VEDA MC

VEDA VC series

Programmable Controller Instruction and Programming Manual

VEDA VC Series Programmable Controller Instructions and Programming Manual

Table of Contents

Preface

- Target Audience

This book is suitable for automation technicians to help them master the programming, system design and debugging techniques of VEDA Programmable Logic Controller (PLC); it provides a reference for those who have preliminary and in-depth learning of PLC programming knowledge.

- Manual Content

This manual describes in detail the programming principles, software and hardware programming resources, supported programming languages and detailed instruction descriptions of VEDA VC series PLCs, as well as technical reference content such as high-speed input and output, communication, etc. Applications.

- Manual Layout

The chapters of this manual are arranged from the whole to the details. Each chapter has independent content. You can read through and gradually master the comprehensive content of the VEDA VC series PLC. You can also refer to the chapters at any time as some technical reference materials.

- Guide to Reading

1. Readers unfamiliar with PLC

For readers who are initially exposed to PLC, it is recommended to read Chapters 1 to 4 first. These chapters explain the basic knowledge of PLC, including PLC function description, programming language, program elements, data type, addressing mode, device definition, program comment function and programming, use of main program and subprogram, etc.

2. Readers familiar with PLC

For readers who are already familiar with the basic concepts and programming tools of PLC, you can directly read Chapter 5 Basic Instructions and Chapter 6 Application Instructions in this book. These two chapters provide a complete description of the VEDA VC series PLC instructions. If you want to know how to use the sequential function chart, high-speed IO, interrupt and communication functions, please refer to Chapters 7 to 9. If you want to know the functions of positioning control, please refer to Chapter 11 Guide for Using the Positioning Function. At the same time, for the convenience of readers, Appendix 8 Instruction Sorting Index Table and Appendix 9 Instruction Classification Index Table of this book also provide readers with instructions for finding relevant instructions according to the instruction classification and the alphabetical order of the English name of the instruction.

3. Relevant programming manuals can be downloaded from the official website:www.drives.ru
4. VEDA MC provides customers with all-round technical support. Users can contact the nearest VEDA MC office or customer service center, or directly contact the company headquarters.
5. The intellectual property of this manual belongs to the copyright of VEDA MC. The company is committed to product optimization and improvement, and constantly updates and improves this manual according to product optimization. This version of the manual is subject to update without notice. Users are welcome to visit our website at any time to download the latest version of the manual and materials.

Here we warmly welcome users and readers to consult and exchange usage methods in various forms, and feedback errors and omissions in the manual.

Service Hotline: +7 (495) 792-57-57

company website:www.drives.ru

Address: 143581 Moscow region, Istra city, Leshkovo 217

Chapter 1 Product Overview

1.1 VEDA VC series PLC Product Introduction

VEDA VC series products are PLCs with integrated structure, with built-in high-performance microprocessor and core operation control system, integrating input points and output points, expansion module bus, etc. The product series also includes I/O expansion modules, special modules; The main module integrates 2~3 communication ports; the I/O configured by the main module also includes high-speed counting and high-speed pulse output channels, which can be used for precise positioning; it has rich built-in programming resources, and adopts three standardized programming languages. The powerful VEDA PCT programming software can realize debugging and monitoring means; it has a perfect user program safety protection mechanism.

1.1.1 VEDA VC series product performance specifications

Specification	VC-B Series	VC-S Series	VC-SM Series	VC-P Series
Program Capacity	16K	64K		200K
Basic command speed	0.2µS	0.065µs		
High-speed input	2-way 50kHz; 6-way 10kHz	8 channels*200kHz		
High speed output	3 way*100kHz	8 way*200kHz		
Digital filtering	X0 to X7 with digital filtering can be set			
Power down storage	2K Bytes	48K Bytes		
COM Serial Communication	Two-way communication port COM0：RS232 COM1：RS485	Two-way communication port COM0：RS232 COM1：RS485		
USB	Support USB-Type-C interface			
CANopen	Not supported	Self-contained CAN communication port (Master supports up to 64 configurations, slave supports up to 8 PDO's)		
Ethernet	Not supported	Comes with Ethernet communication port Program on download, Modbus-TCP, Supports up to 16 slave connections		
Left communication extension type	Support 1-channel RS-485 extension			
Right communication extension type	Supports up to 15 modules, including up to 8 special modules			
Positioning commands	Support multiple positioning functions	Newly added multiple positioning function types		
Real Time Clock	Support			
S-curve acceleration and deceleration	Not supported	Support		
Interpolation Instructions	Not supported	Not supported	Support two-axis linear and Two-axis circular interpolation	
Electronic cams	Not supported	Not supported	Supports 4-axis electronic cams	
Flying shear / chase shear non-standard	Not supported	Not supported	Support 4-axis flying shear/tracking shear	

1.1.2 VC-B series main module interface description

The external structure of the VC-B series main module is shown in the figure below (take VC-B-A-16-14R as an example).

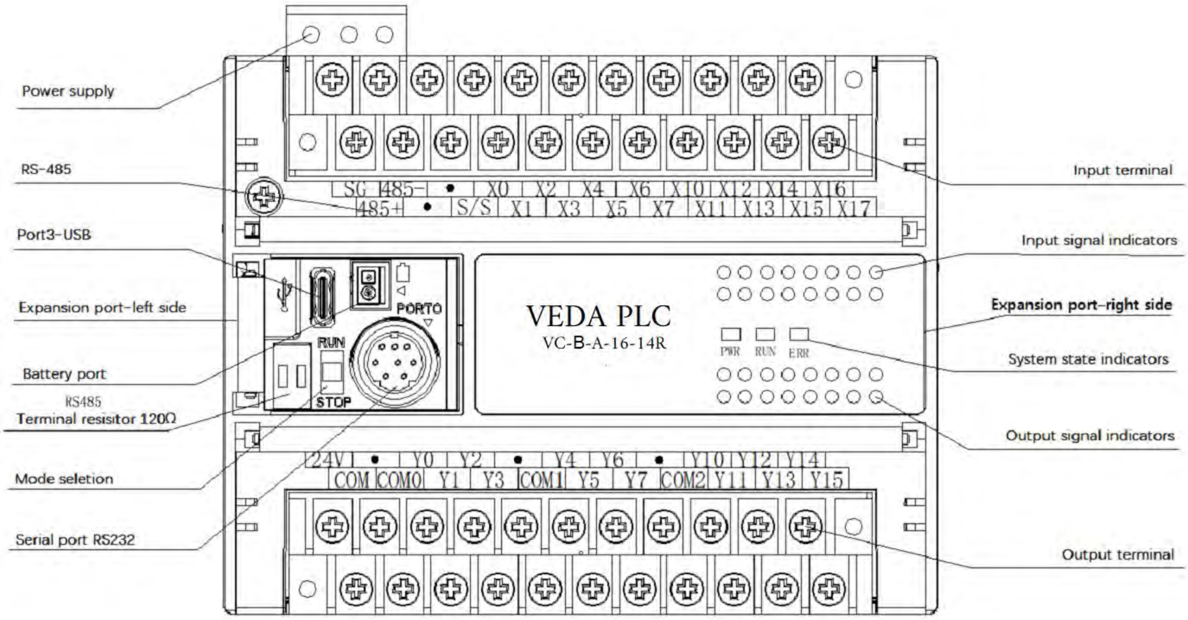


Figure 1-1 VC-B series main module outline structure

1.1.3 VC-S series main module interface description

The external structure of the VC-S series main module is shown in the figure below (Take VC-S-A-16-16T as an example).

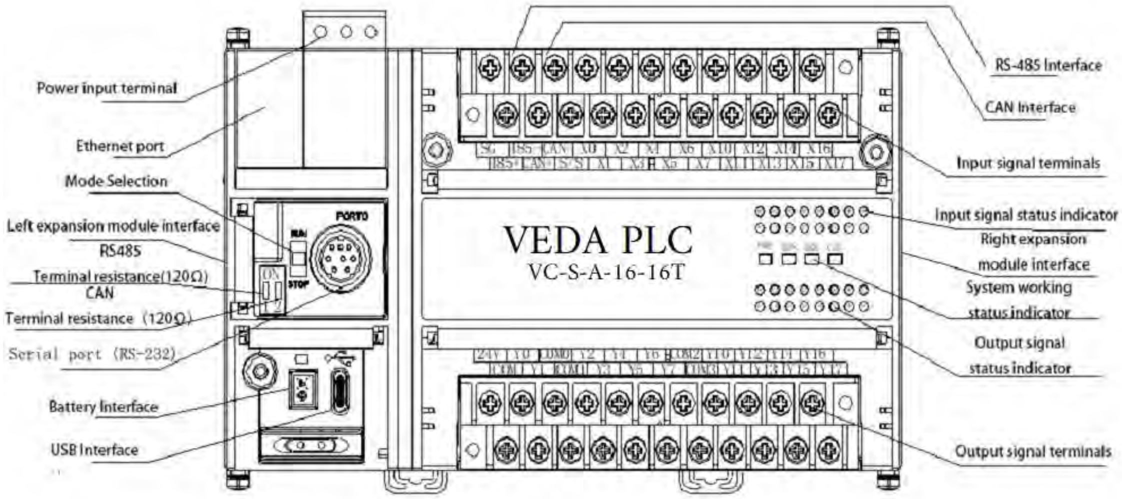


Figure 1-1 VC-S series main module outline structure

1.1.4 VC-P series main module interface description

The external structure of the VC-P series main module is shown in the figure below (Take VC-P-A-16-16T-16 as an example).

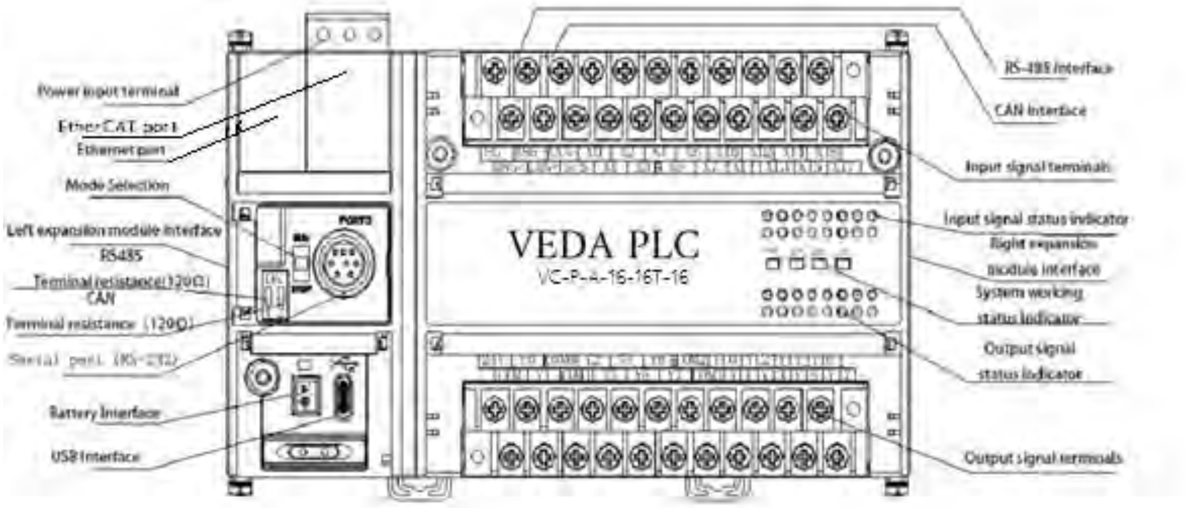


Figure 1-1 VC-P series main module outline structure

1.2 VEDA PCT Programming Software

VEDA PCT is a special programming software for VEDA VC series PLC products. The software can be downloaded from the VEDA MC website.

VEDA PCT programming software is a Standard Windows software, a graphical PLC programming tool, which is operated by mouse and keyboard. 3 standard languages can be selected for programming: Ladder Diagram, Instruction List, SFC Sequential Function Chart.

The connection between VEDA PCT programming software and PLC adopts serial programming cable, Modbus network programming can also be realized through serial port conversion, and remote programming can also be realized through Modbus. For the content of Modbus programming and remote monitoring, please refer to the relevant content of 《VEDA PCT Programming Software User Manual》.

1.2.1 Basic configuration

VEDA PCT programming software runs on IBM PC microcomputer or compatible machine, and needs to be installed on the Microsoft Windows series operating system. Compatible operating systems include Windows 98, Windows Me, NT 4.0, Windows 2000 and Windows XP.

The minimum and recommended configurations required by VEDA PCT are as follows:

Project	Minimum configuration	Recommended configuration
Cpu	Equivalent to intel's pentium 233 or above cpu	Equivalent to intel's Pentium 1G or above cpu
Memory	64M	128M
Graphics card	Can work in 640×480 resolution, 256 color mode	Can work in 800×600 resolution, 65535 color mode
Communication port	There must be a rs232 serial port output by a db9 socket (or use a usb interface through a usb-rs232 converter, a separate converter must be provided)	
Other equipment	VEDA PLC special programming cable	

1.2.2 VEDA PCT programming software installation process

The VEDA PCT installation package released by VEDA MC is a separate executable program. Double-click to start the installation process, and install it step by step according to the installation wizard. Users can choose different installation paths according to their own needs.

After the installation is complete, the VEDA program group will appear in the start menu; at the same time, the installer will also install the VEDA PCT shortcut icon on the desktop, and double-click the shortcut icon to run the program.

Uninstallation operation: Software uninstallation can be performed through the Windows Control Panel. To upgrade and install a new version of VEDA PCT software, please uninstall the old version of VEDA PCT software first.

1.2.3 VEDA PCT running interface

The main interface of this program basically includes seven parts: menu, tool bar, connect window, command tree window, message window, status bar and work area.

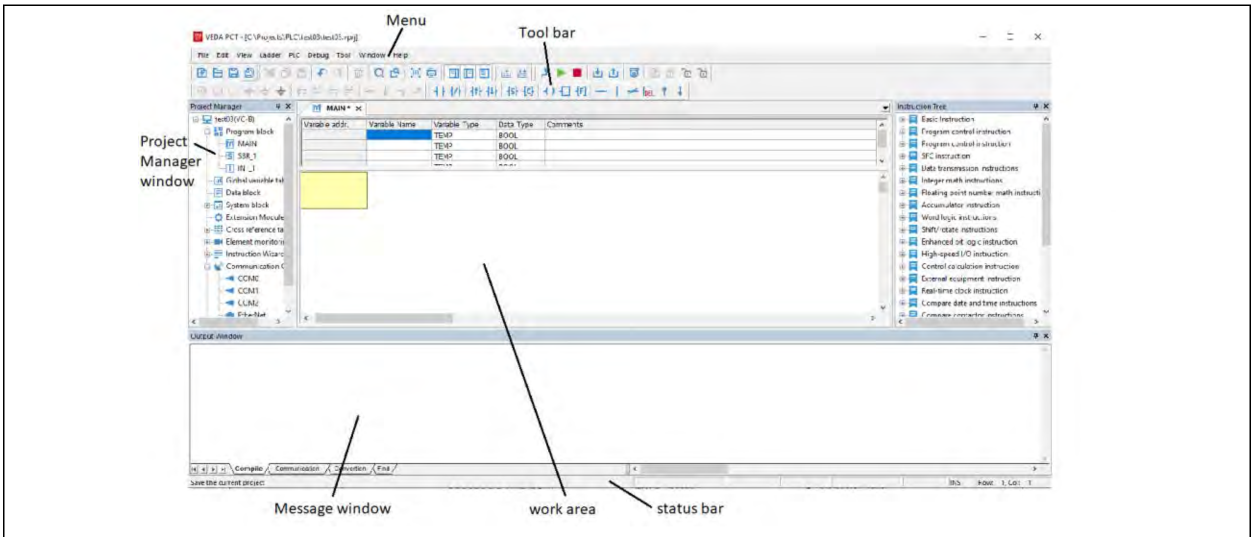


Figure 1-1 VEDA PCT main interface

Please refer to 《VEDA PCT Programming Software User's Manual》, which introduces the use of VEDA PCT programming software in detail.

1.2.4 Programming cable

Customers can program and debug the PLC through the serial programming cable or USB cable provided by VEDA MC.

Chapter 2 Function Description

2.1 Programming Resources and Principles

2.1.1 VC-B series programming resources

Name		Indicators and descriptions	
I/O configuration	Maximum number of I/O points	128 points (theoretical value)	
	Number of expansion modules	The total number of I/O expansion modules + special modules does not exceed 15	
User file capacity	User program capacity	16k steps	
	Data block size	8000 D elements	
Command speed	Basic instructions	0.2μs/instruction	
	Application instruction	Severalμs～hundredsμs/command	
Number of instructions	Basic instructions	32	
	Application instruction	234	
Device resource	Input and output points	128 in/128 out (Enter X0~X177, output Y0~Y177);The X, Y element address numbers are in octal addressing,	
	Auxiliary relay	2048 points (M0～M2047)	
	Local auxiliary relay	64 points (LM0 to LM63)	
	Special auxiliary relay	512 points (SM0 to SM511)	
	Sstatus relay	1024 points (S0～S1023)	
	Timer	256 (T0～T255) (1) 100ms accuracy T0～T209 (2) 10ms accuracy T210～T251 (3) 1ms accuracy T252～T255	
	Counter	263 (C0～C263) (1) 16-bit up-counter C0～C199 (2) 32-bit up-down counter C200～C235 (3) 32-bit high-speed counter C236～C263	
	Data register	8000 (D0～D7999)	
	Local data register	64 (v0～v63)	
	Indexed addressing register	16 (z0～z15)	
	Special data register	512 (SD0～SD511)	
Interrupt resource	External input interrupt	16 (interrupt trigger edge can be set by the user, corresponding to the rising and falling edges of X0~X7 terminals)	
	High-speed counter interrupt	8	
	Internal timed interrupt	3	
	Serial port interrupt	6	
	PTO output complete interrupt	3	
Communication function	Communication port	2 asynchronous serial communication ports: communication port 0: RS232; communication port 1: RS485; 1 USB interface;	
	Protocol	Modbus communication protocol, free port protocol, N: N (VEDA MC special protocol),	
Special function	High-speed counter	X0, X1	Single input: 50khz. When X0~X7 are input at the same time, the sum of the frequency is not more than 80khz
		X2～X7	Single input: 10khz
	High-speed pulse output	Y0, Y1, Y2	100khz three independent outputs (transistor output type only)
	Digital filter function	X0~X7 use digital filtering, other ports use hardware filtering	
	Subroutine call	A maximum of 64 user subprograms are allowed, and 6 levels of subprogram nesting are allowed. Support local variables, each subroutine can provide up to 16 parameters to pass, support variable aliases	

Name		Indicators and descriptions	
Special function	User program protection measures	Upload password	Provide 3 forms of password, the password does not exceed 8 characters, each character is an alphanumeric combination, case sensitive
		Download password	
		Cclock password	
		Subroutine encryption	Password no more than 16 characters, each character is alphanumeric, case sensitive
		Other protective measures	Provide the function of prohibiting formatting and uploading
	Programmatically	VEDA PCT programming	It needs to be installed and run in IBM PC microcomputer or compatible machine
	Real time clock	Built-in, back-up battery powered	

2.1.2 VC-S series programming resources

Name		Indicators and descriptions
I/O configuration	Maximum number of I/O points	512 points (256 in/256 out)
	Number of expansion modules	The total number of I/O expansion modules + special modules does not exceed 15
User file capacity	User program capacity	64k steps
	Data block size	8000 D elements, 32K R elements
Command speed	Basic instructions	0.065μs/instruction
	Application instruction	Several μs～hundreds μs/command
Number of instructions	Basic instructions	32
	Application instruction	286
Device resource	Input and output points	512 in/512 out (Enter X0~X777, output Y0~Y777)
	Auxiliary relay	10240 points (M0～M10239)
	Local auxiliary relay	64 points (LM0 to LM63)
	Special auxiliary relay	1024 points (SM0 to SM1023)
	Status relay	4096 points (S0～S4095)
	Timer	512 (T0～T511) (1) 100ms accuracy T0～T209 (2) 10ms accuracy T210～T479 (3) 1ms accuracy T480～T511
	Counter	263 (C0～C263) (1) 16-bit up-counter C0～C199 (2) 32-bit up-down counter C200～C235 (3) 32-bit high-speed counter C236～C263
	Data register	8000 (D0~D7999) 32768 (R0~R32767)
	Local data register	64 (V0～V63)
	Indexed Addressing Register	16 (Z0～Z15)
	Special data register	1024 (SD0～SD1023)
Interrupt resource	External input interrupt	16 (interrupt trigger edge can be set by the user, corresponding to the rising and falling edges of X0~X7 terminals)
	High-speed counter interrupt	8
	Internal timed interrupt	3
	Serial port interrupt	6
	PTO output complete interrupt	8

Name		Indicators and descriptions	
Communication function	Cmmunication port	2 asynchronous serial communication ports: communication port 0: RS232; communication port 1: RS485; 1 USB interface; 1 CAN communication port; 1 Ethernet communication port;	
	Protocol	Modbus communication protocol, free port protocol, N: N (VEDA MC special protocol), can form 1:N, N: N communication network	
Special function	High-speed counter	X0~X7	200kHz*8 high-speed input
	High-speed pulse output	Y0~Y7	200kHz*8 independent outputs (only for transistor output type)
	Digital filter function	X0~X7 use digital filtering, other ports use hardware filtering	
	Subroutine call	A maximum of 64 user subprograms are allowed, and 6 levels of subprogram nesting are allowed. Support local variables, each subroutine can provide up to 16 parameters to pass, support variable aliases	
Special function	User Program Protection Measures	Upload password	Provide 3 forms of password, the password does not exceed 8 characters, each character is an alphanumeric combination, case sensitive
		Download password	
		Clock password	
		Subroutine encryption	Password no more than 16 characters, each character is alphanumeric, case sensitive
		Other protective measures	Provide the function of prohibiting formatting and uploading
	Programmatically	VEDA PCT programming software	It needs to be installed and run in IBM PC microcomputer or compatible machine
	Real Time Clock	Built-in, back-up battery powered	

