8418	System reservation	D8418	System reservation
M8419	System reservation	D8419	System reservation
M8420	Monitoring in Y0 pulse output	D8420	Y0 current value register (PLS, lower 16 bits)
M8421	Y0 clear signal output valid flag	D8421	Y0 current value register (PLS, high 16 bits)
M8422	Y0 origin return direction specification	D8422	Y0 maximum speed (Hz, low 16 bits)
M8423	Y0 forward limit	D8423	Y0 maximum speed (Hz, high 16 bits)
M8424	Y0 reverse limit	D8424	Y0 origin return speed (Hz, low 16 bits)
M8425	Y0 near-point signal logic inversion	D8425	Y0 origin return speed (Hz, high 16 bits)
M8426	Y0 zero signal logic inversion	D8426	Y0 crawling speed (Hz)
M8427	Y0S curve acceleration and deceleration enable	D8427	Y0 base speed (Hz)
M8428	Y0 keeps the mark of the current position after returning to zero	D8428	Y0 acceleration time (ms)
M8429	Y0 pulse output stop sign	D8429	Y0 deceleration time (ms)
M8430	Y0 acceleration/deceleration time setting and pulse change are valid	D8430	Y0 clear device number
M8431	System reservation	D8431	Y0S curve filter value
M8432	Y0 output completion interrupt enable	D8432	System reservation
M8433	Y0 acceleration time Modify	D8433	System reservation

	enable		
M8434	Y0 abnormal end flag	D8434	System reservation
M8435	Y0PLSV2 instruction acceleration	D8435	System reservation
M8436	Y0PLSV2 command deceleration	D8436	Y0 current value real-time register (PLS, low 16 bits)
M8437	Y0 allows deceleration inversion	D8437	Y0 current value real-time register (PLS, high 16 bits)
M8438	System reservation	D8438	System reservation
M8439	System reservation	D8439	System reservation
M8440	Monitoring in Y1 pulse output	D8440	Y1 current value register (PLS, lower 16 bits)
M8441	Y1 clear signal output is valid	D8441	Y1 current value register (PLS, high 16 bits)
M8442	Y1 origin return direction specification	D8442	Y1 maximum speed (Hz, low 16 bits)
M8443	Y1 forward limit	D8443	Y1 maximum speed (Hz, high 16 bits)
M8444	Y1 reversal limit	D8444	Y1 origin return speed (Hz, low 16 bits)
M8445	Y1 near-point signal logic inversion	D8445	Y1 origin return speed (Hz, high 16 bits)
M8446	Y1 zero signal logic inversion	D8446	Y1 crawling speed (Hz)
M8447	Y1S curve acceleration and deceleration enable	D8447	Y1 substrate speed (Hz)
M8448	Y1 keeps the mark of the current position after returning to zero	D8448	Y1 acceleration time (ms)
M8449	Y1 pulse output stop sign	D8449	Y1 deceleration time (ms)
M8450	Y1 acceleration/deceleration time setting and pulse change are valid	D8450	Y1 clear device number
M8451	System reservation	D8451	Y1S curve filter value
M8452	Y1 output completion interrupt enable	D8452	System reservation
M8453	Y1 acceleration time Modify enable	D8453	System reservation
M8454	Y1 abnormal end flag	D8454	System reservation
M8455	Y1PLSV2 instruction acceleration	D8455	System reservation
M8456	Y1PLSV2 command deceleration	D8456	Y1 current value real-time register (PLS, low 16 bits)
M8457	Y1 allows deceleration inversion	D8457	Y1 current value real-time register (PLS, high 16 bits)
M8458	System reservation	D8458	System reservation