2.1.3 VC-P series programming resources

Name		Indicators and descriptions
I/O configuration	Maximum number of I/O points	512 points (256 in/256 out)
	Number of expansion modules	The total number of I/O expansion modules + special modules does not exceed 15
User file capacity	User program capacity	200k steps
	Data block size	8000 D elements, 32K R elements
Command speed	Basic instructions	0.065μs/instruction
	Application instruction	Several μs～hundreds μs/command
Number of instructions	Basic instructions	32
	Application instruction	286
Device resource	Input and output points	512 in/512 out (Enter X0~X777, output Y0~Y777)
	Auxiliary relay	10240 points (M0～M10239)
	Local auxiliary relay	64 points (LM0 to LM63)
	Special auxiliary relay	1024 points (SM0 to SM1023)
	Status relay	4096 points (S0～S4095)
	Timer	512 (T0～T511) (1) 100ms accuracy T0～T209 (2) 10ms accuracy T210～T479 (3) 1ms accuracy T480～T511
	Counter	263 (C0～C263) (1) 16-bit up-counter C0～C199 (2) 32-bit up-down counter C200～C235 (3) 32-bit high-speed counter C236～C263
	Data register	8000 (D0~D7999) 32768 (R0~R32767)

Name		Indicators and descriptions	
	Local data register	64 (V0～V63)	
	Indexed Addressing Register	16 (Z0～Z15)	
	Special data register	1024 (SD0～SD1023)	
Interrupt resource	External input interrupt	16 (interrupt trigger edge can be set by the user, corresponding to the rising and falling edges of X0~X7 terminals)	
	High-speed counter interrupt	8	
	Internal timed interrupt	3	
	Serial port interrupt	6	
	PTO output complete interrupt	8	
Communication function	Cmmunication port	2 asynchronous serial communication ports: communication port 0: RS232; communication port 1: RS485; 1 USB interface; 1 CAN communication port; 1 Ethernet communication port; 1 EtherCAT communication port	
	Protocol	Modbus communication protocol, free port protocol, N: N (VEDA MC special protocol), can form 1:N, N: N communication network	
Special function	High-speed counter	X0~X7	200kHz*8 high-speed input
	High-speed pulse output	Y0~Y7	200kHz*8 independent outputs (only for transistor output type)
	Digital filter function	X0~X7 use digital filtering, other ports use hardware filtering	
	Subroutine call	A maximum of 64 user subprograms are allowed, and 6 levels of subprogram nesting are allowed. Support local variables, each subroutine can provide up to 16 parameters to pass, support variable aliases	
Special function	User Program Protection Measures	Upload password	Provide 3 forms of password, the password does not exceed 8 characters, each character is an alphanumeric combination, case sensitive
		Download password	
		Clock password	
		Subroutine encryption	Password no more than 16 characters, each character is alphanumeric, case sensitive
		Other protective measures	Provide the function of prohibiting formatting and uploading
	Programmatically	VEDA PCT programming software	It needs to be installed and run in IBM PC microcomputer or compatible machine
	Real Time Clock	Built-in, back-up battery powered	

2.2 PLC Operating Principle

2.2.1 PLC operating mechanism (scan cycle model)

The main module of VEDA VC series PLC operates according to the scan cycle model.

The system executes four tasks sequentially and cyclically: executing user programs, communication, internal affairs, and refreshing I/O. Each round of tasks is called a scan cycle.

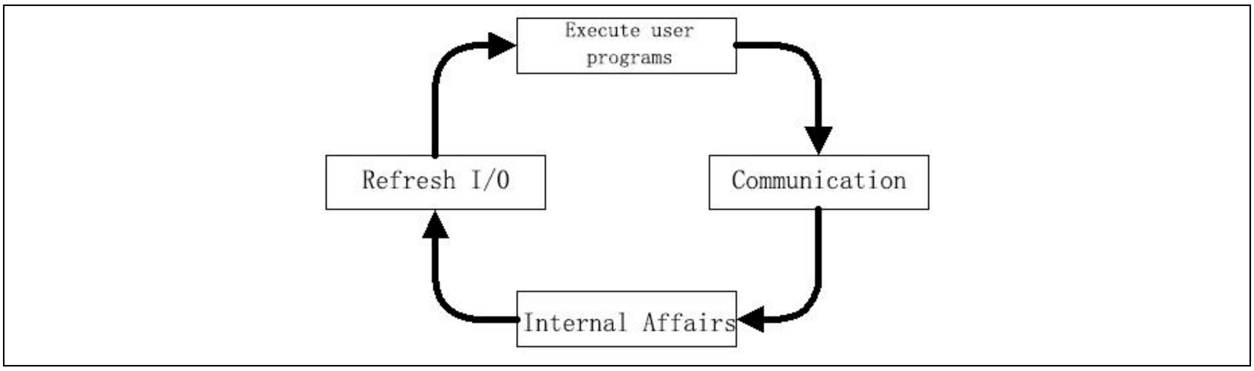


Figure 2-1 PLC operating mechanism

1) **Execute user program tasks**

The system executes the instruction sequence of the user program sequentially, starting from the first main program instruction, and executes the instruction sequence in the user program one by one until the end instruction of the main program is executed.

2) **Communication tasks**

It communicates with the programming software and responds to programming communication commands such as download, run, and stop issued by the programming software.

3) **Internal Affairs tasks**

Handles various system housekeeping, such as refreshing panel indicator lights, updating software timer timing values, refreshing special auxiliary relays and special data registers.

4) **Refresh I/O tasks**

Refresh I/O includes an output refresh stage and an input refresh stage.

Output refresh stage: According to the value of the Y element (ON or OFF), turn on or off the corresponding hardware output point.

Input refresh stage: Convert the on/off state of the hardware input point to the corresponding X element value (ON or OFF).

2.2.2 User program runs watchdog function

The system will monitor the running time of the user program in each scan cycle. Once it is found that the running time of the user program exceeds the set value, it will stop the running of the user program. The user can set the watchdog time in the system block dialog **Time setting** page of the VEDA PCT background software interface.

2.2.3 Constant scan operation mode

The constant scan operation mode means that the system is in the running state, and the time of each scan cycle is the same. The user can activate the constant scan mode and set the constant scan time in the **Time setting** page of the System Block dialog box of the VEDA PCT background software interface. The default value of the constant scan period is 0, that is, the constant scan is prohibited; when the actual scan period is greater than the constant scan period, the program runs according to the actual scan period.

📖 Notice

The constant scan time setting cannot be greater than the watchdog timer time

2.2.4 User file download and storage

The main module can be programmed and controlled by downloading specific user files to the main module.

User files include four types: user program files, data block files, system block files, and user auxiliary information files. User auxiliary information files include: global variable table, user data source files.

Users can choose to download user program files, data block files, and system block files. When the download operation is selected, the corresponding user assistance information file will also be bundled and downloaded.

All user files of VEDA VC series are solidified into the FLASH area of the main module for permanent storage.

📖 Notice

In order to ensure that the downloaded file can be properly solidified into the main module, the main module should be powered normally within a period of time (more than 30 seconds) after the file is downloaded.

2.2.5 Component initialization

When the PLC enters the running state (STOP→RUN), it will initialize the related soft components according to the data, data blocks and component values saved after power failure. The priority order of various data is as shown in the table below.

Table 2-1The priority order of various data initialization when the PLC enters the running state

Memory type	Power OFF→ON	STOP→RUN
Save data when power off	Highest	Highest
Data block (when "Data block valid" is selected in the advanced settings of the system block)	Middle	Middle
Component value (when "component value hold" is selected in the advanced settings of the system block)	—	Low

2.2.6 Power-off save data function

1) Conditions for saving data when power off

When the system confirms that a power failure occurs, it will stop the running of the user program, and save the component data values within the specified storage range in the system block to the power failure backup file.

2) Component power-on recovery

After power-on, if the power-off backup file is correct, the value of the specified device will be restored to the value saved at the last power-off.

After power-on, the system clears the components in the non-saved range.

If the backup file is lost or wrong, the system will clear all components.

3) Save range settings

The range of the holding element can be set in the system block holding range, seeFigure 2-1and examples.

VC-B series save range setting only supports 1 group, VC-S series save range setting supports 2 groups.

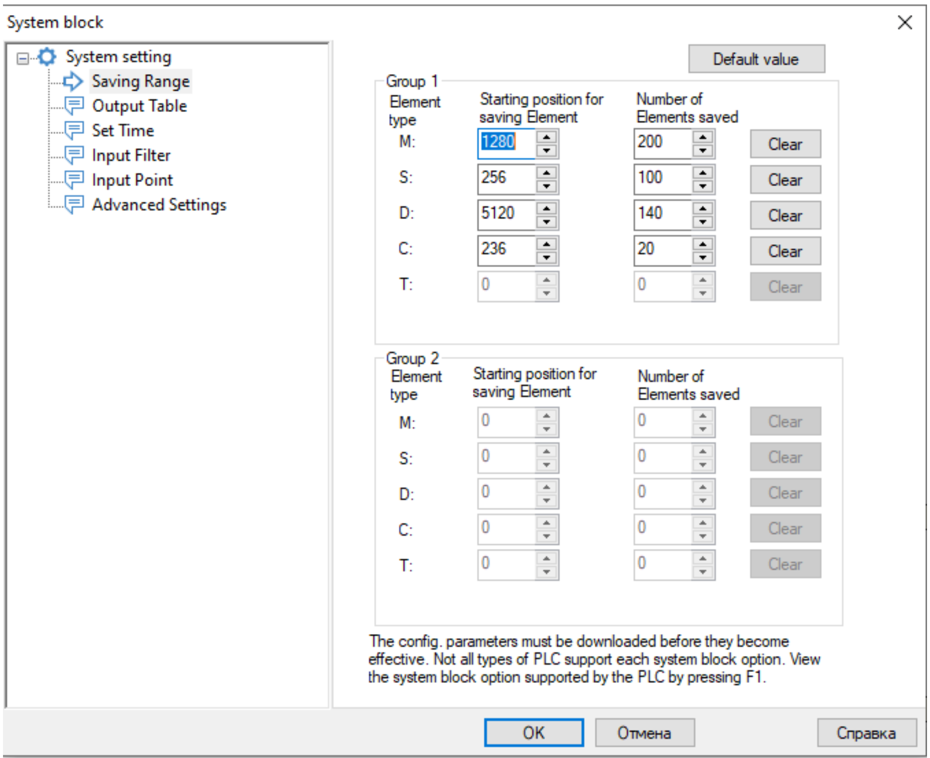


Figure 2-1 Set save range

📖 Notice

After the VC-B series PLC is powered off, the data of its holding components are stored in the permanent storage medium.

2.2.7 Digital filter function for input points

The input points (X0~X7) of the main module are equipped with digital filtering function, which can filter out the interference signal of the port. You can configure the **Input filter** item in the system block to change the setting of the input filter constant.

2.2.8 No battery mode

VEDA VC series main modules can work without batteries. When the user selects the batteryless mode, the system will not report system errors caused by the lack of batteries (loss of hold elements, loss of mandatory tables, errors in user program files).

See the description of the No battery mode configuration item in the advanced settings of the system block.

2.2.9 User program protection measures

The PLC is designed with security strategies such as multi-level password protection.

Table 2-1User Program Protection Measures

User Program Protection Measures	Illustrate
Formatting is prohibited	After setting the format prohibition in the system block and downloading the system block into the PLC, the user program, system block and data block inside the PLC cannot be deleted by formatting. To unblock formatting, you must re-download a new system block, and the system block should not be set to forbid formatting
Download password	Used to restrict download function
Disable upload	When the download operation is performed and the option to prohibit upload is selected in the download dialog box, the user will not be able to upload in the future even if there is an upload password. To unblock uploads, the user data must be downloaded again and the Allow uploads option selected in the download dialog
Upload password	Used to limit upload functionality
Clock password	Used to limit the clock setting function
Program password	For the main program, subprogram and interrupt subprogram, the programmer can set a password for encryption. When the project is opened in the programming software, the encrypted content of the above program cannot be viewed and edited. Only after the decryption dialog box is opened and the correct password is entered, the program can be decrypted for viewing and editing. Encryption method: Right-click the program to be encrypted, select Encrypt/Decrypt in the menu, and then enter the password and confirm the password to realize encryption. Decryption method: Right-click the program to be decrypted, select Encrypt/Decrypt in the menu, and enter the correct password to decrypt

📖 Notice

If the password is continuously input and retried for 5 times, the VEDA VC series small PLC will prohibit the password input function for 5 minutes.

2.3 System Configuration

2.3.1 System block

The PLC configuration information configured by the system block is an important part of the PLC user file, which is called the system block file. The PLC needs to compile and download the system block file before using it.

System block configuration includes the following:

- Save range (component save range)
- Output table (Output Table Settings)
- Setup time (watchdog, constant scan time)
- Input filter (set X0~X7 filter time)
- Input point (input point power-on mode)
- Advanced settings (Data Blocks, Memory Element Retention, No Battery Mode, Disable Formatting)

After configuring the system block, select the **PLC/Compile All** menu, the system block file of the project is compiled, and then the download operation can be performed.

A. Saving Range

When the PLC is powered off, it can save some data in the components of the set storage range to the power-off storage area, and can continue to retain and use these data after the power is turned on again.

in the dialog on the first page, you can see the **Saving Range**, configure the save element address range, such as Figure 2-1 shown.

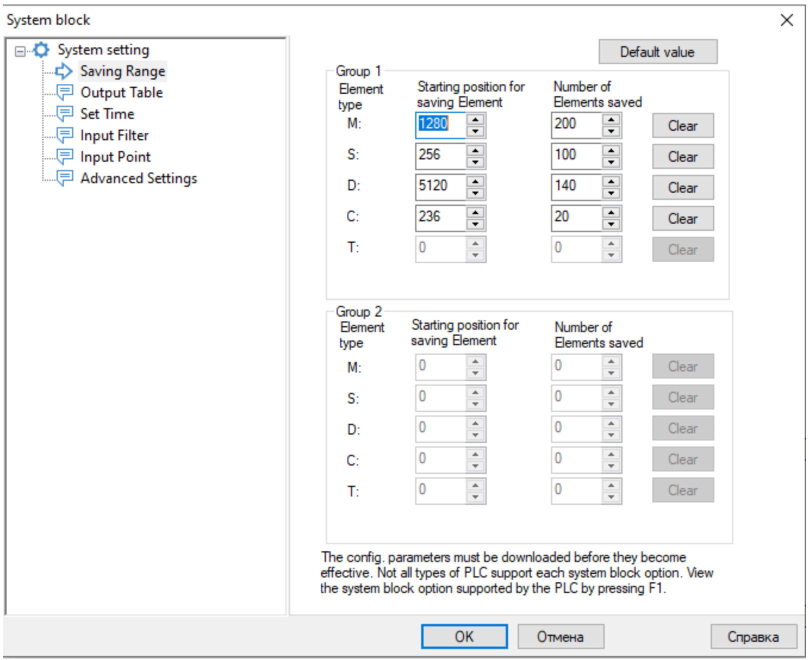


Figure 2-1 Configuration save element address range

📖 Notice

The range of device addresses and the number of groups that can be saved differ depending on the PLC model.

The **D, M, S, T,** and **C** elements are automatically set to save a certain range by default without modification to the system block.

You can modify the address range of the components to be saved on this page. Click the **Clear** button to the right of each row of components to set the saved number of corresponding components to zero.

The VC-B series save range can define a set of reserved ranges at most.

📖 Notice

VC-B series PLC cannot save the data of T element.

System operation when power off: PLC will save the components in the power-off backup file according to the range defined in the above figure.

System operation at power-on: PLC checks whether the data saved in the power-off save area is correct. If the data in the power-off save area is successfully saved, the reserved area of the SRAM memory remains unchanged. If the content in the saved power-down saving area is wrong, the PLC will clear the elements in the SRAM (including the reserved and non-reserved areas).

B. Output Table

Click the output table label, you can set the output point status when the PLC stops, such as Figure 2-2 shown.

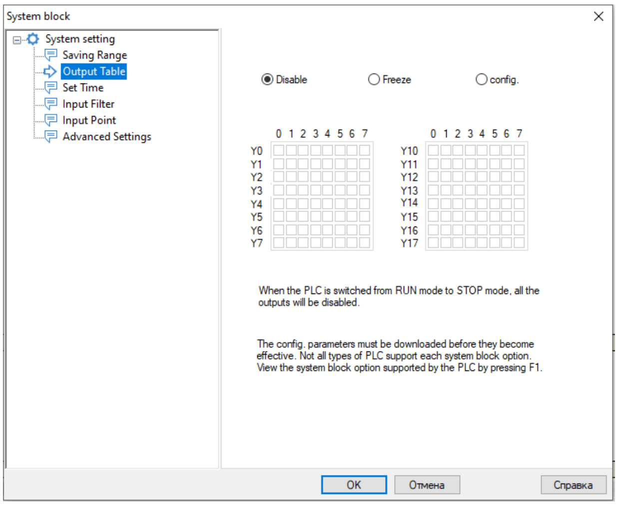


Figure 2-2 set output table

The function of the output table setting is to set the output point configuration in the stop state. When the CPU is in the stop state, the output point configuration has the following three options:

(1) Forbidden: PLC will prohibit all output points when it stops, and it will take effect when PLC switches from running state to stop state.

(2) Freeze: The PLC will freeze all output points in the last state in the stop state.

(3) Configuration: The PLC will set the output point to a known state when it stops. The default state of all output points is the off (0) state.

C. Set Time

Figure 2-5 is the setting time page.

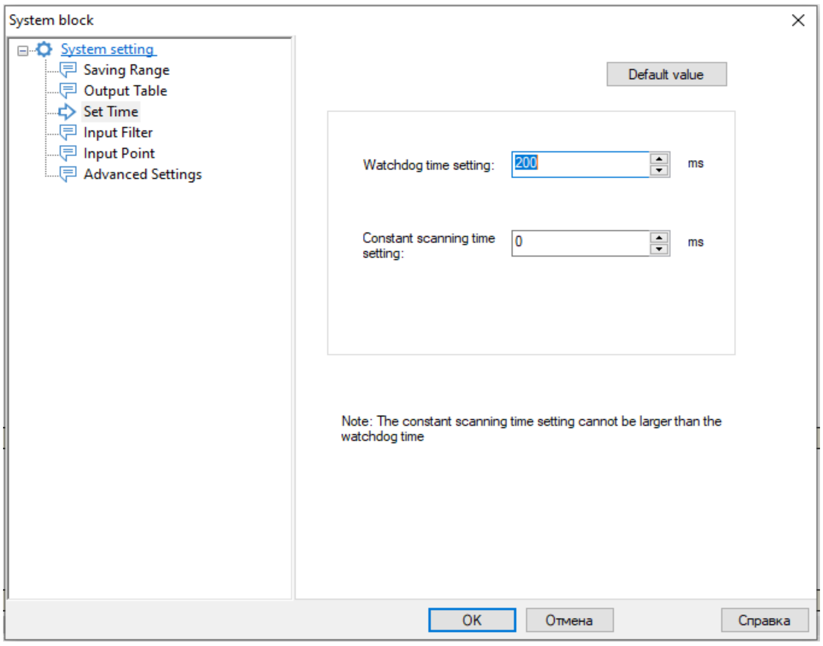


Figure 2-3 set time

D. Watchdog timer setting

Set the user program running watchdog time. The watchdog time refers to the maximum time that the user program is allowed to run. When the actual execution time of the user program exceeds the watchdog time, the PLC will stop the user program, light the program alarm light (red), and output according to the system configuration. The watchdog time can be set from 100ms to 1000ms, and the default value is 200ms.

E. Constant scan time setting

Constant scan time refers to the time that the system scans the registers in constant time. Read the constant scan time setting register of the system, and scan the user program only once within the constant time. The settable range of the constant time is: 0ms～1000ms. The default is 0ms, which does not enable constant scan time. Non-zero enables the set constant scan time.

F. Input Filter

Click the **Input Filter** label, you can set an input filter constant for the PLC input point, and filter out the interference signal introduced from the outside of the input point through the digital filter function. The switch input points with digital filtering function are X0~X7. Other switch input points use hardware filtering technology. VEDA VC series input filter can be set separately for each input port, VC-B filter constant unit: (ms) can be set continuously from 0us to 60ms, VC-S filter constant unit: (0.25us) can be set continuously from 0us to 60ms. The input filter settings of VC-B and VC-S are as follows Figure 2-4 shown.