M8459	System reservation	D8459	System reservation
M8460	System reservation	D8460	System reservation
M8461	System reservation	D8461	System reservation
M8462	System reservation	D8462	System reservation
M8463	System reservation	D8463	System reservation
M8464	System reservation	D8464	System reservation
M8465	System reservation	D8465	System reservation
M8466	System reservation	D8466	System reservation
M8467	System reservation	D8467	System reservation
M8468	System reservation	D8468	System reservation
M8469	System reservation	D8469	System reservation
M8470	System reservation	D8470	System reservation
M8471	System reservation	D8471	System reservation
M8472	System reservation	D8472	System reservation
M8473	System reservation	D8473	System reservation
M8474	System reservation	D8474	System reservation
M8475	System reservation	D8475	System reservation
M8476	System reservation	D8476	System reservation
M8477	System reservation	D8477	System reservation
M8478	System reservation	D8478	System reservation
M8479	System reservation	D8479	System reservation
M8480	System reservation	D8480	System reservation
M8481	System reservation	D8481	System reservation
M8482	System reservation	D8482	System reservation
M8483	System reservation	D8483	System reservation
M8484	System reservation	D8484	System reservation
M8485	System reservation	D8485	System reservation
M8486	System reservation	D8486	System reservation
M8487	System reservation	D8487	System reservation
M8488	System reservation	D8488	System reservation
M8489	System reservation	D8489	System reservation
M8490	System reservation	D8490	System reservation
M8491	System reservation	D8491	System reservation
M8492	System reservation	D8492	System reservation
M8493	System reservation	D8493	System reservation
M8494	System reservation	D8494	System reservation
M8495	System reservation	D8495	System reservation
M8496	System reservation	D8496	System reservation
M8497	System reservation	D8497	System reservation
M8498	System reservation	D8498	System reservation
M8499	System reservation	D8499	System reservation
M8500	System reservation	D8500	Y0-Yn positioning command execution maximum speed (low 16

			bits)
M8501	System reservation	D8501	Y0-Yn positioning command execution maximum speed (high 16 bits)
M8502	System reservation	D8502	Y0-Yn positioning execution base speed
M8503	System reservation	D8503	Y0-Yn positioning acceleration and deceleration time
M8504	System reservation	D8504	System reservation
M8505	System reservation	D8505	System reservation
M8506	System reservation	D8506	System reservation
M8507	System reservation	D8507	System reservation
M8508	System reservation	D8508	System reservation
M8509	System reservation	D8509	System reservation
M8510	System reservation	D8510	System reservation
M8511	System reservation	D8511	System reservation

Appendix II System error code description

System error code D8061

error code	Error indicating content	Action when wrong
6101	Hardware SRAM failure	Stop running
6106	Hardware SPIFLASH failure	Stop running
6107	System IO setting error	Stop running
6108	FPGA loading failure	Stop running
6109	FPGA version failure	Stop running
6110	Hardware Ethernet error	Stop running
6119	Hardware capacitor failure	Stop running
16100	Hardware EEPROM failure	Stop running
16101	Hardware 422 failure	Stop running
16102	Hardware COM1 failure	Stop running
16103	Hardware COM2 failure	Stop running
16104	Hardware AD failure	Stop running
16105	Hardware clock failure	Stop running
16110	IO module search failed	Stop running
16111	IO module received ID number error three times in a row	Stop running
16112	The IO module master station receives the CRC check error three times in succession.	Stop running
16113	IO module gives three consecutive slaves an error	Stop running
16114	IO module dropped from the station	Stop running
16115	IO module slave type error	Stop running

System error code D8062

error code	Error indicating content	Action when wrong
16204	Ethernet response error	Keep running
16260	The mechanical unit setting value is incorrect (see D8009 for the specific axis error)	Keep running
16261	The electronic gear ratio setting value is incorrect (see D8009 for the specific axis error)	Keep running
16262	A cam table that is not configured in the background is used (see D8009 for specific axis errors)	Keep running
16263	The electronic cam does not select the external input spindle (see D8009 for specific axis errors)	Keep running
16264	The electronic cam slave axis speed is too large, exceeding the maximum allowable output speed (see D8009 for specific axis error)	Keep running
16265	The lower synchronization limit is greater than the upper synchronization limit (see D8009 for specific axis errors)	Keep running
16266	The key point of the spindle is illegal and the spindle position deviation is too large (see D8009 for the specific axis error)	Keep running
16267	Delay start pulse number setting is incorrect (see D8009 for specific axis error)	Keep running
16268	The key point of the cam is written to the command, and the key point value is illegal (see D8009 for the specific axis error)	Keep running
16269	The cam is encrypted and does not allow instructions to read keypoint data (see D8009 for specific axis errors)	Keep running
16270	Electronic cam scaling error from the axis (see D8009 for specific axis errors)	Keep running
16271	Electronic cam configuration unit error (see D8009 for specific axis error)	Keep running
16272	The Modify of the electronic cam is unsuccessful (see D8009 for specific axis errors)	Keep running
16273	Electronic cam Modify instructions are reused (see D8009 for specific axis errors)	Keep running
16274	The number of acyclic cycles exceeds the maximum value (see D8009 for specific axis errors)	Keep running
16275	Flying shear data is unreasonable (see D8009 for specific axis errors)	Keep running
16276	Flying shear data is unreasonable (see D8009 for specific axis errors)	
16278	The direction setting is unreasonable (see D8009 for specific axis errors)	
16279	Speed ratio calculation error (see D8009 for specific axis error)	
16281	The flyback return factor setting is unreasonable (see D8009 for specific axis errors)	Keep running