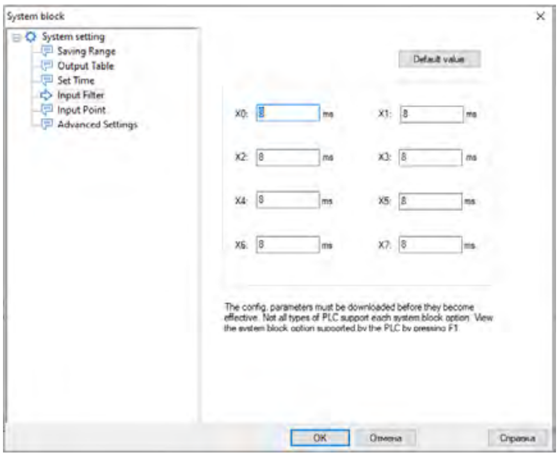
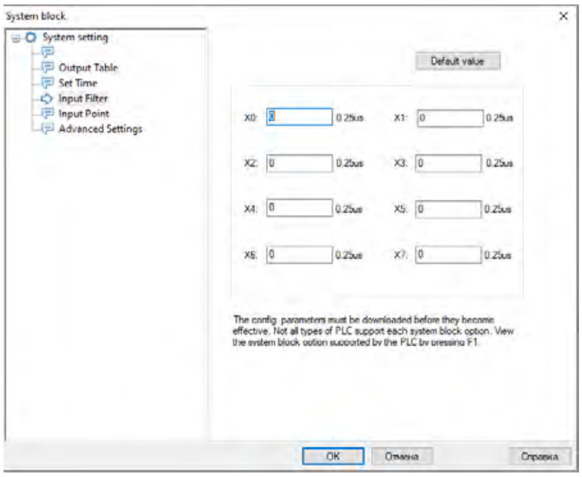
VC-B series	VC-S series
	

Figure 2-4 set input filter

G. Input Point

1. Specifies the input point for power on

When the **Disable Input Point** checkbox is not checked, an input point in X0~X17 can be designated to force the PLC to enter the RUN state. When the system is in the STOP state, when the point is detected as ON, the system state is switched from stop to running. state.

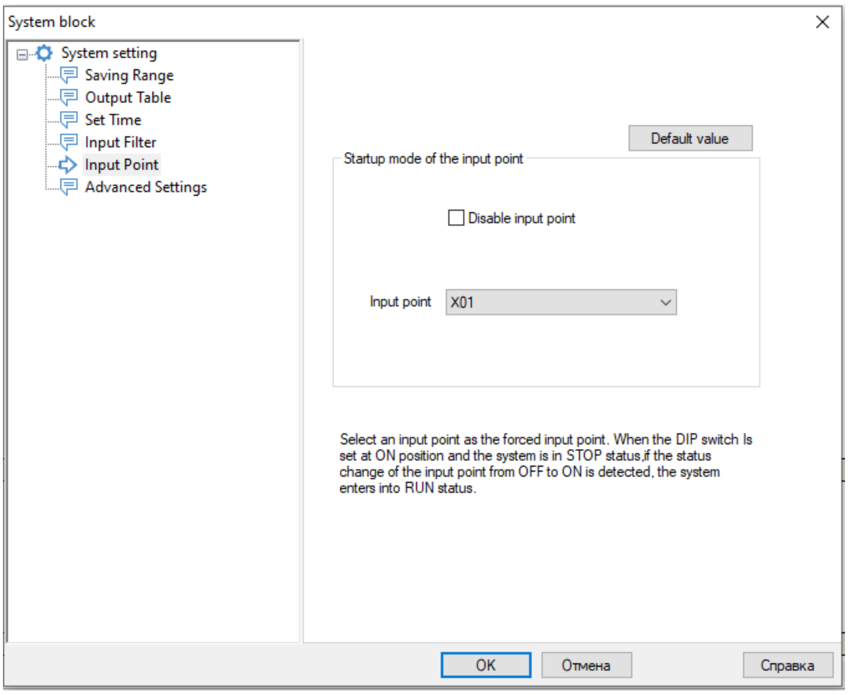


Figure 2-5 set input point

H. Advanced settings

Function: Configure some advanced settings such as data block valid, component value retention, no battery mode, etc.

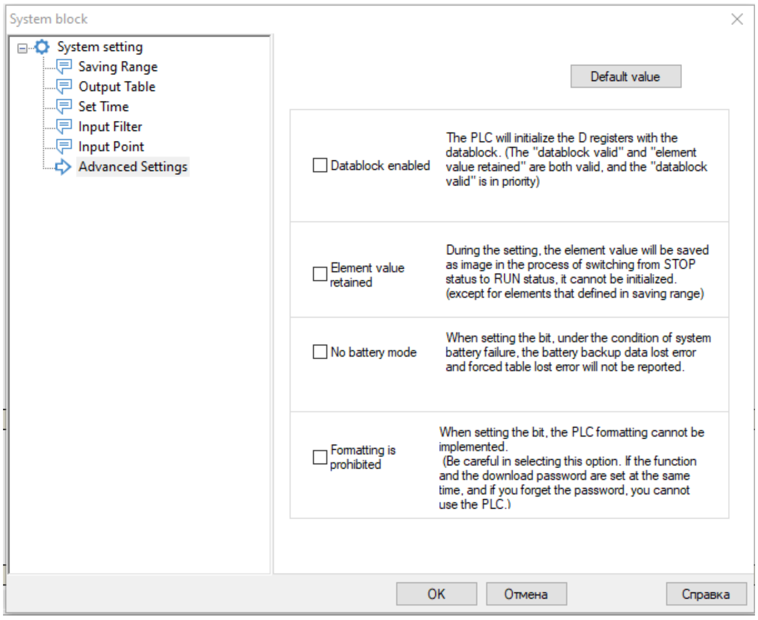


Figure 2-6 advanced settings

I. Data block is valid

If this item is selected, the PLC will use the data block to initialize the D element from the STOP state to the RUN state.

J. Component value hold

If this option is selected, the component values are stored as mirror images from the STOP state to the RUN state, and no initialization is performed.

📖 Notice

When the **Data block valid** and the **Element value** remains valid at the same time, the **Data block valid** takes precedence. See2.2.5 Component initialization.

K. No battery mode

If this option is selected, the system will not report the battery backup data loss error and mandatory table loss error when the backup battery fails.

2.3.2 Data block

The data block is used to set the default value in the D element. After the setting is completed and compiled, it can be downloaded to the PLC. After the PLC enters the running state, the PLC will first use the data block to initialize the relevant D element.

In the data block editor, initial data assignments can be made to the D register (data memory), and word or double word assignments of D elements can be assigned, but bytes cannot be assigned. Comments can also be written in the data block editor. Adding a double slash before the string can set the following content as a comment.

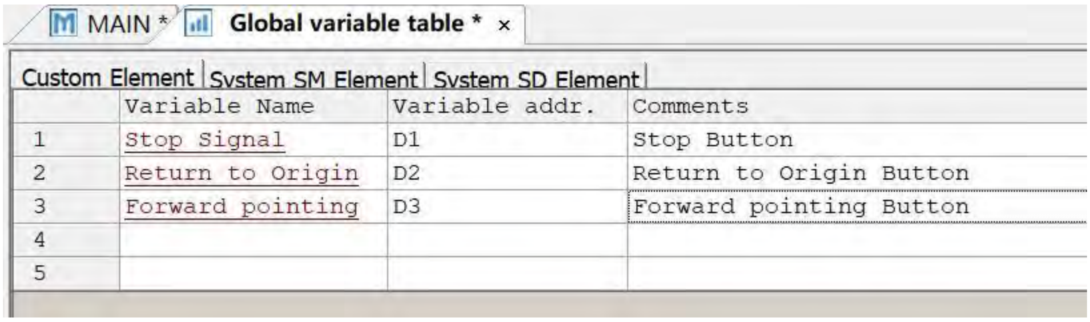
For detailed operation instructions of data blocks, please refer toChapter 4 4.2.3 Data Block

2.3.3 Global variable table

(1) A global variable is a meaningful symbolic name defined for an address of the PLC. The symbolic name can be accessed in the entire project scope, which is equivalent to using the component corresponding to the variable. The global variable is defined in the global variable table. The global variable table contains three attributes: **Variable name**, **Variable address**, and **Comment**.

(2) The definition rules of global variables are: A to Z, a to z, 0 to 9, underscores, and Chinese characters are mixed and combined. The variable name cannot start with a number, it also cannot be a separate number. The name is not case-sensitive, the length cannot exceed 8 bytes, and the component type letters and numbers cannot be used as program and variable names. The variable name cannot contain spaces, and cannot use the same name as the keyword. The reserved keywords include: basic data type names, instruction names, and operators in the instruction list language.

(3) For VC-B series small PLCs, the number of global variables allowed to be downloaded cannot exceed 140 (according to the maximum amount of comments). If the number exceeds140, it can only be saved locally, but cannot be downloaded to the PLC program. As shown in Figure 2-9:



	Variable Name	Variable addr.	Comments
1	Stop Signal	D1	Stop Button
2	Return to Origin	D2	Return to Origin Button
3	Forward pointing	D3	Forward pointing Button
4			
5			

Figure 2-1 global variable table

2.4 Operation Mode And State Control

There are three ways to control the PLC to enter or exit the running state:

1. through the mode selector switch;
2. By setting the input point power-on mode and external terminal in the system block, it is controlled by the designated terminal;
3. If the mode selection switch is in the ON position, the operation and stop of the PLC can also be controlled through the programming software.

2.4.1 System operation stop state concept

The working state of the main module is divided into running state and stop state.

1) Running state (run)

When the main module is running, the user program will be executed by the system, that is, one scan cycle completely includes four tasks (execute user program→communication→housekeeping→refresh i/o).

2) Stop state (stop)

When the main module is in the stopped state, the system does not execute the user program, but the other three tasks are still executed by the system in each scan cycle (communication → housekeeping → refresh i/o).

2.4.2 Run stop state transition

A. How to enter the running state (STOP→RUN)

1. Reset method

When the mode selection switch is in the on position, the system automatically enters the running state after reset (including system power-on reset).

📖 Notice

If the input point control mode system configuration item in the main module is valid, the state of the specified input terminal should be on, otherwise it cannot enter the running state.

2. Manual way

In the stop state, when the mode selection switch is toggled from the off position to the on position, the system enters the running state.

3. Input point boot mode

When the system block **input point power-on mode system** configuration item is valid, in the stop state, the system detects that the specified input point (x0~x17) has changed from off to on state, and the main module enters the running state.

📖 Notice

When the input point control mode is selected, the mode selection switch should be in the on position at the same time, otherwise it cannot enter the running state.

B. How to enter the stop state (run→stop)

1. Reset method

When the mode selection switch is in the off position, the system automatically enters the stop state after reset (including system power-on reset).

Notice

Even if the mode selection switch is in the on position, if the **input point control mode** system configuration item is valid and the state of the specified input point is off, the system can automatically enter the stop state after reset.

2. Manual way

In the running state, when the mode selection switch is toggled from the on position to the off position, the system enters the stop state.

3. Command control method

In the running state, when the stop instruction in the user program is effectively executed, the system enters the stop state.

4. Error stop method

When the system detects that there is a serious error (such as user program error, user program runs overtime, etc.), it automatically stops the execution of the user program.

2.4.3 Output point state setting in stop state

The user can set the output state of the output point (y) in the stop state, and three modes are provided for the user to choose:

1. Disable output mode—all output points are off in stop state.
2. Freeze output mode—stopped all output points remain in the state they were in before stopping.
3. Configure output mode—in stop state, user can set the state of output point in stop state as needed.
4. The user can set the state of the output point in the stop state in the system block **output table**. See2.3.1 System blockoutput table settings.

2.5 System Debugging

2.5.1 Program download and upload

1) Download

The download function is used to download the system blocks, data blocks, and user programs generated by VEDA PCT software to the PLC through the serial port, and the PLC is required to be in a stopped state during downloading. When downloading, if the program has changed since the last compilation, you will be prompted whether to recompile the program, as shown in Figure 2-10:

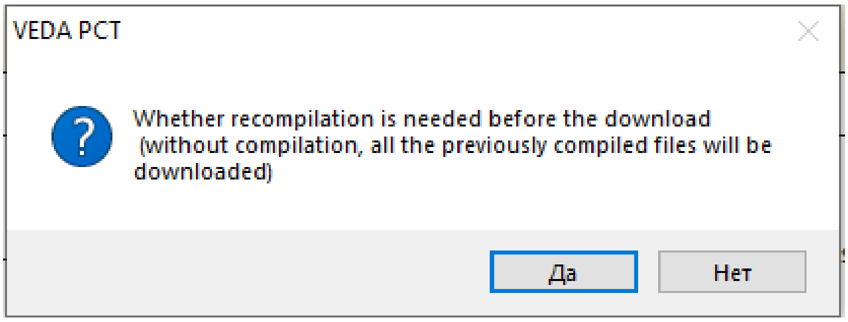


Figure 2-1 Recompile the program prompt

Notice

Select **No (N)** not to recompile, the software will use the result of the last edit, and the program downloaded to the PLC to run and the program displayed on the software interface will be different.

When downloading, if there is a download password and no download password is entered after starting the software, the software will pop up a password window to ask for the download password. After the password input is verified correctly, the download starts. If the password is incorrect, you will be prompted to re-enter the password. Click the Cancel button to exit the download.

2) Upload

The upload function is used to upload the system block, data block, user program and other contents in the PLC to the computer through the serial port, and save it for the new project. When the battery backup data is valid, if the battery backup data is valid, the corresponding user auxiliary information files will be bundled and uploaded when you select upload. Figure 2-11 shows the upload dialog box.

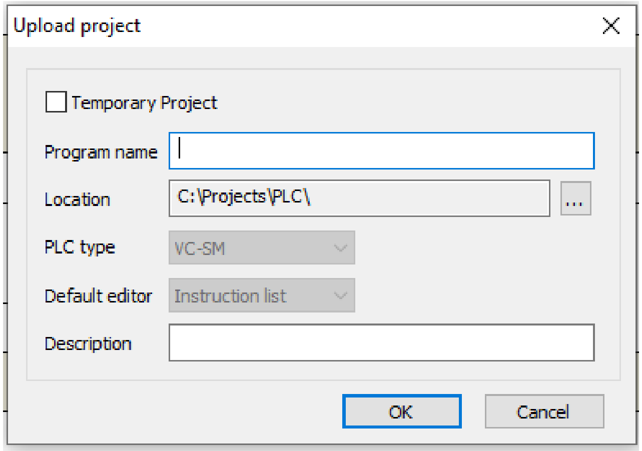


Figure 2-2 upload dialog

When uploading a program, if no upload password is set, the program can be uploaded directly. If there is an upload password, and the upload password is not entered after starting the software, the software will pop up a password window to ask for the upload password. If the password is entered correctly, the upload will start. If the password is incorrect, the software will prompt and return to the upload dialog interface.

If the upload prohibition function is selected when downloading the program, the PLC will not be able to upload the program in the future unless the correct password is entered to cancel the upload prohibition function.

2.5.2 Error reporting mechanism

The system can detect and report two types of errors: system errors and user program running errors.

1. System errors are errors caused by abnormal system operation.
2. User program running errors are errors caused by abnormal execution of user programs.

All errors are numbered uniformly, each error number represents an error, see details0system error code table.

1) System error reporting mechanism

When the system detects that there is a system error, the system error number will be written into the special data register sd3, and the special relay sm3 will be set at the same time, and you can read the error number stored in sd3 to know what system errors are currently occurring

When multiple system errors occur at the same time, the system indicates the error with the highest severity in sd3 according to the severity of the error.

Serious system errors will cause the user program to stop running, and will cause the err indicator on the main module to light up for a long time.

2) Error reporting mechanism for user program running errors

When a user program running error occurs, the system will set the special relay sm20, and at the same time write the number of the current error into the special data register sd20.

When the next application instruction is executed correctly, sm20 will be cleared, but the last wrong number is still recorded in sd20.

The system records successive user program running errors in the form of error record stack. Special data registers sd20~sd24 form an error record stack with a size of 5. Sd20~sd24 records the error codes of the last 5 user program operation errors.

When a user program running error occurs, and the current error code is inconsistent with the record in sd20, the error record push operation will occur. The following figure demonstrates the process of pushing the error code to the stack when the user program runs an error:

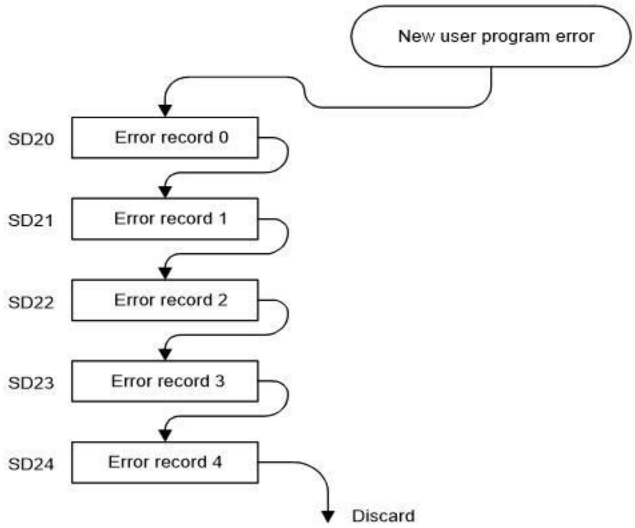


Figure 2-1 Error code push operation process

Serious user program running errors will cause the user program to stop running, and will cause the ERR indicator on the main module to light up for a long time, while generally serious user program running errors will not light the ERR indicator on the main module.

3) Check the error message online

In the case of connecting with the PLC through the serial port, the VEDA PCT programming software can read various status information of the current PLC, including the code and description information of the above-mentioned system errors and user program running errors.

In VEDA PCT software, click **PLC->PLC** Information option to open the following window.

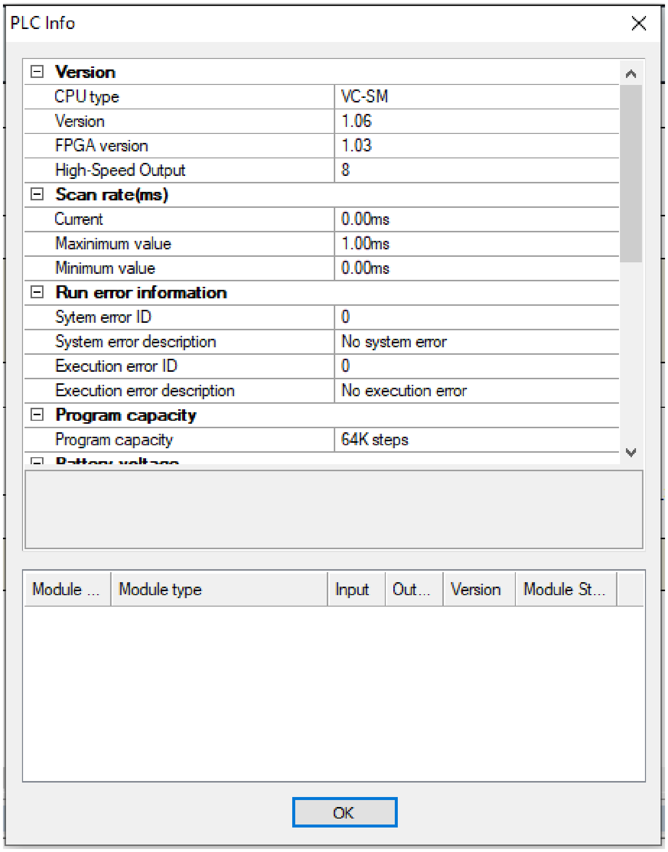


Figure 2-2 PLC information

The system error number in the figure is the system error number stored in SD3, and the **Execution error number** is the user program running error number stored in SD20. The related error descriptions displayed at the same time are available for users to reference.

2.5.3 Modify online

When you need to modify the program in the PLC running state, you can use the Modify online function.

🕮 warn

In the occasions that may cause personal injury or property damage, the online modification user program function should be used by professionals under the guarantee of corresponding safety measures.

1) How to operate

After ensuring that the software has successfully established a communication relationship with the PLC hardware, and the PLC is running, click the **Debug->Online Modify** menu to switch to the online modification state.

In the online modification state, the contents of the main program, subprogram and interrupt subprogram can be modified as in normal editing. After modification, click **PLC->Download** menu, the software will compile all programs of the current project and automatically download them to the PLC in hardware. After the download is complete, the PLC will run according to the newly downloaded program.

2) Limitation factor

1. In the online modification state, the global variable table and the local variable table of any program cannot be modified, nor can any subroutine/interrupt subroutine be added or deleted;

2. When the program is in the online modification state, if the PLC is stopped, the software will automatically exit the online modification state.

2.5.4 Clear and format

Clearing operations include: PLC component value clearing, PLC program clearing, and PLC data block clearing.

Formatting is to clear all data and programs in the PLC.

1) PLC component value clear

The PLC component value clearing function clears all component values in the PLC, and clearing the component values requires the PLC to be in a stopped state.

Clearing the component valuesin the PLC may cause the PLC to run incorrectly or lose the intermediate work data. Please use this function with caution. To prevent misoperation, the software will display a confirmation window during operation.

2) PLC program clear

The PLC program clearing function clears the user program in the PLC, and clearing the user program requires the PLC to be in a stopped state.

Clearing the user program in the PLC will cause the PLC to run without executing any user program. Please use this function with caution. To prevent misoperation, the software will display a confirmation window during operation.

3) PLC data block clear

The PLC data block clearing function clears all data block settings in the PLC, and clearing the data block requires the PLC to be in a stopped state.

Clearing the data block in the PLC will cause the PLC to no longer initialize the d element with the preset value of the data block after running. Please use this function with caution. To prevent misoperation, the software will display a confirmation window during operation.

4) PLC format

The PLC formatting function formats all the data in the PLC, including clearing the user program, restoring the default configuration, clearing and clearing the data block, and clearing the data block requires the PLC to be in a stopped state.

This operation will lose all the data that has been downloaded and set in the PLC, please use this function with caution. To prevent misoperation, the software will display a confirmation window during operation.

2.5.5 PLC information online query

1) PLC information

The PLC information function acquires and displays various operating data and important PLC information from the PLC.

On the information display window, you can see important information about the current operation of the PLC, such as Figure 2-1 shown.

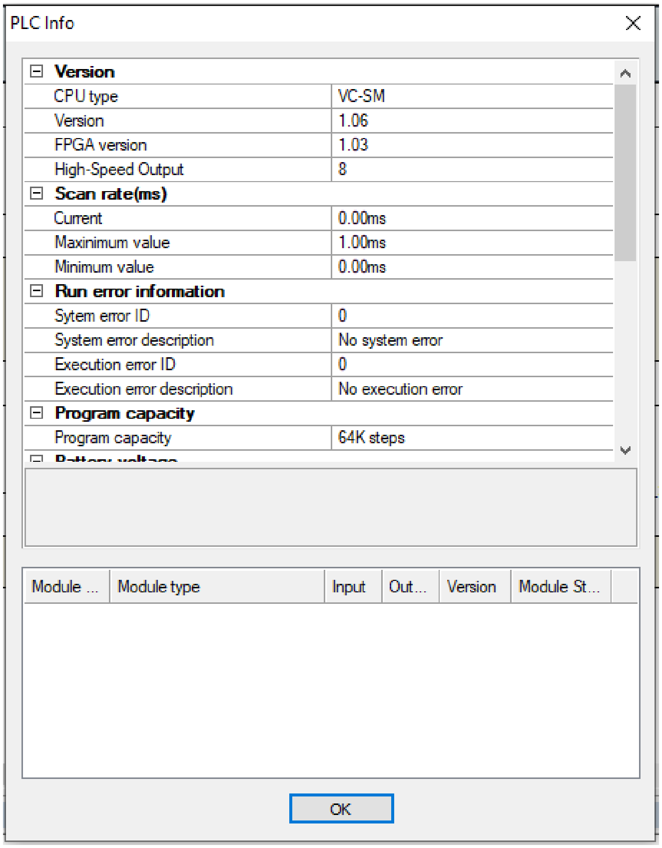


Figure 2-1 PLC current running information

2) PLC time

The PLC time function is used to display and set the current time of the PLC. The PLC time setting dialog box is as follows:Figure 2-2 shown:

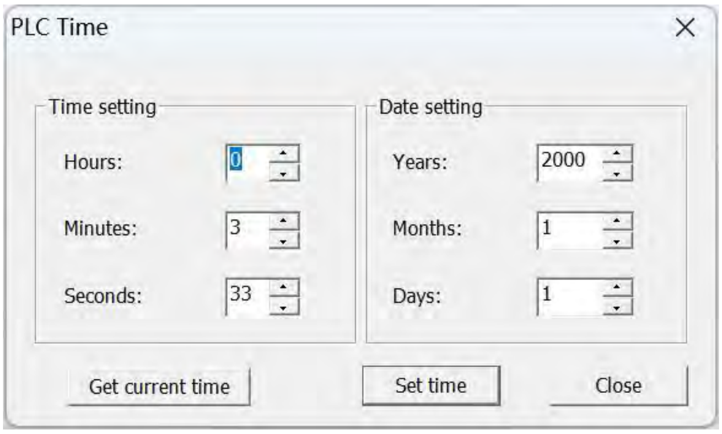


Figure 2-2 Set PLC time

The window displays the date and time currently read from the PLC, you can enter a new date and time, and click the **Set time** button to set the new time to the PLC.

2.5.6 Component value writing and forcing, component monitoring table

A. Component value write and force

① In the debugging process, it may be necessary to manually change the value of some soft components in order to achieve certain conditions. This function is provided by component value writing and forcing. The difference between writing and forcing is that writing the component value is only valid once, and the value after writing may be changed with the running of the program, but the forced component value will always be recorded in the PLC hardware until the forcing is canceled. .

② When you need to execute the write or force function, first select the component to be written or forced, and select **Write** or **Force** from the right-click menu. At this time, a corresponding dialog box will pop up, listing all the device addresses referenced by the selected component. Some soft component values can be selectively written or forced. After confirmation, these values will be sent to the PLC hardware. When these values take effect in the hardware, the change results can be seen in the subsequent debugging process. Write dialog see Figure 2-1:

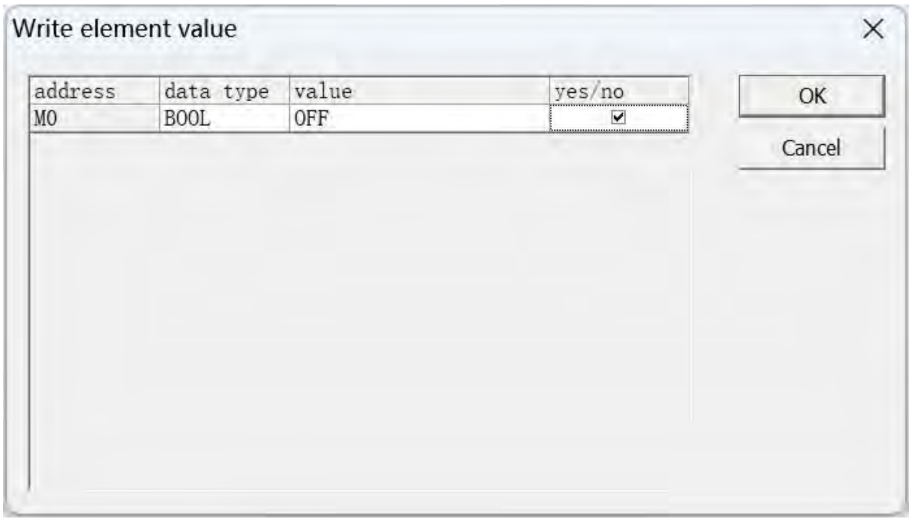


Figure 2-1 write dialog

Force dialog see Figure 2-2:

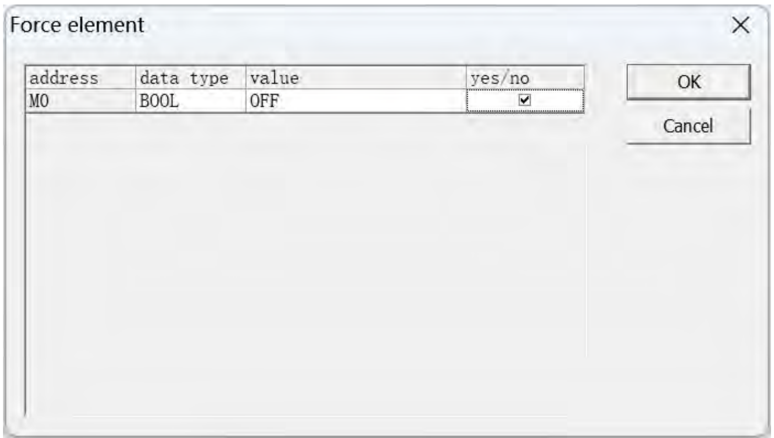


Figure 2-2 Force dialog

Mandatory device, there will be a lock Sign in the ladder diagram, such as Figure 2-3 shown:

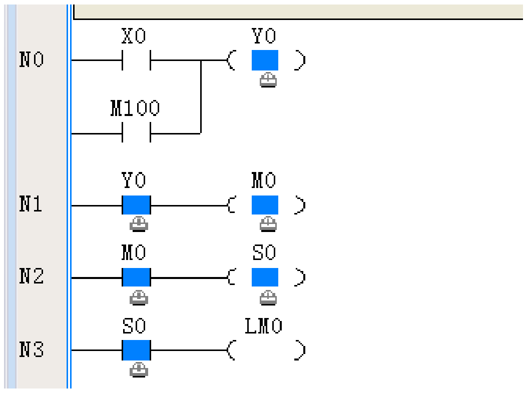


Figure 2-3 Lock Sign of forced device

B. Cancel mandatory

For components that no longer need to be enforced, you can unenforce it. When you need to use the unenforcement function, first select the component to be unenforced, and select **Unforce** from the right-click menu. At this time, a corresponding dialog box will pop up, listing the soft components that have been forced in the selected component, and you can selectively release some of them. For the forced value of the soft element, after clicking OK, these forced values will be deleted from the PLC hardware, and the lock Sign corresponding to the soft element will also disappear. The unforcing dialog box can be found in Figure 2-4.

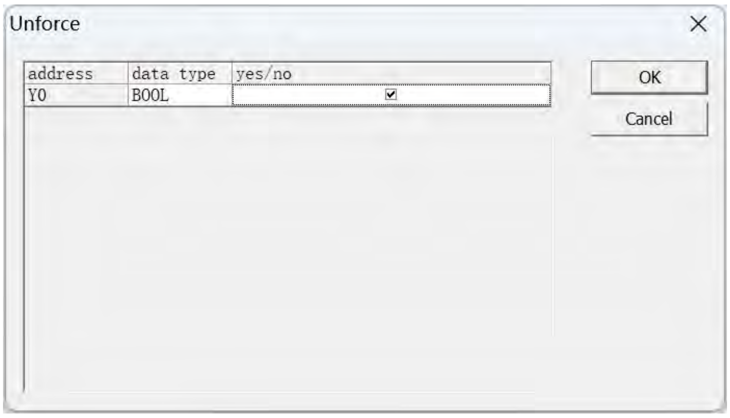


Figure 2-4 Cancel Mandatory Dialog

C. Component monitoring table

The component status monitoring table provides the function of monitoring component values during the debugging process, allowing program input components, output components, register bits and word components to be placed in the component monitoring table to track the status of the program after downloading the program to the PLC .

The component monitoring table has two modes: edit mode and monitor mode.

① In edit mode, all editing functions can be performed, but monitoring functions cannot be performed.

② In monitor mode, monitor functions and editing functions can be performed at the same time.

③ The component status monitoring table will automatically refresh component values in monitor mode. Either the modified component value or the forced component value will be updated in time.

④ The component status monitoring table can provide functions such as editing, sorting, searching, automatically refreshing and displaying the current value of the specified component, writing the component value, forcibly specifying the value of the component/variable, and releasing the force. For the component status monitoring table, please refer to Figure 2-5:

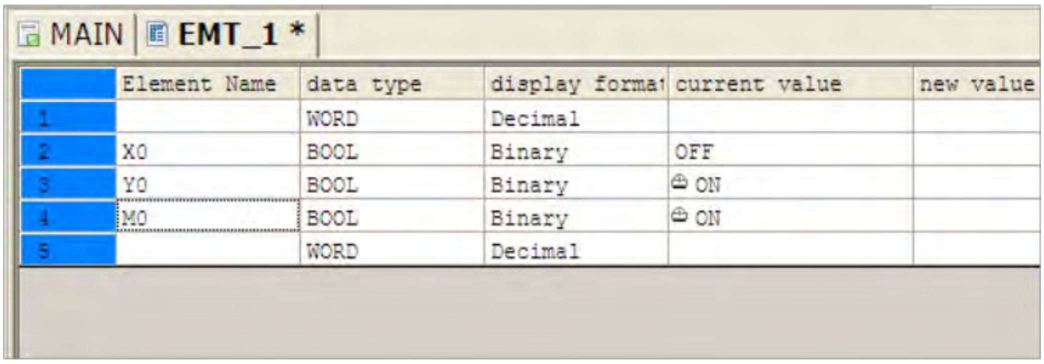


Figure 2-5 Schematic diagram of component status monitoring table

2.5.7 Generate data blocks from RAM

The data values of up to 500 D registers are continuously read from the PLC and displayed, and the results can be merged into a data block or overwritten with the original data block.

Open the Generate Blocks from RAM window, e.g.Figure 2-1shown:

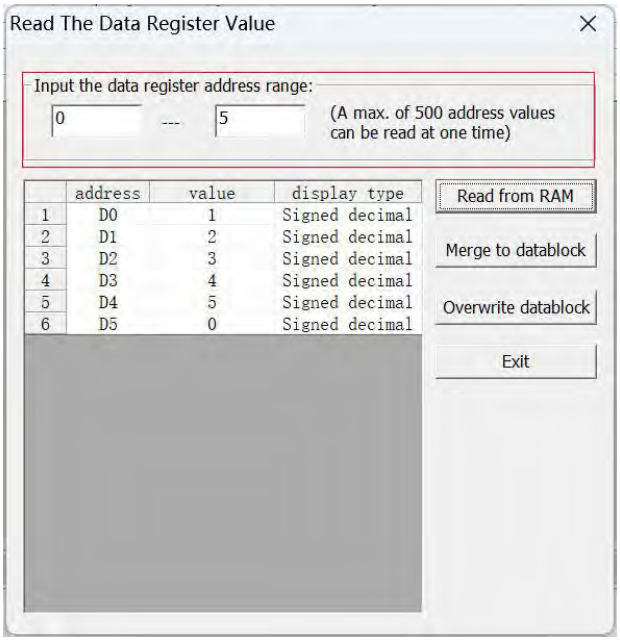


Figure 2-1 RAM generation block window

① Enter the range of the data block to be read, click the **Read from RAM** button, and the data will be read into the list after the execution is correct.

② You can choose to display data in 16, 10, 8, or 2 in the **Display type**.

③ After the reading is successful, the **Merge to Data Block** and **Overwrite Data Block** buttons become available. Click the Merge to Data Block button to append the generated result to the content of the current data block; click the **Overwrite Data Block** button to replace the generated result with the existing data block. content. After exiting the register value reading window, the software prompts that the data block has changed and automatically opens the data block window.

Chapter 3 Devices and Data

3.1 Types and Functions of Software Components

3.1.1 Device overview

PLC configures a variety of virtual components in the system design to replace the real ordinary relays, time relays and other devices in the relay control circuit. These virtual components are collectively referred to as soft components. PLC uses soft components to carry out program operation and system function configuration, so as to realize all operation and control functions. Since the soft element is a virtual element, it can be used repeatedly in the program. There is no theoretical limit on the number (in fact, it is related to the program capacity), and the soft element does not have the mechanical and electrical faults of the real device, so that the PLC's The reliability is much higher than that of the relay control circuit, it is easy to program, and it is more convenient to modify the logic.

The types and functions of the soft components of the VEDA VC series PLC are shown in the figure below.

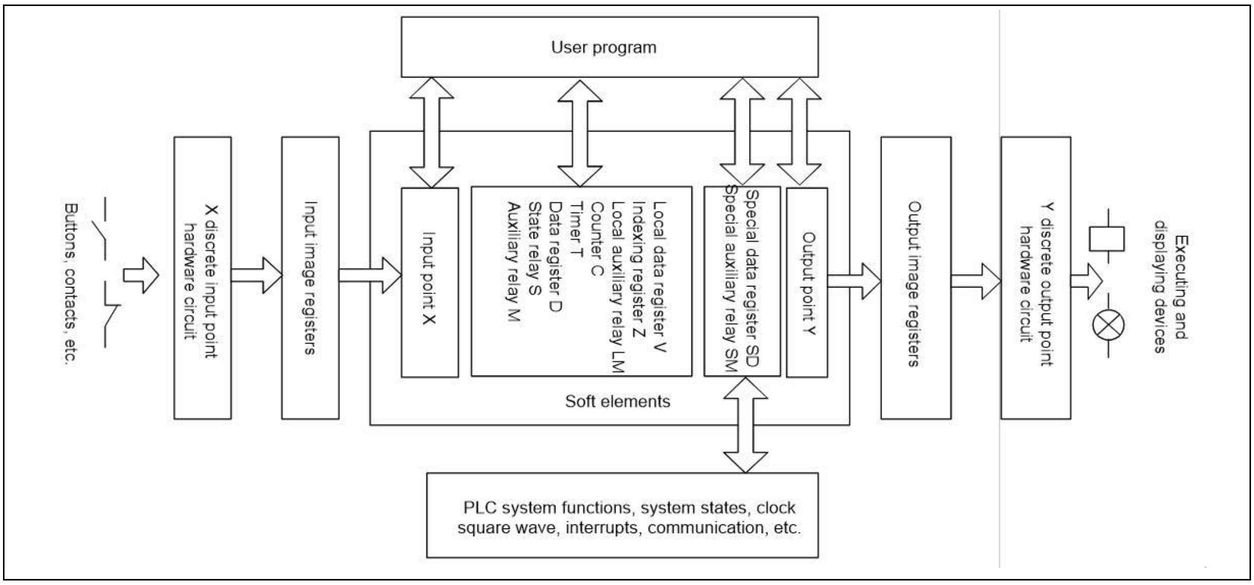


Figure 3-1 Types and functions of PLC soft components

In this manual, the device is abbreviated as "so-so device" according to the type name. E.g:

1) Input point X is simply referred to as "X element"
2) Output point Y is abbreviated as "Y element"
3) Auxiliary relay M is abbreviated as "M element"
4) Data register D is simply referred to as "D element"
5) Status relay S is referred to as "S element" for short

3.1.2 List of devices

The types of soft components of VEDA VC series small PLC are compiled and divided by function, different components perform different functions, and the addressing is simple.

VEDA VC series PLC device list

		VC-B series	VC-S series	Component Addressing Method	Remark
Device resource Note 4	Input and output points	128 in/128 out (input X0～X177, Output Y0 to Y177) Note 1	512 in/512 out (input X0~X777, output Y0~Y777) Note 1	Octal	
	Auxiliary relay	2048 points (M0～M2047)	10240 points (M0～M10239)	10 hex	
	Local auxiliary Relay Note 5	64 points (LM0 to LM63)	64 points (LM0 to LM63)	10 hex	
	Special auxiliary relay	512 points (SM0 to SM511)	1024 points (SM0 to SM1023)	10 hex	
	Status relay	1024 points (S0～S1023)	4096 points (S0～S4095)	10 hex	

	Timer	256 (T0～T255) Note 2	512 (T0～T511) Note 2	10 hex	
	Counter	264 (C0 to C263) Note 3	264 (C0 to C263) Note 3	10 hex	
	Data register	8000 (D0～D7999)	8000 (D0～D7999)	10 hex	
	Data register R	None	32768 (R0～R32767)	10 hex	
	Local data register Note 5	64 (V0～V63)	64 (V0～V63)	10 hex	
	Indexed addressing register	16 (Z0～Z15)	16 (Z0～Z15)	10 hex	
	Special data register	512 (SD0～SD511)	1024 (SD0～SD1024)	10 hex	

Notes:

1: The address numbers of the X and Y components are addressed in octal, and the address X10 represents the 8th input point. The maximum number of input and output points here is the system capacity, and the actual number of hardware points that can be expanded needs to be determined according to the PLC system configuration (including the type and number of available expansion modules, power supply capacity limitations, etc.).

2: The T element addresses are divided into three categories according to the timing accuracy:

VC-B series

- 100ms accuracy T0～T209
- 10ms accuracy T210～T251
- 1ms accuracy T252～T255

VC-S series

- 100ms accuracy T0～T209
- 10ms accuracy T210～T479
- 1ms accuracy T480～T511

3: The C element addresses are divided into three categories according to the width and function of the count value:

- 16-bit up counter C0～C199
- 32-bit up-down counter C200～C235
- 32-bit high-speed counter C236～C263

4: Some PLC internal soft element resources have been reserved for internal use, and such elements should be avoided as much as possible in the user program.

5: These two types of devices are local variables and cannot be defined in the global variable table. When calling the subroutine and returning to the main program, it will be cleared, or the parameter value or status will be obtained according to the interface parameter transfer function

3.1.3 Input and output points

1) Short name

- X element (discrete input points)
- Y element (discrete output point)

2) Effect

They are the soft components representing the input state of the hardware X terminal and the output state of the hardware Y terminal, respectively.

The acquisition of the state of the X element is carried out through the input image register. The output of the Y element state is realized by driving the output circuit through the output image register. These two operations are performed in the I/O refresh phase in the PLC scan cycle model, as shown in Figure 3-2. For details, see**Ошибка! Источник ссылки не найден.**

It can be seen that during the operation of the user program, the PLC response to I/O has a short delay characteristic, which is related to input filtering, communication, housekeeping and scan cycle.

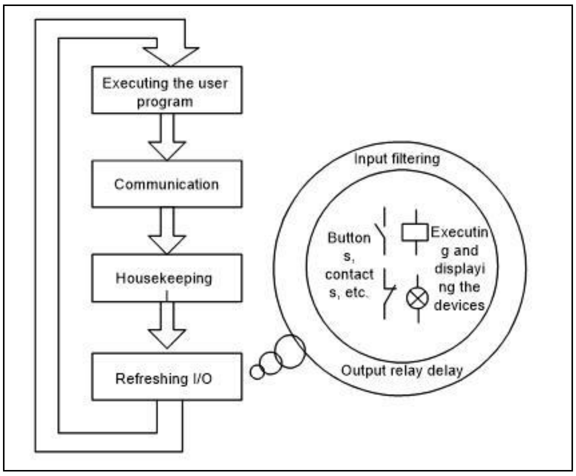


Figure 3-2 I/O refresh principle

3) Classification

Input channels corresponding to X components: X0~X7 have digital filtering function, and the filtering time can be set through the system block; other X input points are hardware filtering. X0~X7 can be used as counting input terminals of high-speed counter soft components; X0~X7 can also be used as input terminals of external interrupt, pulse capture, and SPD frequency measurement instructions. The Y element is divided into a high-speed output terminal and an ordinary output terminal.