16282	The cutter movement range setting is unreasonable (see D8009 for specific axis errors)	Keep running
16283	The relationship between the cutter movement range, material length, and length of the synchronization zone is unreasonable (see D8009 for specific axis errors).	Keep running
16285	The linear type of the key point is wrong (see D8009 for the specific axis error)	Keep running
16286	The number of key points is incorrectly modified (see D8009 for specific axis errors)	Keep running
16287	Pole point data error (see D8009 for specific axis error)	Keep running

System error code D8063

error code	Error indicating content	Action when wrong
6320	422 communication address bit error	Keep running
6321	422 communication address word error	Keep running
6330	COM0_ MODBUS slave address setting error, the address is greater than 247 ;	Keep running
6331	COM0_ data frame length is wrong, the frame length does not meet the requirements, or the frame length is less than 5 ;	Keep running
6332	COM0_ address error, standard error frame; or inconsistent sending and receiving addresses;	Keep running
6333	COM0_CRC check error	Keep running
6334	COM0_ unsupported command code, standard error frame; or inconsistent send and receive commands; or unsupported commands;	Keep running
6335	COM0_ receive timeout	Keep running
6336	COM0_ data error, standard error frame;	Keep running
6337	COM0_ buffer overflow, no	Keep running
6338	COM0_frame error, standard error frame;	Keep running
6339	COM0_ serial port protocol error, when using modbus command or RS command, it is not defined as the corresponding protocol;	Keep running
6340	The COM1_MODBUS slave address is set incorrectly and the address is greater than 247.	Keep running
6341	The length of the COM1_ data frame is incorrect, the frame length does not meet the requirements, or the frame length is less than 5;	Keep running
6342	COM1_ address error, standard error frame; or inconsistent sending and receiving addresses;	Keep running
6343	COM1_CRC check error	Keep running
6344	COM1_ unsupported command code, standard error frame; or inconsistent send and receive commands; or unsupported commands;	Keep running
6345	COM1_ receive timeout	Keep running
6346	COM1_ data error, standard error frame;	Keep running

6347	COM1_ buffer overflow, no	Keep running
6348	COM1_frame error, standard error frame;	Keep running
6349	COM1_ serial port protocol error, when using modbus command or RS command, it is not defined as the corresponding protocol;	Keep running
6350	The COM2_MODBUS slave address is set incorrectly and the address is greater than 247.	Keep running
6351	The length of the COM2_ data frame is incorrect, the frame length does not meet the requirements, or the frame length is less than 5;	Keep running
6352	COM2_ address error, standard error frame; or inconsistent sending and receiving addresses;	Keep running
6353	COM2_CRC check error	Keep running
6354	COM2_ unsupported command code, standard error frame; or inconsistent send and receive commands; or unsupported commands;	Keep running
6355	COM2_ receive timeout	Keep running
6356	COM2_ data error, standard error frame;	Keep running
6357	COM2_ buffer overflow, no	Keep running
6358	COM2_frame error, standard error frame;	Keep running
6359	COM2_ serial port protocol error, when using modbus command or RS command, it is not defined as the corresponding protocol;	Keep running
6380	CAN download data error	Keep running
6381	CAN unknown error	Keep running
6382	CANOPEN send buffer overflow	Keep running
6383	CANOPEN receive buffer overflow	Keep running
6384	CAN general error	Keep running
6385	CAN passive error	Keep running
6386	CAN bus is off	Keep running
6387	CAN heartbeat error	Keep running
6388	CAN protocol error	Keep running
6389	CANPDO length error	Keep running
6390	CANRPDO timeout	Keep running
6391	CAN overload	Keep running
6392	CANPDO send and receive processing error	Keep running
6393	CANPDO transmission type error	Keep running
6394	CAN received the wrong message	Keep running
6395	CAN receives emergency message	Keep running
6396	The number of CAN slaves exceeds the limit	Keep running
6397	CANSDO returns the wrong command code	Keep running
6398	CAN download error	Keep running
6399	CAN writes wrong data	Keep running