4) Addressing method

Octal, starting at address 0. The addressing of the X and Y elements of the main module and the I/O extension module is continuous. For X elements, the continuous addressing is X0~X7, X10~X17, X20~X27... For Y, the continuous addressing is Y0~Y7, Y10~Y17, Y20~Y27....

5) Type of data

Both X and Y elements areboolean element (element value is ON or OFF).

6) Available forms

The normally open contact and normally closed contact of the X element can be used during programming (referenced by two kinds of instructions). The normally open and normally closed contacts have opposite state values, and in some occasions, they are called a contact and b contact respectively.

The normally open and normally closed contacts of the Y element can also be used for programming.

7) Assignment method

1. The X element only accepts the hardware input state and the forced operation state value. It cannot be modified by the output and setting instructions in the user program, nor can it accept the written state value during system debugging.

2. The Y element can be given its status value through the coil output command, and can also be set with a status value. It can also accept forced and written status values during system debugging.

3. The output state of the Y element in the STOP state can be set by the system block.

3.1.4 Auxiliary relay

1) Short name

M element

2) Effect

A discrete state element provided by the system to the user, similar to the intermediate relay in the real electrical control circuit, can be used to save various intermediate states in the user program.

3) Addressing method

Decimal, starting at address 0.

4) Type of data

Boolean (component value is ON or OFF).

5) Available forms

Normally open and normally closed contacts.

6) Assignment method

1. Command operation; 2. Forcing and writing status values during system debugging.

7) Power-down retention

State	M-elements set to hold-down	Non-retentive M element
Power down	Save unchanged	Clear
RUN → STOP	Save unchanged	Save unchanged
STOP → RUN	Constant	Clear
Note: The retentive address range is set by the system block. See2.3.1 System block		

📖 Notice

When using the N: N protocol function, some M components will be called by the system, please pay attention when programming and modifying the program.

3.1.5 Status relay

1) Short name

S element

2) Alias

step status

3) Effect

Mainly used in the programming of sequential function chart, as a sign of stepping state. For details, see Chapter 7Sequential Function Chart.

4) Classification

S0~S19 are the initial step symbols, and the rest are ordinary step symbols.

5) Addressing method

Decimal, starting at address 0.

6) Type of data

Boolean (element value is on or off).

7) Available forms

1. Represents the stepping state (used for programming the STL instruction in the sequential function chart);

2. Normally open contact and normally closed contact (not used when programming STL instruction in sequential function chart). Its characteristics are similar to the M element, and the normally open and normally closed contacts of the S element can be used during programming.

8) Assignment method

1. Command operation; 2. Forcing and writing status values during system debugging.

9) Power-down retention

State	S-elements set for Power-down Retention	Non-retentive S element
power down	Save unchanged	Clear
RUN → STOP	Save unchanged	Save unchanged
STOP → RUN	Constant	Clear
Note: The retentive address range is set by the system block. See2.3.1 System block		

3.1.6 Timer

1) Short name

T element

2) Effect

The T element is a composite type of soft element, which includes a word element (2 bytes) and a bit element. The T-word element records the 16-bit timing value, which can be used as a numerical value in the program; the T-bit element reflects the status of the timer coil and is used for logic control.

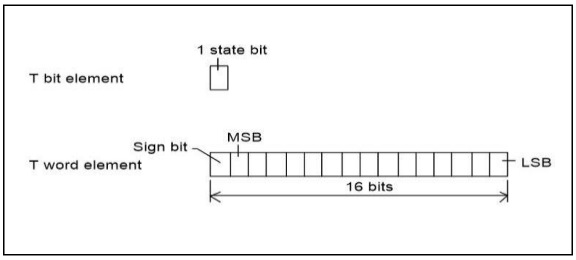


Figure 3-1 T element

3) Classification

There are three kinds of timing precision of T element. The following table shows the T elements of different address segments and their corresponding timing accuracy, which should be paid attention to when using them.

T element	Timing accuracy
VC-B series	100ms accuracy T0～T209 10ms accuracy T210～T251 1ms accuracy T252～T255
VC-S series	100ms accuracy T0～T209 10ms accuracy T210～T479 1ms accuracy T480～T511

For the T element with a timing accuracy of 1ms, its timing is an interrupt trigger and has nothing to do with the PLC scan cycle, so the timing action time is the most accurate. For T elements with timing accuracy of 10ms and 100ms, the refresh and action time of the timing value is related to the PLC scan cycle.

4) Addressing method

Decimal, starting at address 0.

5) Type of data

Boolean (element value is on or off), character.

6) Available forms

The timing and behavior of the T element depends on the timing instruction that invokes it. There are 4 kinds of instructions: ON-delay timing instruction, OFF-delay timing instruction, memory type ON-delay timing instruction, and non-retrigger monostable timing instruction. For a description of these 4 commands seeChapter 5Basic Instructions

7) Assignment method

1. Command operation; 2. Forcing and writing status values during system debugging.

8) Power-down retention

State	T-element set to hold-down (VC2/3/5 series only)	Non-retentive T-element
power down	save unchanged	clear
RUN → STOP	save unchanged	save unchanged
STOP → RUN	constant	clear
Note: The retentive address range is set by the system block. See2.3.1 System block		

📖 Notice

The maximum timing value of T element is 32767, and the default value is -32768～32767. Since the T element is actuated by the timing value greater than or equal to the preset value, it is meaningless to set the preset value to a negative number.

3.1.7 Counter

1) Short name

C element

2) Effect

The C element is a composite soft element, which includes a bit element and a single-word or double-word element (2 bytes or 4 bytes). The C word element records the 16-bit or 32-bit count value, and the C-bit element reflects the status of the counter coil. The C word element can be used as a numerical value in the program, and the C bit element is used for logic control.

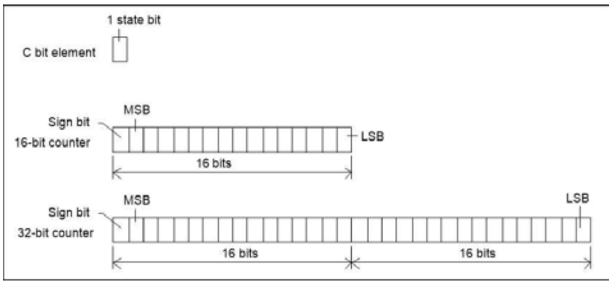


Figure 3-1 C element

3) Classification

There are two types of 16-bit counters and 32-bit counters.

4) Addressing method

Decimal, starting at address 0.

5) Type of data

Boolean (element value is on or off), single word or double word.

6) Available forms

There are four types of counting instructions for calling C components, which are 16-bit up-counter instructions, 16-bit loop counting instructions, 32-bit increment and decrements counting instructions, and high-speed I/O instructions. For a description of these 4 types of instructions seeChapter 5Basic InstructionsandChapter 6Application Instruction. C components are classified as shown in the table below.

C element	Count function	Applicable instruction types
C0～C199	16-bit up counter	16-bit count up instruction 16-bit loop count instruction
C200~C235	32-bit up-down counter	32-bit increment and decrements counting instructions
C236~C263	32-bit high-speed counter	High-speed I/O instructions

7) Assignment method

1. Command operation; 2. Forcing and writing status values during system debugging.

8) Power-down retention

State	C Elements Set to Retentive	Non-retentive C element
power down	Save unchanged	Clear
RUN → STOP	Save unchanged	Save unchanged
STOP → RUN	Constant	Clear
Note: The retentive address range is set by the system block. See2.3.1 System block		

3.1.8 Data register

1) Short name

D element, R element

2) Effect

As data elements, many operations and control instructions use d or r elements as operands.

3) Addressing method

Decimal, starting at address 0.

4) Type of data

Each D or R element is a 16-bit register that can store 16-bit data, such as a 16-bit integer.

Two D or R elements can be combined into a double word element for storing 32-bit data such as long or floating point data.

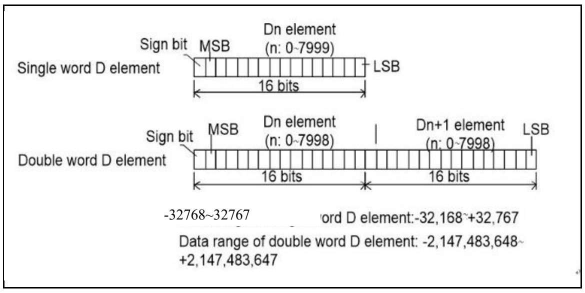


Figure 3-1 D or R element

📖 Notice

In the double word D or R element, the upper 16 bits are in the first D or R element, and the lower 16 bits are in the second D or R element.

5) Available forms

Many operations and control instructions use d or r elements as operands.

6) Assignment method

1. Data block initialization; 2. Instruction operation; 3. Force and write status values during system debugging.

7) Power-down retention

State	D Elements Set to Retentive	Non-retentive D element
power down	Save unchanged	Clear
RUN → STOP	Save unchanged	Save unchanged
STOP → RUN	Constant	Clear
Note: The retentive address range is set by the system block. See2.3.1 System block *The R element cannot be saved after power down*		

📖 Notice

When using inverter command, N: N protocol and other functions, some D components will be called by the system, users should pay attention when programming and modifying the program.

3.1.9 Special auxiliary relay

1) Short name

SM element

2) Effect

SM components are soft components closely related to PLC system functions. SM components reflect the PLC system function and status. For a detailed functional description of all SM components, please refer to this manual Chapter 13 Classification

Commonly used components of this type are:

① SM0: Monitor running bit, keep ON state in RUN state.

② SM1: Initial running pulse bit, ON during the first scan cycle of running.

③ SM3: System error, ON when a system error is detected after power-on or from STOP to RUN.

④ SM10～SM12: 10ms, 100ms, and 1s clock oscillation square waves, respectively, flip once every half cycle. The state modification of some SM components can also call, control, and change the PLC system functions. Commonly used components of this type are:

- SM25～SM71: Interrupt control Flag bit, setting these SM components can enable the corresponding interrupt function.

3) Addressing method

Decimal, starting at address 0.

4) Type of data

boolean (element value is ON or OFF).

5) Available forms

Normally open and normally closed contacts.

6) Assignment method

1. Command operation; 2. Forcing and writing status values during system debugging.

For a detailed functional description of all SM components, please refer to this manual 0 Special auxiliary relay.

📖 Notice

A read-only SM element cannot be assigned a value.

3.1.10 Special data register

1) Short name

SD element

2) Effect

Soft components closely related to PLC system functions reflect PLC system function parameters, status code values, and command operation data. For a detailed functional description of all SD components, please refer to this manual0special data register.

3) Addressing method

Decimal, starting at address 0.

4) Type of data

Word, double word (integer) elements.

5) Available forms

Integer storage and operation.

6) Assignment method

1. Command operation; 2. Forcing and writing status values during system debugging.

📖 Notice

Read-only SD elements cannot be assigned values.

3.1.11 Indexed addressing register

1) Short name

Z element

2) Effect

16-bit register element that can store signed integer data. For Indexed addressing, see3.1.15 Indexed addressing mode (Z addressing mode).

3) Addressing method

Decimal, starting at address 0.

4) Type of data

character element.

5) Available forms

Used for Indexed addressing functions. To use the Z element, first write the data of the address offset to the Z element.

6) Assignment method

1. Command operation; 2. Force and write status values during system debugging.

3.1.12 Local auxiliary relay

1) Short name

LM element

2) Effect

LM elements are local variables. LM elements can be used in main programs and subprograms. They are locally effective variable elements in each independent program body (main program, subprogram and interrupt program), therefore, the state of any LM element cannot be directly shared between different program bodies. When a program body is left in the execution of the user program, the system will redefine the LM element. When returning to the main program or calling a subroutine, the value of the redefined LM element will be cleared, or according to Interface parameter passing function to obtain the corresponding status.

Interface parameters that can be used to define subroutines, implementing Interface parameter passing function. For details, please refer to4.4 Subroutine.

3) Addressing method

Decimal, starting at address 0.

4) Type of data

Boolean (element value is ON or OFF).

5) Available forms

normally

open and normally Closed Contacts.

3.1.13 Local data register

1) Short name

V element

2) Effect

The V element is a local variable. V elements can be used in main programs and subprograms. They are locally valid variable elements in each independent program body (main program and subprogram), therefore, the data of any V element cannot be directly shared between different program bodies. When a program body is left in the execution of the user program, the system will redefine the V element. When returning to the main program or calling a subroutine, the value of the redefined V element will be cleared, or according to Interface parameter passing function to obtain the corresponding data.

V elements can be used to define interface parameters of subroutines, implementing-interface parameter passing function. For details, please refer to4.4 Subroutine.

3) Addressing method

Decimal, starting at address 0.

4) Type of data

Boolean (element value is ON or OFF).

5) Available forms

Word element, which can save information of numerical type.

6) Assignment method

1. command operation;

Soft Component Addressing Method

3.1.14 Bit string combination addressing mode (Kn addressing mode)

A. Bit string combination addressing mode concept

The bit string combination addressing mode (Kn addressing mode) is used to combine bit element strings into words or long words.

B. Bit string combinatorial Addressing Method

The combined addressing format of the bit string is K(n)(U), where n is an integer from 1 to 8, indicating that the length of the element string is n×4 bits. U represents the start bit element address of the element string.

Concrete example:

1. K1X0 represents: a word composed of a 4-bit long bit string (X0, X1, X2, X3).

2. K3Y0 represents: 12-bit long bit string (Y0, Y01, Y02, Y03), (Y04, Y05, Y06, Y07), (Y10, Y11, Y12, Y13) to form a word for use.

3. K4M0 represents: 16-bit long bit strings M0, M1, M2, M3..., M15 form a word for use.

4. K8M0 represents: 32-bit long bit strings M0, M1, M2, M3..., M31 form a double word for use.

C. Kn addressing mode data storage format

An example to illustrate how a specific data is stored in the Kn addressing mode:

MOV 2#10001001 K2M0 (equivalent to MOV 16#89 K2M0 or MOV 137 K2M0). When this command is executed, the specific storage format of K2M0 is shown in the following table:

Data	Highest bit	Middle position						Lowest bit
K2M0	M7	M6	M5	M4	M3	M2	M1	M0
16#89	1	0	0	0	1	0	0	1

D. Bit string Combinatorial Addressing Considerations

If the Destination operand of the instruction uses Kn addressing mode, and the data width that needs to be stored to the Destination operand is larger than the width specified by Kn addressing, the system stores the data according to the rules of retaining the low-order part and discarding the high-order part.

The following example illustrates this situation:

Execute the instruction "DBITS 16# FFFFFFF0 K1M0".

After the instruction is executed, the operation result that should be stored in operand 2 (K1M0) is 16#1c (28), but because the width of the data that K1M0 can store is 4, the operation result 16#1c cannot be completely stored. The part will be rounded off, so the actual result of result operand 2 is: K1M0=16#c(12).

3.1.15 Indexed addressing mode (Z addressing mode)

1) Indexed addressing concepts

VEDA VC series PLC provides Index addressing mode (Z addressing mode), users can use Z components (Indexed Addressing Register), to achieve the purpose of indirect addressing access to components.

2) How to use the Z addressing mode:

The target address of Index addressing = the base address of the element + the address offset stored in the Z element.

For example:

Indexed addressing D0Z0 (where Z0=3), indicating that D0 is the base address of Indexed addressing, the address offset of Indexed addressing is stored in Z0 (the address offset is equal to 3), and the target address should be D3

Therefore, in the case of Z0=3, the two instructions "MOV 45 D0Z0" and "MOV 45 D3" are equivalent, and D3 will be assigned 45 after the instruction is executed effectively.

3) Indexed addressing example

1. Bit Element Indexed Addressing Example

LD M01

MOV 6 Z1

SFTR X0Z1 M0 8 2

The above command is actually equivalent to:

LD M0 1

SFTR X6 M0 8 2

The addressing process is as follows:

Z1=6

X0Z1 = X(0+Z1) = X6

2. Word element indexed addressing example

LD M0 1

MOV 30 Z20

MOV D100Z20 D0

The above command is equivalent to:

LD M0 1

MOV D130 D0

The addressing process is as follows:

Z20=30

D100 Z20 = D (100 + Z20) = D130

4) Notes on Indexed Addressing

1. In the Indexed addressing mode (Z addressing mode), the Z element stores the address offset, which is always treated as a signed integer by the system, that is, the Z addressing mode supports negative address offsets.

For example:

MOV-30 Z20

MOV D100Z20 D0

The above command is equivalent to:

MOV D70 D0

2. SM element and SD element do not support Index addressing mode.

3. When using the Z addressing mode, the user should avoid the Z addressing out-of-bounds situation, for example: D7999Z0 (where Z0=9) has the Z addressing out-of-bounds situation (the maximum address of the D element is D7999).

3.1.16 Indexed addressing with bit string combination

1) The bit string combination addressing mode can also be used in conjunction with the Index addressing mode, that is, in the form of K1X0Z10. This addressing mode first determines the address of the starting bit element of the bit string

combination through Z addressing, and then determines the length of the bit string through Kn addressing.

2) The following example illustrates the specific addressing process:

LD M1

MOV 3 Z10

MOV K1X0Z10 D0

The above command is equivalent to:

LD M1

MOV K1X3 D0

The addressing process is as follows:

Z10=3

K1X0Z10=K1X(0＋Z10) = K1X3

3.1.17 Storage and addressing of 32-bit data by D, R, V elements

1) The storage method of 32-bit data in D, R, V elements

The data of DINT and REAL types are all 32-bit wide, and a D, R or V element is only 16-bit wide, so two D, R or V elements with consecutive addresses are required to store 32-bit data.

VEDA VC series PLC uses the Big Endian method to store 32-bit data, that is, the components with small address numbers are used to store the high word of 32-bit width data, and the components with large address numbers store the low word of 32-bit width data.

For example: the unsigned long integer data 16# FEA8_67DA is stored in the (D0, D1) element, and its actual storage format is as follows:

D0	0xFEA8
D1	0x 67DA

2) D, R, V element address addressing 32-bit data

A D, V element address can address a 16-bit data (such as INT type data), can also address a 32-bit data (such as DINT type data).

If the instruction operand refers to a D, R address, or a V element address, then whether the address represents a 16-bit data or a 32-bit data will be determined by the data type of the operand.

For example: in the instruction "MOV 16#34 D0", the address D0 only addresses the single D0 element, because the data type of the operand 2 of the MOV instruction is INT. In the instruction "DMOV 16# FEA867DA D0", the address D0 represents the two consecutive word elements D0 and D1 starting from D0, because the data type of the operand 2 of the DMOV instruction is DINT type.

3.2 Data

3.2.1 Type of data

The operands of the instruction all have data type attributes, and four data types are supported, as shown in the following table.

The data type of the operand

Type of data	Type description	Data width	Scope
BOOL	Bit	1	On, off (1, 0)
INT	Signed integer	16	-32768～32767
DINT	Signed long integer	32	-2147483648～2147483647
REAL	Floating point number	32	±1.175494e-38～±3.402823e+38

3.2.2 Component and data type matching relationship

The component type selected by the instruction operand should maintain a certain matching relationship with the data type. The matching relationship between the applicable components and data types is shown in the following table.

Matching relationship between components and data types

Type of data	Device													
BOOL										C	T			
	X	Y	M	S	LM	SM								
INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R
DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R
REAL	Constant							D				V		R

If the programming of the instruction does not satisfy the matching relationship, the instruction will be regarded as illegal. For example, the instruction "MOV 10 X0" is illegal. This is because the data type of the operand 2 of the MOV instruction is a signed integer type, and the element X0 can only store bits type of data.

📖 Description

1. When the data type of the operand is INT, the applicable soft elements are KnX, KnY, KnM, KnS, KnLM, KnSM, 1≤n≤4.
2. When the data type of the operand is DINT, the applicable soft elements are KnX, KnY, KnM, KnS, KnLM, KnSM, 5≤n≤8.
3. When the data type of the operand is INT, the number of the applicable C element should be C0~C199.
4. When the data type of the operand is DINT, the address of the applicable C element should be C200~C263.

3.2.3 Constant

Users can use constants as the operands of instructions. VEDA VC series PLC supports various input methods of constants. The expressions of constants are shown in the following table:

Constant expression

Constant type	Expression example	Effective range	Illustrate
Constant decimal 16-bit signed integer	-8949	-32768～32767	/
Constant decimal 32-bit signed integer	-2147483646	-2147483648～2147483647	/
16-bit Constant in hexadecimal	16#1FE9	16#0～16#FFFF	Hexadecimal, octal, and binary constants have no positive or negative meaning. Hexadecimal, octal or binary constants are selected as instruction operands. The positive, negative and size of the operands are determined according to the data types of the operands.
Hexadecimal 32-bit Constant	16#FD1EAFE9	16#0～16#FFFFFFFF	
Octal 16-bit Constant	8#7173	8#0～8#177777	
Octal 32-bit Constant	8#71732	8#0～8#37777777777	
Binary 16-bit Constant	2#10111001	2#0～2#1111111111111111	
Binary 32-bit Constant	2#101110011111	2#0～2#1111111111111111 1111111111111111	
Single-precision floating-point Constant	-3.1415E-16 3.1415E+3 0.016	±1.175494E-38～±3.402823E+38	Compliant with IEEE-754 standard. The programming software can display and input floating-point constants with 7-digit effective precision

Chapter 4 Programming Concepts

4.1 Introduction to Programming Languages

There are three programming languages: Ladder Diagram (LAD), Instruction List (IL), and Sequential Function Chart (SFC).