System error code D8064

error code	Error indicating content	Action when wrong
6426	User program is incomplete	Stop running
16400	Gcode string error	Stop running
16401	Interpolation cycle maximum overflow	Stop running
16402	Set control cycle time error	Stop running
16403	System override setting error	Stop running
16404	The system is running	Stop running

System error code D8065

error code	Error indicating content	Action when wrong
6503	Instruction parameter error	Stop running
6504	Duplicate label definition	Stop running
6506	Use undefined instructions	Stop running
6507	Bad label P definition	Stop running
6508	Defective label I	Stop running
16500	Wrong interrupt ROM label	Stop running
16501	Subprogram ROM suffix	Stop running
16502	Ladder address error	Stop running
16503	Ladder instruction error	Stop running
16504	The number of pulse instructions exceeds the limit	Stop running
16505	The defined subroutine exceeds the maximum encoding	Stop running
16506	The defined T exceeds the maximum encoding	Stop running
16507	ZRST type error	Stop running
16508	ZRST invalid type	Stop running
16509	ZRST operand 1 number is greater than operand 2 number	Stop running

System error code D8066

error code	Error indicating content	Action when wrong
6627	Ladder diagram without RET instruction	Stop running
6630	Ladder diagram without SRET or IRET instructions	Stop running
6631	SRET is in an unusable location	Stop running
6632	RET is in an unusable location	Stop running
6633	IRET is in an unusable location	Stop running

System error code D8067

error code	Error indicating content	Action when wrong
6701	Jump label overrun	Keep running
6702	Called more than 6 times	Keep running

6705	VZ data calculation overflow	Keep running
6706	Unreasonable data or return overrun	Keep running
6720	CALL does not correspond to SRET	Keep running
6713	The positioning position is too small and does not satisfy the deceleration	Keep running
6732	Input filter parameters are incorrect	Keep running
6748	Input and output type error	Keep running
6749	Input and output number is wrong	Keep running
6760	High speed count exceeds limit	Keep running
6761	High-speed count output address error, component range is out of limits	Keep running
6762	High speed count output type error	Keep running
6763	High-speed counter input count port component exceeds range	Keep running
6770	High speed port conflict	Keep running
6771	Pulse command port error	Keep running
6772	MC instruction error	Keep running
6773	Call subroutine number of MC subroutine error	Keep running
6775	MC subroutine location error	Keep running
6776	Probe setting error	Keep running
6780	PID sampling period is less than or equal to 0	Keep running
6782	PID filter time error	Keep running
6783	PID scale factor error	Keep running
6784	PID integral coefficient error	Keep running
6785	PID differential coefficient error	Keep running
6786	PID lower limit is greater than upper limit	Keep running
6787	PID gain selection error	Keep running
16700	MC address error	Keep running
16701	MC subroutine, ROM checksum error	Keep running
16702	MC terminator error	Keep running
16703	MC status error	Keep running
16704	MC speed error	Keep running
16705	MC type error	Keep running
16706	MC axis is running	Keep running
16707	MC setting speed is illegal	Keep running
16740	VZ data D calculation overflow	Keep running
16741	VZ data T calculation overflow	Keep running
16742	VZ data C calculation overflow	Keep running
16743	VZ data C200 calculation overflow	Keep running
16744	VZ data SD calculation overflow	Keep running
16745	VZ data R calculation overflow	Keep running
16746	VZ data K calculation overflow	Keep running
16747	VZ data H calculation overflow	Keep running
16748	VZ data S calculation overflow	Keep running
16749	VZ data M calculation overflow	Keep running
16750	VZ data SM calculation overflow	Keep running

16751	VZ data X calculation overflow	Keep running
16752	VZ data Y calculation overflow	Keep running
16753	Unknown VZ type	Keep running
16754	32-bit instruction uses the wrong V	Keep running
16755	Unknown word component	Keep running
16756	Unknown device type	Keep running

深圳市伟创电气有限公司
地址：宝安区石岩街道塘头社区塘头 1 号路
领亚工业园春生楼三楼
电话：0755-36861688

苏州伟创电气设备技术有限公司
地址：吴中经济技术开发区淞葭路 1000 号
电话：0512-66171988

Version change record

Version	release time	Update content
A1.0	2019.08.28	第一次发布