4.1.1 Ladder Diagram (LAD)

A. Ladder Diagram Concept

Ladder diagram is a graphical PLC programming language similar to electrical (relay) control diagrams, and is a widely used PLC programming language. Its main features include:

1. With the left busbar, while the right busbar is omitted.

2. All control output elements (coils) and function blocks (application commands) have only one power flow input.

There is a certain equivalent relationship between the electrical control diagram and the ladder diagram, as shown in the following figure:

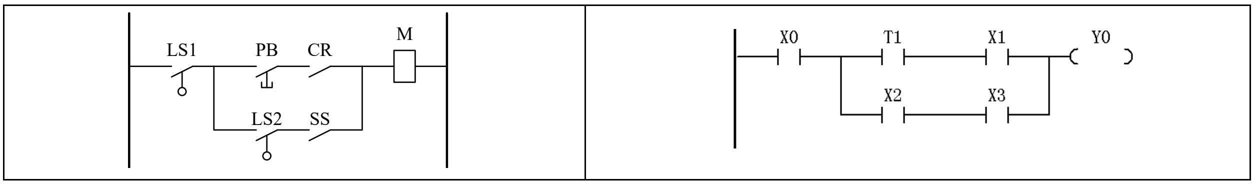


Figure 4-1 Equivalent relationship between electrical control diagram and ladder diagram

B. Ladder Diagram Basic Programming Elements

Ladder diagram abstracts several basic programming elements according to the principle of electrical (relay) control diagram:

1. Left bus: Corresponding to the control bus in the electrical control diagram, providing control power for the control loop.

2. Connecting line ( ): Represents the electrical connections of the electrical control diagram, which are used to conduct other components connected to each other.

3. Contact ( ): represents the input contact in the electrical control diagram, controls the on-off of the control current in the loop, and determines the direction of the control current. The parallel and series connection of the contacts essentially represents the operation relationship of the input logic of the control circuit, which controls the transfer of energy flow.

4. Coil ( ): Represents the relay output in the electrical control diagram.

5. Function block( ): also known as application instructions, corresponding to the actuators or functional devices connected in the electrical control diagram to complete special functions, function blocks can complete specific control functions or control calculation functions (such as data transmission, data operations, timers, counters, etc.).

C. Energy Flow

The energy flow is a very important concept in the ladder diagram program. The energy flow is used to drive the coil components and application instructions, which is similar to the control current of the drive coil output and the mechanism execution in the electrical control diagram.

In the ladder diagram, the front end of the coil or application instruction must be connected to the power flow. When the power flow is valid, the coil element can be output and the application instruction can be effectively executed.

The following figure demonstrates the power flow transfer in the ladder diagram and the driving effect of the power flow on the coil or function block.

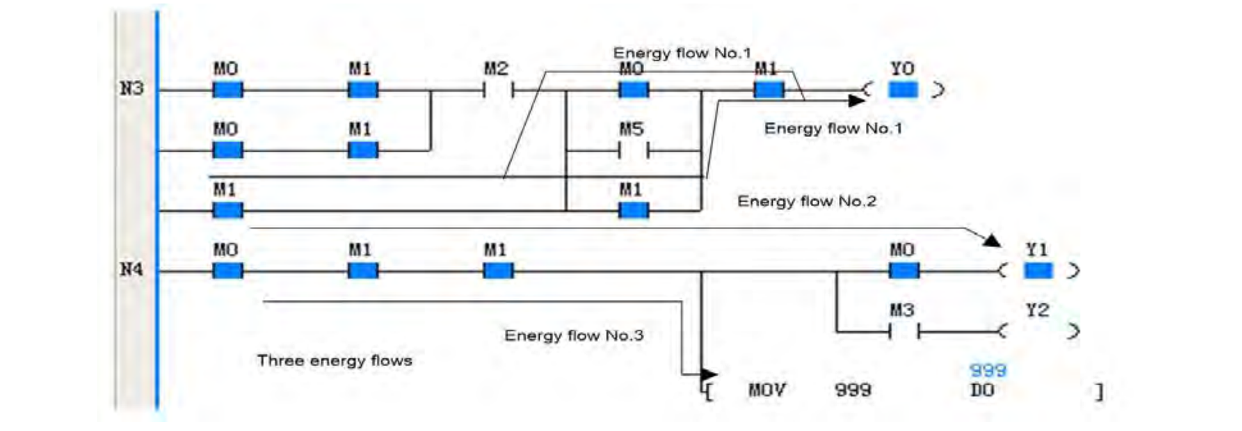


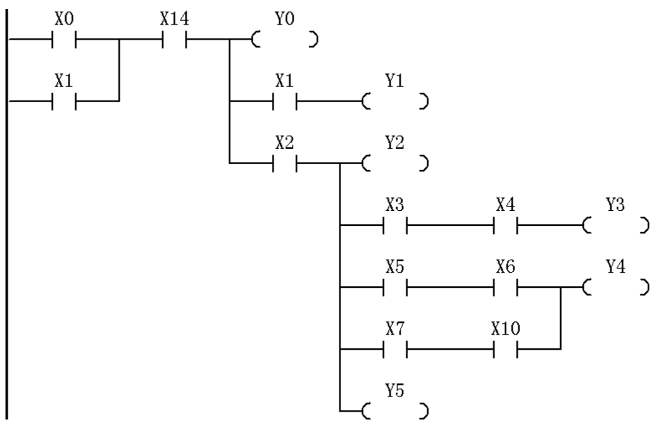
Figure 4-2 Energy flow transfer and driving effect of energy flow

4.1.2 Instruction List (IL)

The instruction list is a textual user program, which is a set of instruction sequences written by the user.

The user program stored in the PLC main module for execution is actually an instruction sequence identifiable by the main module. The system executes each instruction in the sequence one by one to realize the control function of the user program.

The following figure is an example of converting a ladder diagram into an instruction list.

Ladder Diagram	Command list
	LD -----X0 OR -----X1 AND -----X14 MPS OUT -----Y0 AND -----X1 OUT -----Y1 MPP AND -----X2 MPS OUT----- Y2 AND -----X3 AND -----X4 OUT -----Y3 MRD LD -----X5 AND -----X6 LD -----X7 AND -----X10 ORB ANB OUT -----Y4 MPP OUT -----Y5

4.1.3 Sequential Function Chart (SFC)

Sequential function chart is a graphical user programming framework design language, which is usually used to implement sequential control functions.

Sequence control refers to a control process that can be divided into multiple processes (processing steps) and processed in a certain working order.

The user program designed according to the sequence function diagram, the program structure is consistent with the actual sequence control process, and is more intuitive and clear.

The following figure is an example of a simple sequential function diagram.

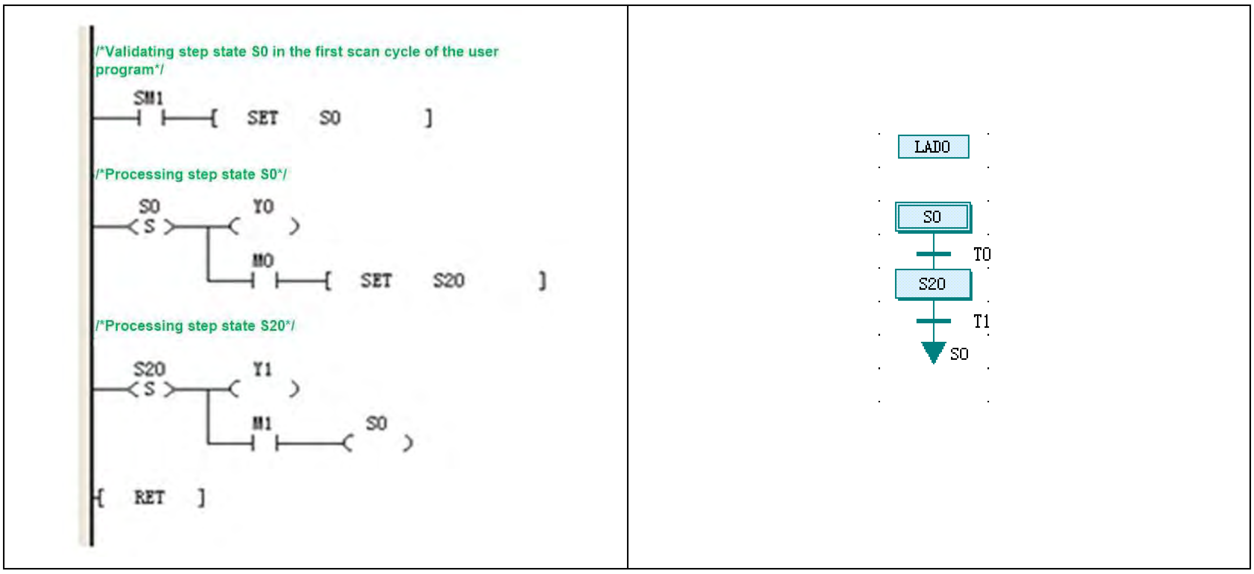


Figure 4-1 Sequential Function Chart Example

4.2 Program Elements

User programs, system blocks and data blocks are called program elements. The user can modify three program elements through programming.

4.2.1 User program

The user program is the program code written by the user, which is compiled into an executable instruction sequence, downloaded to the controller, and the controller executes the control function of the user program.

User program consists of main program, subprogram and interrupt program three types of program body (POU).

1) Main program (MAIN)

The main program is the main body and frame of the user program. When the system is running, the main program is executed cyclically.

Any user program has one and only one main program.

2) Subroutine (SBR)

A subprogram is a user program that is independent in structure and function and can be called by other program bodies. It usually has a calling operand interface and is executed only when it is called.

A user program can have no subprograms, or it can contain one or more subprograms.

3) Interrupt routine (INT)

An interrupt routine is a section of user program that handles specific interrupt events. A specific interrupt event always corresponds to a specific interrupt routine.

As long as an interrupt event occurs, a normal scan cycle will be interrupted, the user program flow will automatically jump to the interrupt program execution, and the system will resume the normal scan cycle process until the interrupt return instruction is executed.

A user program can have no interrupt routine, or it can contain one or more interrupt routines.

4.2.2 System block

The system block contains multiple system configuration options. Users can modify, compile and download the system block to achieve the purpose of configuring the operating mode of the main module.

For details on how to use the system configuration items, please refer to this manual2.3.1 System block, or refer to the introduction about system blocks in 《VEDA PCT Programming Software User Manual》.

4.2.3 Data block

The data block contains the setting value of D or R element. When the data block is downloaded to the controller, the specified D or R element will be assigned the setting value, so as to achieve the purpose of batch setting the value of D or R element.

If the controller is configured in the data block valid operating mode, the D or R element specified in the data block will be initialized according to the content of the data block before the user program is run.

4.3 Program Block Comments and Variable Comments

4.3.1 Block comment

When programming, you can add block comments in the program, and block comments describe a certain section of the program in text. Each block comment takes up an entire line of space.

Right-click the mouse in the program, open the right-click menu, and select **Line Insert**, you can insert a blank line in the program. Generally, blank lines should be used as block boundaries.

When you need to enter a block comment, first select a blank line, and then select **Switch Insert/Overwrite Mode** from the right-click menu, as shown in the following figure:

Figure 4-1 Add block comments

Enter the comment text in the pop-up block comment dialog box and confirm, as shown in the following figure:

Figure 4-2 Block Comment Input Dialog

The software will automatically add "/*" and "*/" on both sides of the entered text, and display them in green, as shown in the following figure:

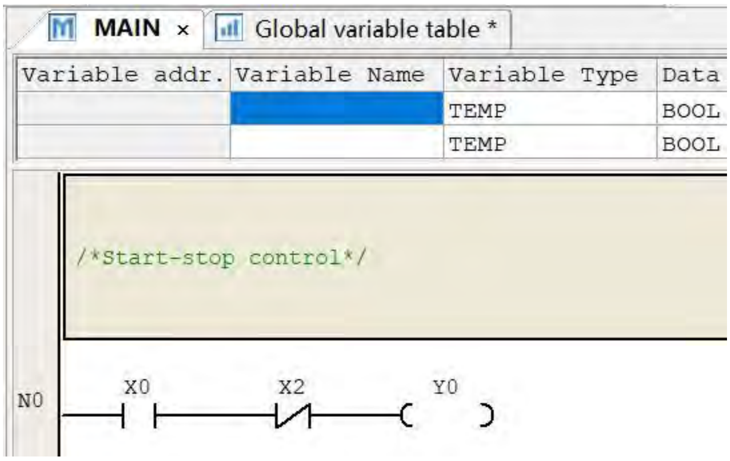


Figure 4-3 block comments in the program

Since block comments will occupy the entire line space, if there are other components in a line, you cannot enter block comments on this line; similarly, lines already occupied by block comments cannot enter any other components.

4.3.2 Comments for variables

Variables can be defined in the global variable table and the local variable table (for specific definition methods, see2.3.3 Global variable tableand4.4.3 Subroutine variable table definition), correctly defined variables can be used in the ladder diagram. When a certain address needs to be used, the variable name representing the address can be used to enhance the readability of the program. The following figure shows the variables defined in the global variable table:

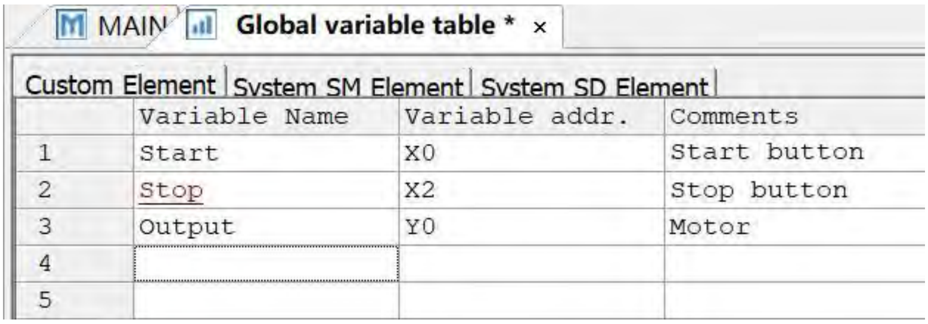


Figure 4-1 Variables defined in the global variable table

4) Symbolic addressing

After using the defined variable, you can switch between the variable name and the component address by selecting the **Symbol addressing** menu. The following figures show the same ladder program in the two display modes:

States with unchecked symbolic addressing:

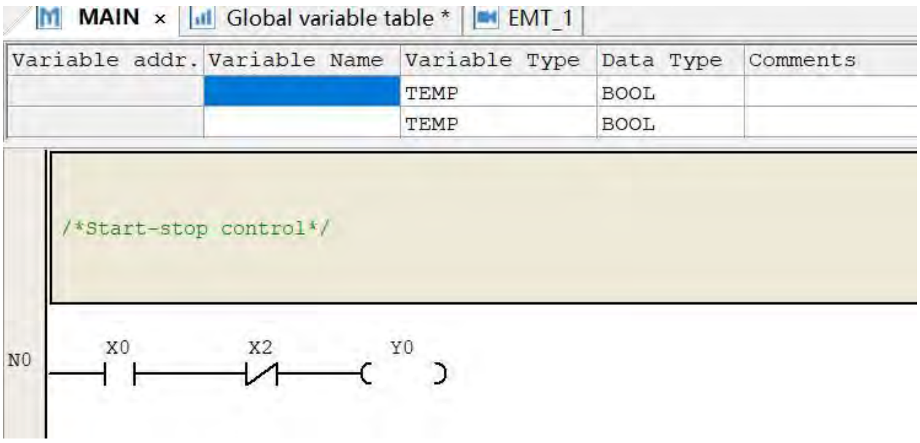


Figure 4-2 The state where symbolic addressing is not selected

With symbolic addressing:

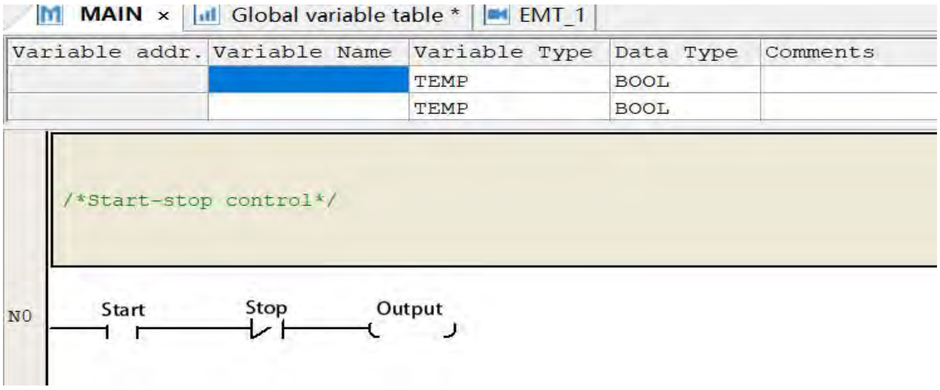


Figure 4-3 using symbolic addressing

5) Component notes

You can control whether to display component comments in the ladder program by selecting the component **comment menu**. The following is the ladder program when the component comment is displayed:

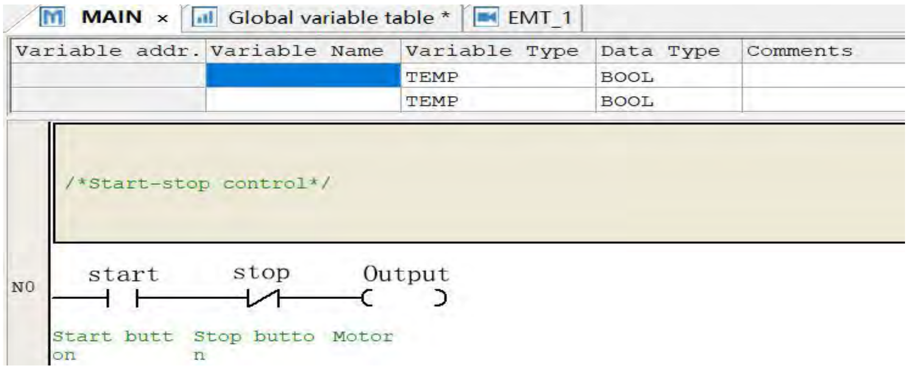


Figure 4-4 Ladder program when component comment is displayed

4.4 Subroutine

4.4.1 Subroutine concept

A subprogram is an independent program body that can be called by the main program or other subprograms. Subroutines are optional components of a user program.

Writing user programs with subroutines has the following advantages:

1. It can reduce the size of the user program, and the repeated user program code segment with the same function can be written as a subprogram to be called repeatedly.

2. Make the structure of the program clearer, especially the main program structure can be simplified.

3. Improve the portability of user programs.

4.4.2 Precautions for the use of subroutines

When writing or calling subroutines, pay attention to the following:

A. Mosaic calling of subroutines is supported, and the maximum number of mosaic calling layers is 6.

The following example demonstrates a valid 6-level mosaic call relationship:

MAIN→SBR1→SBR2→SBR3→SBR4→SBR5→SBR6.

(→represents calling the corresponding subroutine with the CALL instruction)

B. Recursive and cyclic calls to subroutines are not supported.

The following two examples demonstrate the illegal subroutine call relationship:

① MAIN→SBR0→SBR0 (recursive call, illegal)

② MAIN→SBR0→SBR1→SBR0 (loop call, illegal)

C. A maximum of 64 subroutines can be defined in a user program.

D. A maximum of 16 bit type and 16 word type variables can be defined in the variable table of a subroutine.

E. When calling a subprogram, it should be noted that the attributes of the operand filled in the CALL instruction should match the variable attributes defined in the variable table of the subprogram, and the compiler will check the correctness of the matching.

F. No subroutine calls are allowed in an interrupt routine.

4.4.3 Subroutine variable table definition

A. Subroutine variable table

The function of the subprogram variable table is to declare the interface parameters and local variables (collectively referred to as variables) of the subprogram, and to specify their usage attributes.

B. Description of attribute items of subroutine variables

The interface parameters and local variables of a subroutine (collectively called variables) have the following properties:

1. Variable address

Each subroutine interface parameter or local variable is assigned a fixed LM element or V element address. The address is automatically assigned to the subprogram interface parameter or local variable by the programming software according to the data type of the variable and the principle of continuous address.

2. Variable name

You can take a variable name (alias) for the subprogram interface parameter or local variable, and you can use the variable in the program by using the variable name reference.

3. Variable type

Subprogram interface parameters or local variables are divided into IN type, OUT type, IN_OUT type, and TEMP type:

① IN type variables are used to transfer the input value of the subroutine when the subroutine is called.

② The OUT variable is used to pass the return value of the subroutine call when the subroutine returns.

③ IN_OUT variables are used to pass input values when the subroutine is called. When the subroutine returns, it is used to pass the return value of the call.

④ Variables of type TEMP are used only as valid local variables within the scope of the subroutine.

4. Variable data type

The variable data type attribute specifies the data width and data range of the variable. The following table lists the types of variable data types:

Kind of variable data type

Variable data type	Data Type Description	Occupy LM/V component address
BOOL	Bit variable	Occupies 1 LM element address
INT	Signed integer variable	Occupies 1 V component address
DINT	Signed long integer variable	Occupies 2 consecutive V element addresses
REAL	floating point variable	Occupies 2 consecutive V element addresses

4.4.4 Subroutine parameter passing

When calling a subprogram in the main program, if the local input and output variables are defined in the subprogram, the interface parameters of the subprogram must be filled with corresponding values or global/temporary variable elements. Note that the data types of local variables and interface parameters should be consistent.

4.4.5 Example of the use of subroutines

The following shows how to write and call subroutines with an example

1) Sample function introduction

Call the subroutine SBR_1 in the main program, let the subroutine SBR_1 complete the addition operation of two integer constants (10+5), and assign the operation result 15 to D2.

2) Example operation procedure

Step 1: Create a subprogram in the project and name the subprogram SBR_1.

Step 2: Write subroutine SBR_1

1. The call operand interface of the subroutine is established in the variable table of the subroutine SBR_1.

1) Define variable 1: Take the variable name as Number1, which is an IN-type parameter and is used as INT-type data, which is sequentially assigned a V element address V0.

2) Define variable 2: Take the variable name as Number2, which is an IN-type parameter and is used as INT-type data, which is sequentially assigned a V element address V1.

3) Define variable 3: Name the variable SumResult, which is an OUT type parameter and is used as INT type data, which is sequentially assigned a V element address V2.

2. Write the implementation code of the subroutine SBR_1:

```
LD    SM0
ADD   # Number1   # Number2   # SumResult
```

The following figure demonstrates the writing process of subroutine SBR_1:

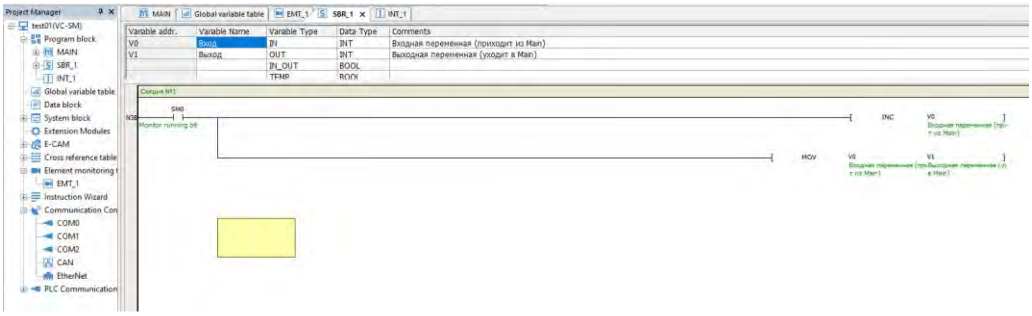


Figure 4-1 The writing process of subroutine SBR_1

Step 3: Write the main program and call the subprograms

In the main program, use the CALL instruction to call the subroutine SBR_1.

The code of the whole main program is as follows:

```
LD    M100
CALL  SBR_1   10 5   D2
```

You can use the parameter transfer correspondence table to fill in the parameters brought or returned when calling the subroutine.

① Bring in the parameter Number1 and pass the Constant integer 10

② The Constant integer 5 is passed in the parameter Number2

③ The return value SumResult is passed to D2

See figure below:

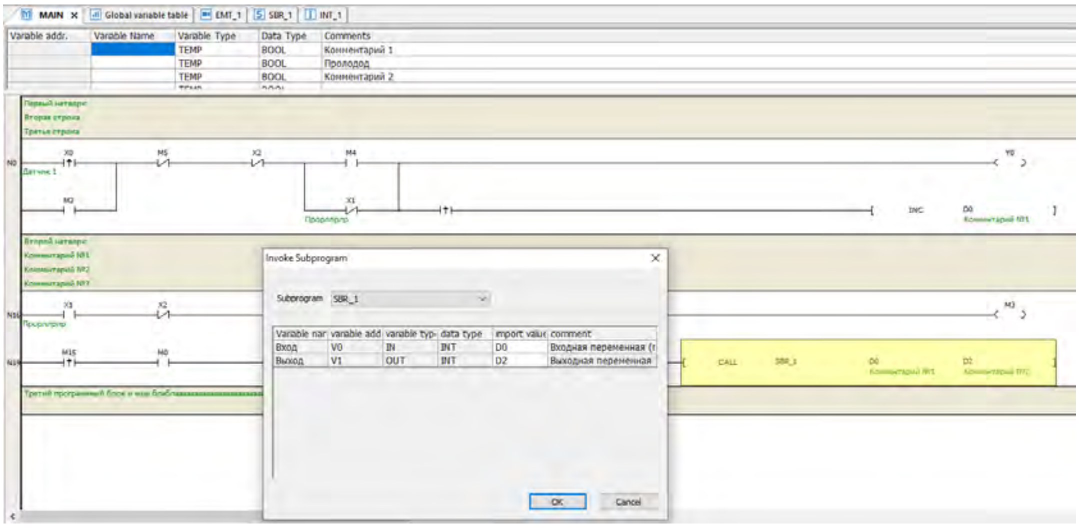


Figure 4-2 call subroutine

Step 4: Compile, download and run user programs to verify the logical correctness of subprograms.

3) Example execution result

When M100=ON , the SBR_1 subroutine is called, and the operand Number1 is brought in. After Number2 is passed the values 10 and 5, the addition operation is completed and the return value is 15, and finally D2=15.

4.5 General Instructions

4.5.1 The operands of the instruction

The operands of instructions can be divided into the following two categories.

① Source operand: The instruction reads its data for operation processing. In the instruction description, it is represented by S, and if there is more than one, it is represented by S1, S2, S3, etc.

② Destination operand: The instruction controls or outputs the Destination operand. In the instruction description, it is represented by D, and if there is more than one, it is represented by D1, D2, etc.

Operands have bit elements, single-word elements or double-word elements, and constants. For details, please refer to the detailed description of the relevant instructions in Chapter 5 and Chapter 6.

4.5.2 Flag bit

Instruction operations may affect three Flag bits.

1) Zero flag SM80

If the instruction operation produces a zero result, the zero flag is set.

2) Carry flag SM81

If the instruction operation has a carry, the Carry flag is set.

3) Borrow flag SM82

If the instruction operation has a borrow, set the borrow flag.

4.5.3 Restrictions on the use of directives

There are some restrictions on the application of some commands, some of which are listed below. For details, please refer to the relevant instructions for details.

1) Exclusive hardware resources

When some instructions are executed, they will occupy hardware resources, and other instructions related to the hardware resources cannot be used at the same time.

For example: high-speed counting command, SPD frequency measurement command, etc. Any such command will occupy some input points of X0~X7. Using these commands at the same time will conflict with each other.

2) Time exclusive

Some instructions execute for a period of time. Therefore, when using these instructions, it is necessary to ensure that the instructions have enough time to complete the function, and only one can be executed at a certain time when the system is running.

For example, due to the time nature of communication, only one command XMT can be sent to the free port at the same time; similarly, the same is true of the command RCV received by the free port. Each time a Modbus command is executed, there is also an exclusive situation for a period of time. The same applies to high-speed output commands and positioning commands.

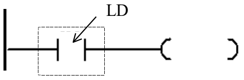
3) Directive application scope restrictions

Some instructions are limited in scope and cannot be used under certain circumstances. For example, the MC/MCR instruction pair cannot be used in the step state of SFC sequential function diagram programming.

Chapter 5 Basic Instructions

5.1 Contact Logic Instruction

5.1.1 LD: Normally open contact command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: LD ***(S)***	**Step size**	**1**

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

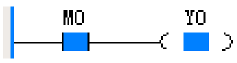
- **Operand Description**

S: Source operand

- **Function Description**

Connect the left bus for making (state ON) or disconnecting (state OFF) power flow.

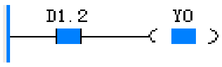
◆ **Example of use**



LD M0
OUT Y0

When M0 is ON, Y0 output is ON.

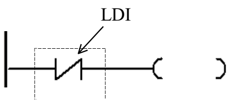
◆ **Example of use**



LD D1.2
OUT Y0

When the second bit of D1 is 1, Y0 output is ON.

5.1.2 LDI: Normally closed contact command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: LDI ***(S)***	**Step size**	**1**

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

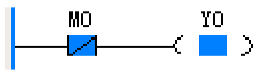
- **Operand Description**

S: Source operand

- **Function Description**

Connect the left bus for making (state OFF) or disconnecting (state ON) power flow.

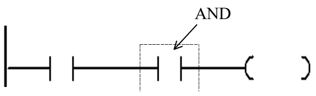
- **Example of use**



LDI M0
OUT Y0

When M0 is OFF, Y0 output is ON.

5.1.3 AND: Normally Open contact and command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:AND ***(S)***	**Step size**	**1**

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

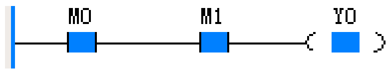
- **Operand Description**

- **Example of use**

S: Source operand

- **Function Description**

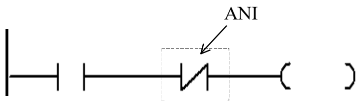
The ON/OFF state of the specified contact (S) and the current energy flow are ANDed, and then assigned to the current energy flow.



LD M0
AND M1
OUT Y0

When M0 is ON and M1 is ON, Y0 output is ON.

5.1.4 ANI: Normally closed contact and command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:ANI *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

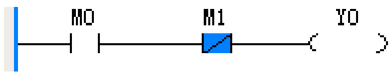
- **Operand Description**

S: Source operand

- **Function Description**

After inverting the ON/OFF state of the specified contact (S), perform an AND operation with the current power flow value, and assign it to the current power flow.

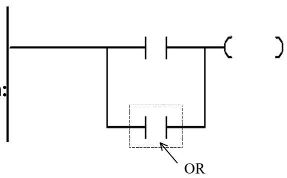
- **Example of use**



LD M0
ANI M1
OUT Y0

When M0 is ON and M1 is OFF, Y0 output is ON.

5.1.5 OR: Normally open contact or command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:OR *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

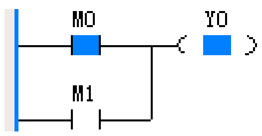
- **Operand Description**

S: Source operand

- **Function Description**

After the ON/OFF state of the specified contact (S) and the current power flow are "OR" operation, it is assigned to the current power flow.

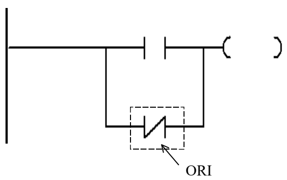
- **Example of use**



LD M0
OR M1
OUT Y0

When M0 or M1 is ON, Y0 output is ON.

5.1.6 ORI: Normally closed contact or command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:ORI *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

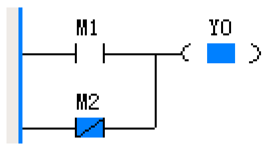
- **Operand Description**

S: Source operand

- **Function Description**

After inverting the ON/OFF state of the designated contact (S) and the current power flow value, perform an "OR" operation and assign it to the current power flow.

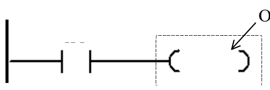
- **Example of use**



LD M1
ORI M2
OUT Y0

When M1 is ON or M2 is OFF, Y0 output is ON.

5.1.7 OUT: Coil output command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:OUT *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

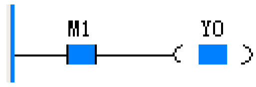
- **Operand Description**

S: Source operand

- **Function Description**

Assigns the current power flow value to the specified coil (D).

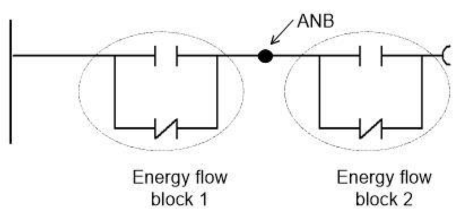
- **Example of use**



LD M1
OUT Y0

When M1 is ON, Y0 output is ON.

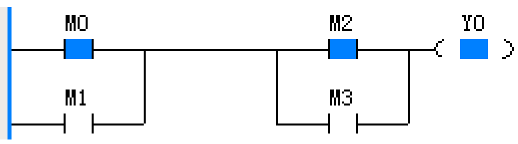
5.1.8 ANB: Power Flow Blocks and Instructions

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: ANB	Step size	1

- **Operand Description**
- **Function Description**

Perform the AND operation on the power flow values of the two power flow blocks and assign them to the current power flow.

- **Example of use**

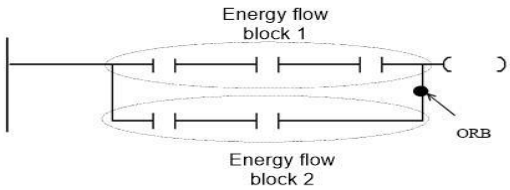


LD M0
OR M1
LD M2
OR M3
ANB
OUT Y0

When one of M0 and M1 is ON, and M2 and M3

When one of them is ON, Y0 output is ON.

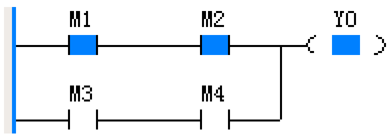
5.1.9 ORB: power flow block or instruction

Ladder Diagram: 	Applicable models	VC-B CV3
	Affect the flag	
Command list: ORB	Step size	1

- **Operand Description**
- **Function Description**

Perform "OR" operation on the power flow values of two power flow blocks, and assign it to the current power flow.

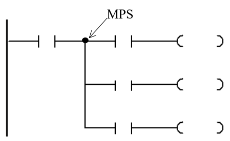
- **Example of use**



LD M1
AND M2
LD M3
AND M4
ORB
OUT Y0

When M1 and M2 are both ON or M3 and M4 are both ON, Y0 output is ON.

5.1.10 MPS: Output can flow into the stack instruction

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: MPS	Step size	1

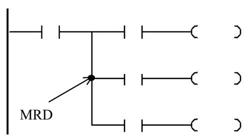
- **Function Description**

The current power flow value is saved on the stack for subsequent power flow calculation of the output branch.

- **Precautions**

In a ladder diagram network, it is forbidden to use MPS more than 8 times in a row (there is no MPP instruction in the middle), otherwise it will cause the overflow of the power flow output stack.

5.1.11 MRD: Read output power flow stack top value instruction

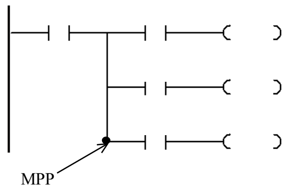
Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	

Command list: MRD	Step size	1

- **Function Description**

Assign the top value of the power flow output stack to the current power flow

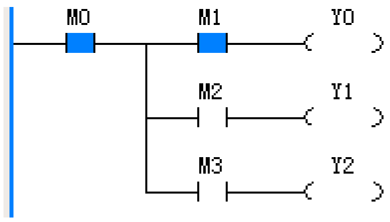
5.1.12 MPP: Output Power flow stack pop instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:MPP	Step size	1

- **Function Description**

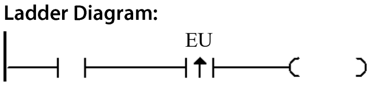
Pop the power flow output stack, and assign the popped value to the current power flow.

- **Example of use**




```
LD   M0
MPS
AND  M1
OUT  Y0
MRD
AND  M2
OUT  Y1
MPP
AND  M3
OUT  Y2
```

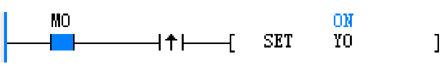
5.1.13 EU: Rising Edge detection command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: EU	Step size	2

- **Function Description**

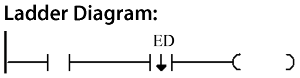
Compare the change of input power flow between this scan and the last scan. When the power flow has a rising edge change (OFF→ON), the output is valid in this scan period.

- **Example of use**




```
LD  M0
EU
SET Y0
```

5.1.14 ED: Falling edge detection command

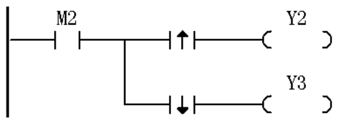
Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	

Command list: ED	Step size	2

- **Function Description**

Compare the change of input power flow between this scan and the last scan. When the power flow has a falling edge change (ON→OFF), the output is valid in this scan period.

- **Example of use**

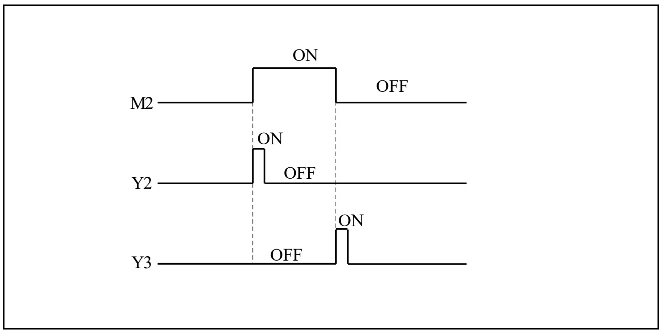


LD M2
MPS
EU
OUT Y2
MPP
ED
OUT Y3

1. In two consecutive scan cycles, the states of the M2 contacts are OFF and ON respectively, and the EU instruction detects the rising edge change, so that Y2 outputs an ON state with a width of one scan cycle.

2. In two consecutive scan cycles, the states of the M2 contacts are ON and OFF respectively, and the ED instruction detects the falling edge change, so that Y3 outputs the ON state of one scan cycle width.

- **Sample Timing Diagram**

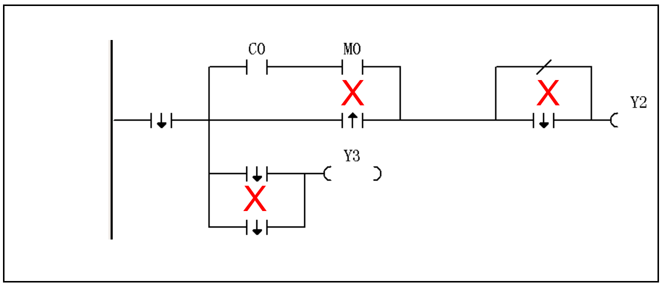


- **Precautions**

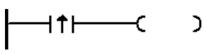
In the ladder diagram, the rising edge contact or falling edge contact command should be used in series with other contact elements, and cannot be used in parallel with other contact elements.

In the ladder diagram, the rising edge contact or falling edge contact command cannot be directly connected to the left power flow bus.

The following are examples of EU/ED instructions incorrectly used in ladder diagrams:



5.1.15 LDP: Contact rising edge power flow load command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: LDP *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM				C	T			

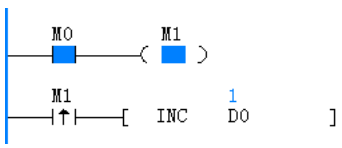
- **Operand Description**

S: Source operand

- **Function Description**

The LDP instruction is used to take the rising edge of the contact signal. If the rising edge transition of the corresponding signal is detected in this scan, the contact is valid, and in the next scan, the contact becomes invalid.

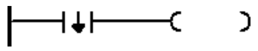
- **Example of use**



LD M0
OUT M1
LDP M1
INC D0

When M0=ON, M1 outputs a high level, at this time, the contact M1 changes from OFF to ON and remains valid for 1 scan cycle, D0 executes a self-increment 1, the next scan cycle, the contact M1 will be invalid, D0 remains 1 not Change.

5.1.16 LDF: Contact Falling Edge Power Flow Load Command

Ladder Diagram: 									**Applicable models**			**VC-B VC-S VC-P**			
									Affect the flag						
Instruction List: LDF *(S)*									**Step size**			**1**			
Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM				C	T			

- **Operand Description**

S: Source operand

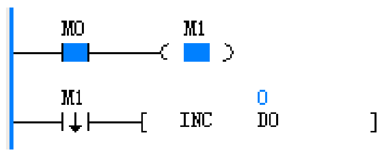
- **Function Description**

The LDF instruction is used to take the falling edge of the contact signal, if this scan

If the falling edge transition of the corresponding signal is detected, the contact is valid,

At the next scan, the contact becomes inactive.

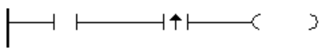
- **Example of use**



LD M0
OUT M1
LDF M1
INC D0

When M0=ON, the output of M1 is ON. At this time, if the contact M1 changes from ON to OFF, M1 will remain valid for one falling edge scan period, D0 will perform a self-increment by 1, and the contact M1 will be in the next scan period. Invalid, D0 remains 1 unchanged.

5.1.17 ANDP: Contact rising edge energy flow and command

Ladder Diagram: 									**Applicable models**			**VC-B VC-S VC-P**			
									Affect the flag						
Command list: ANDP *(S)*									**Step size**			**1**			
Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM				C	T			

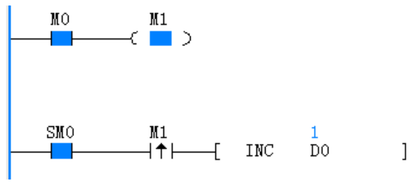
- **Operand Description**

S: Source operand

- **Function Description**

The ANDP instruction is to participate in the AND operation of the rising edge transition state of the contact;

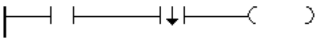
- **Example of use**



LD M0
OUT M1
LD SM0
ANDP M1
INC D0

When M0=ON, M1 outputs a high level. At this time, if the contact M1 changes from OFF to ON, M1 will remain valid for one rising edge scan cycle, D0 will perform a self-increment by 1, and the contact M1 will be in the next scan cycle. Will be invalid, D0 remains 1 unchanged.

5.1.18 ANDF: Contact falling edge energy flow and command

Ladder Diagram: 								Applicable models		VC-B VC-S VC-P					
								Affect the flag							
Command list: ANDF (S)								Step size		1					
Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM				C	T			

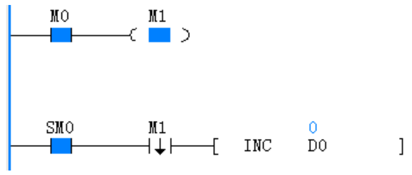
- **Operand Description**

S: Source operand

- **Function Description**

The ANDF instruction is to participate in the AND operation of the falling edge transition state of the contact;

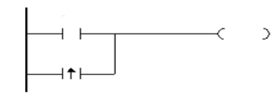
- **Example of use**



LD M0
OUT M1
LD SM0
ANDF M1
INC D0

When M0=ON, the output of M1 is ON. At this time, if the contact M1 changes from ON to OFF, M1 will remain valid for one falling edge scan period, D0 will perform a self-increment by 1, and the contact M1 will be in the next scan period. Invalid, D0 remains 1 unchanged.

5.1.19 ORP: Contact rising edge energy flow or command

Ladder Diagram: 								Applicable models		VC-B VC-S VC-P					
								Affect the flag							
Command list:ORP (S)								Step size		1					
Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM				C	T			

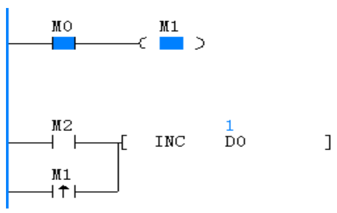
- **Operand Description**

S: Source operand

- **Function Description:**

The ORP instruction involves the rising edge transition state of the contact in the OR operation;

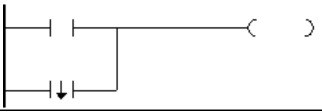
- **Example of use**



LD M0
OUT M1
LD M2
ORP M1
INC D0

When M0=ON, the output of M1 is ON, and when the contact M1 changes from OFF to ON, M1 will remain valid for 1 rising edge scan cycle, D0 will execute a self-increment by 1, and the contact M1 will be invalid in the next scan cycle , D0 remains 1 unchanged.

5.1.20 ORF: Contact falling edge energy flow or command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: ORF *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL	X	Y	M	S	LM	SM				C	T			

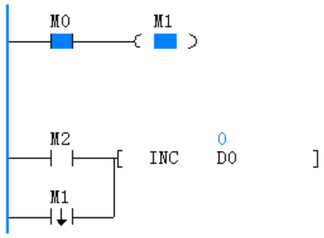
- **Operand Description**

S: Source operand

- **Function Description:**

The ORF instruction is to participate in the OR operation of the falling edge transition state of the contact;

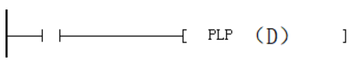
- **Example of use**



LD M0
OUT M1
LD M2
ORF M1
INC D0

When M0=ON, the output of M1 is ON. At this time, if the contact M1 changes from ON to OFF, M1 will remain valid for one falling edge scan period, D0 will perform a self-increment by 1, and the contact M1 will be in the next scan period. Invalid, D0 remains 1 unchanged.

5.1.21 PLP: Rising edge output command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: PLP *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL		Y	M		LM	SM								

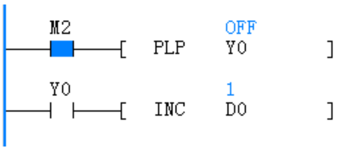
- **Operand Description**

- **Example of use**

S: Source operand

- **Function Description:**

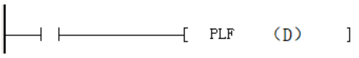
The PLP instruction is to take the rising edge of the coil signal. If the rising transition of the corresponding signal is detected in this scan, the corresponding contact is valid, and in the next scan cycle, the contact corresponding to the coil becomes invalid.



LD M2
PLP Y0
LD Y0
INC D0

When M2=ON, the output of Y0 is ON in this cycle. At this time, for the contact of Y0 to be ON, D0 performs a self-incrementing 1 operation, the coil of Y0 will be invalid in the next scan cycle, and D0 remains 1 unchanged.

5.1.22 PLF: Falling edge output command

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: PLF *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	BOOL		Y	M		LM	SM								

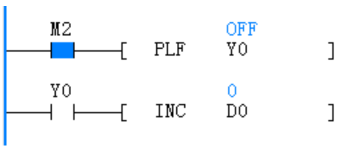
- **Operand Description**

S: Source operand

- **Function Description:**

The PLF instruction is to take the falling edge of the coil signal. If the falling edge of the corresponding signal is detected in this scan, the corresponding contact is valid. In the next scan cycle, the contact corresponding to the coil becomes invalid.

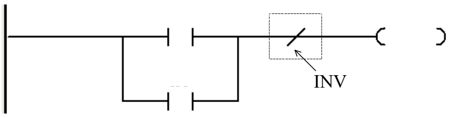
- **Example of use**



LD M2
PLF Y0
LD Y0
INC D0

When M2=ON, the output of Y0 is ON in this cycle. At this time, for the contact of Y0 to be ON, D0 performs a self-incrementing 1 operation, the coil of Y0 will be invalid in the next scan cycle, and D0 remains 1 unchanged.

5.1.23 INV: Energy flow negation instruction

Ladder Diagram: 	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list :INV	Step size	1

- **Function Description**

Invert the current energy flow value, and then assign it to the current energy flow.

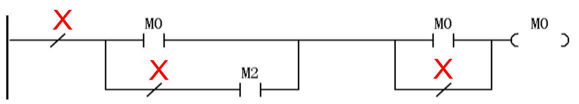
- **Precautions**

In the ladder diagram, the inverse instruction should be used in series with the contact element, and cannot be used in parallel with other contact elements.

INV cannot be used as the first command of the input parallel branch.

In the ladder diagram, the power flow negation instruction cannot be directly connected to the left power flow bus.

The following is an example of an incorrect use of the INV instruction in a ladder diagram:



5.1.24 SET: Coil set command

Ladder Diagram: |—| |—[SET (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: SET (S)	Step size	1

Operand	Type	Applicable devices												Index
S	BOOL		Y	M	S	LM	SM		Dx.y		C	T			

- **Operand Description**

S: Source operand

- **Function Description**

When the power flow is valid, the bit element designated by D will be set.

- **Example of use**

LD M0
SET M1

5.1.25 RST: Coil Clear Command

Ladder Diagram: |—| |—[RST (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: RST (S)	Step size	1

Operand	Type	Applicable devices												Index
S	BOOL		Y	M	S	LM	SM		Dx.y		C	T			

- **Operand Description**

S: Source operand

- **Function Description**

When the power flow is valid, specify the bit element (*D*) will be cleared.

- **Example of use**

LD M0
RST M1

Precautions

If D is component C, the corresponding count value will also be cleared; If D is a T element, the corresponding timing value will also be cleared..

5.1.26 NOP: Null operation instruction

Ladder Diagram: ├─┤ ├─[NOP]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: NOP	Step size	1

- **Function Description**

This command produces no action.

- **Precautions**

In the ladder diagram, this instruction cannot directly connect the left power flow bus.

5.2 Master Command

5.2.1 MC: Master control command

Ladder Diagram: ├─┤ ├─[MC (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list:MC *(S)*	Step size	3

Operand	Type	Applicable devices													Index
S	INT	Constant													

- **Operand Description**

S: Source operand

5.2.2 MCR: Master Clear Command

Ladder Diagram: ├─[MCR (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: MCR *(S)*	Step size	1

Operand	Type	Applicable devices													Index
S	INT	Constant													

- **Operand Description**

S: Source operand

- **Function Description**

1. MC and MCR instructions are matched into a MC-MCR structure. The MC instruction represents the beginning of a MC-MCR structure. Its Operand S is the label of the MC-MCR structure, and its value is a Constant between 0 and 7. MCR represents the end of a MC-MCR structure.

2. When the energy flow before MC instruction is valid, execute the instruction in the middle of MC-MCR structure.

3. When the energy flow before the MC instruction is invalid, the instruction in the middle of the MC-MCR structure is skipped and not executed. The program directly jumps to the structure and executes. The Destination operand corresponding to OUT, TON, TOF, PWM, HCNT, PLSY, PLSR, DHSCS, SPD, DHSCI, DHSCR, DHSZ, DHST, DHSP, and BOUT in the structure will be cleared.

- **Example of use**

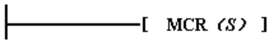
- **Precautions**

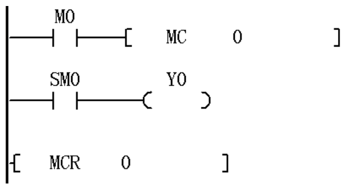
1. In the ladder diagram, the MCR instruction must be directly connected to the left power flow bus.

2. In the ladder diagram, the MCR instruction cannot be connected in parallel or in series with other instructions.

3.Multiple MC-MCR structures with different numbers can be used in mosaic, but the number of mosaic layers cannot exceed 8. The MC-MCR structure with the same number is forbidden to use mosaic.

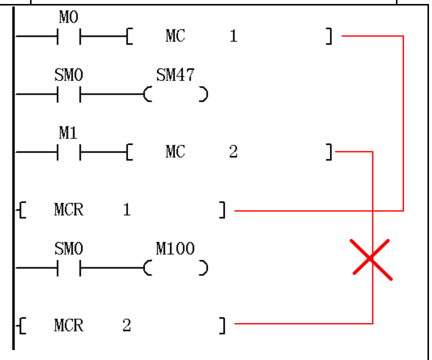
4. Two MC-MCR structures cannot be used interchangeably. The following method is illegal:

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	



LD M0
MC 0
LD SM0
OUT Y0
MCR 0

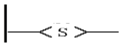
When M0=ON, the instructions in the MC 0 - MCR 0 structure will be executed, and Y0=ON. When M0=OFF, the instructions in the MC 0 - MCR 0 structure will not be executed, and the bit element Y0 specified by the destination Operand of the OUT instruction in the structure will be cleared. Y0=OFF.



Note: Not available in SFC programming.

5.3 SFC Instruction

5.3.1 STL: SFC state load instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: STL (S)	Step size	3

Operand	Type	Applicable devices												Index
S	BOOL						S							

- **Operand Description**

S: Source operand

- **Function Description**

1. Represents the start of a step state(s) process.

2. If the step status is valid (ON), its built-in instruction will be executed.

3. If the step state is changed from valid to invalid (falling edge change), its built-in instruction sequence will not be executed, and the built-in OUT, TON, TOF, DHSCR, DHSZ, DHST, DHSP, BOUT The corresponding Destination operand will be cleared.

4. If the step state is invalid, its built-in instruction sequence will not be executed.

5. Consecutive STL instructions (concatenation of STL elements) represent defining a parallel merge structure, and the maximum number of consecutive STL instructions is 16 times (the maximum number of branches of a parallel branch merge structure is 16).

5.3.2 SET Sxx: SFC state transition

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
├─< S >──┤ ├──[SET (D)]	**Affect the flag**	
Command list: **SET** *(D)*	**Step size**	**3**

Operand	Type	Applicable devices													Index
D	BOOL						S								

- **Operand Description**

D: Destination operand

- **Function Description**

When the power flow is active, the step state (*D*) is set to be valid, and at the same time, the currently valid stepping state is set to be invalid, and the action of stepping state transition is completed.

5.3.3 OUT Sxx: SFC state jump

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
├─< S >──┤ ├──( )	**Affect the flag**	
Command list:**OUT** *(D)*	**Step size**	**3**

Operand	Type	Applicable devices													Index
D	BOOL						S								

- **Operand Description**

D: Destination operand

- **Function Description**

When the power flow is active, the step state (*D*) is set to be valid, and at the same time, the currently valid stepping state is set to be invalid, and the jumping action of the stepping state is completed.

5.3.4 RST Sxx: SFC status clear

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
├─< >──┤ ├──[RST (D)]	**Affect the flag**	
Command list: **RST** *(D)*	**Step size**	**3**

Operand	Type	Applicable devices													Index
D	BOOL						S								

- **Operand Description**

D: Destination operand

- **Function Description**

with the specified step state when power flow is active (*D*) is invalidated.

5.3.5 RET: SFC block end

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
├[RET]	**Affect the flag**	
Command list: RET	**Step size**	**1**

- **Instruction function description**

Marks the end of a SFC program.

- **Precautions**

Can only be used in the main program.

5.4 Timer Command

5.4.1 TON: On-delay timing command

Ladder Diagram: ├─┤ ├──[TON (D) (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: TON (D) (S)	Step size	5

Operand	Type	Applicable devices														Index
D	INT											T				
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

S: Source operand

- **Function Description**

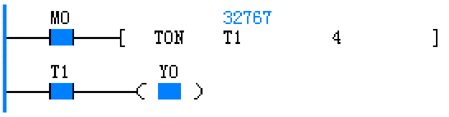
1. When the power flow is valid and the timing value is less than 32,767, the specified T element (*D*)Timing (the timing value is accumulated as time goes by). When the timer value reaches 32,767, the timer value will remain unchanged at 32,767.

2. When the timing value ≥ the preset value (*S*), the timing coil output of the specified T element turns ON.

3. When the power flow is OFF, the timing is stopped, the timing value is cleared to zero, and the output of the timing coil is OFF.

4. When the system executes this command for the first time, the timing coil value of the specified T element will be cleared to OFF, and the timing value will be cleared.

- **Example of use**



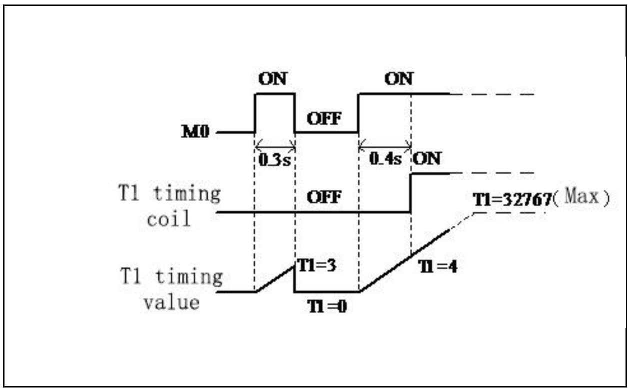
LD M0

TON T1 4

LD T1

OUT Y0

- **Sample Timing Diagram**



5.4.2 TONR: Memory type on-delay timing command

Ladder Diagram: ├─┤ ├──[TONR (D) (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: TONR (D) (S)	Step size	5

Operand	Type	Applicable devices														Index
D	INT											T				
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

S: Source operand

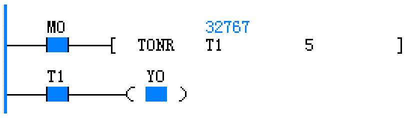
- **Function Description**

1. When the power flow is valid and the timing value is less than 32,767, the specified T element (***D***)Timing, the timing value increases with the travel time. When the timer value reaches 32,767, the timer value will remain unchanged at 32,767.

2. When the timing value ≥ the preset value (***S***), the timing coil output of the specified T element turns ON.

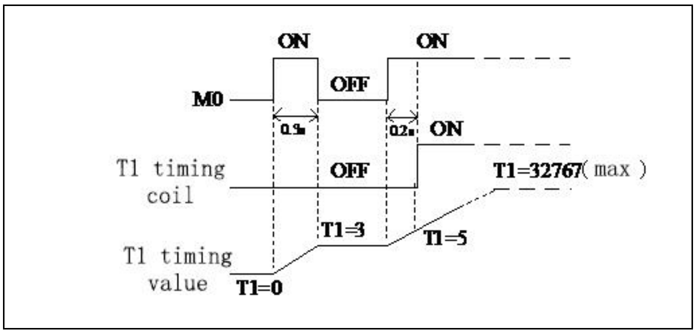
3. When the power flow is OFF, the timing is stopped, and the timing coil and timing value keep the current timing value unchanged.

- **Example of use**



LD M0
TONR T1 5
LD T1
OUT Y0

- **Sample Timing Diagram**



5.4.3 TOF: Off Delay Timer Command

Ladder Diagram: ├─┤ ├─[TOF *(D)* *(S)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: TOF (D) (S)	Step size	5

Operand	Type	Applicable devices													Index
D	INT											T				
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

S: Source operand

- **Function Description**

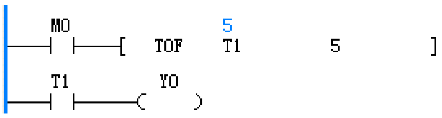
1.When the power flow changes from ON→OFF (falling edge), specify the timer T (*D*) to start the timer.

2. When the power flow is OFF, if the specified timer T has started to count, it will continue to count. until the timer value equals the preset value (*S*), the output of the timing coil of the specified T element is OFF, after that the timing value will remain the preset value and will not change.

3. If the timer is not started, even if the power flow input is OFF, the timer will not be counted.

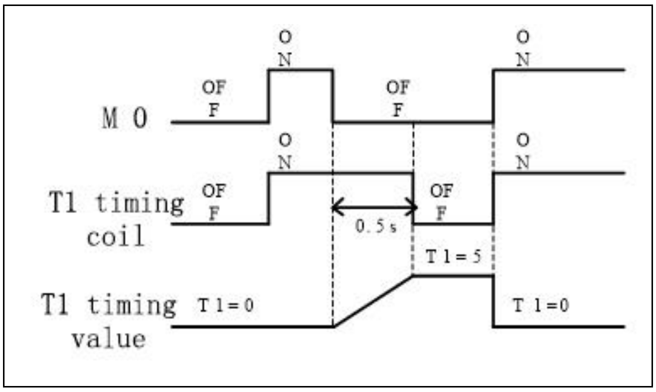
4. When the power flow is ON, the timing is stopped, the timing value is cleared to zero, and the timing coil output is ON.

- **Example of use**



LD M0
TOF T1 5
LD T1
OUT Y0

- **Sample Timing Diagram**



5.4.4 TMON: Do not retrigger the monostable timing command

Ladder Diagram: ├─┤ ├─[TMON *(D)* *(S)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: TMON (D) (S)	Step size	5

Operand	Type	Applicable devices	Index

D	INT											T				
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

● **Operand Description**

D: Destination operand

S: Source operand

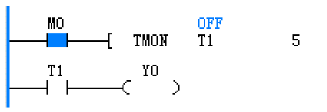
● **Function Description**

1. When the input power flow changes from OFF→ON (rising edge), and it is in the untimed state, start the specified timer T (*D*) timing (starting from the current value), in the timing state (the length of the timing state is determined by S), keep the timing coil output ON.

2. In the timing state (the timing length is determined by *S*OK), no matter how the power flow changes, keep timing, and the timing coil output remains ON.

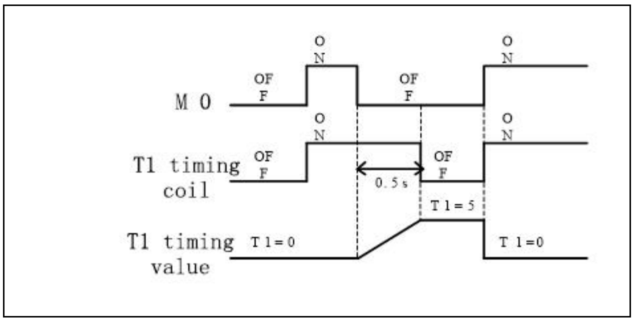
3. When the timing value arrives, the timing is stopped, the timing value is cleared to zero, and the coil output is cleared to OFF.

● **Example of use**




```
LD    M0
TMON  T1 5
LD    T1
OUT   Y0
```

● **Sample Timing Diagram**



5.5 Counter Instruction

5.5.1 CTU: 16-bit up counter instruction

Ladder Diagram: ├─┤ ├─[CTU (D) (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: CTU (D) (S)	Step size	5

Operand	Type	Applicable devices														Index
D	INT											C				
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

● **Operand Description**

D: Destination operand

S: Source operand

● **Function Description**

1. When the power flow changes from OFF→ON (rising edge), the count value of the specified 16-bit counter C (D) increases by one.

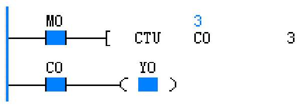
2. When the count value reaches 32,767, the count value remains unchanged.

3. When the count value is greater than or equal to the count preset value (S), the count coil is set to ON.

● **Precautions**

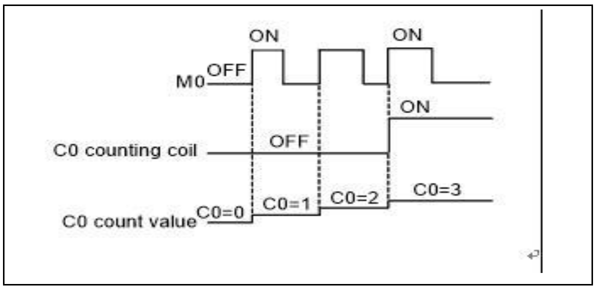
(D) The address of the designated 16-bit counter C should be within C0 to C199.

● **Example of use**




```
LD    M0
CTU   C0 3
LD    C0
OUT   Y0
```

● **Sample Timing Diagram**



5.5.2 CTR: 16-bit loop count instruction

Ladder Diagram: ├─┤ ├─[CTR (D) (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	

Command list: **CTR**(D) (S)									**Step size**		**5**					
Operand	Type	Applicable devices														Index
D	INT											C				
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

S: Source operand

- **Function Description**

1. When the input power flow changes from OFF→ON (rising edge), the count value of the designated 16-bit counter C (D) increases by 1.

2. When the count value is equal to the count preset value (S), the count coil is set to ON.

3. When the count value is equal to the count preset value (S), if the input power flow changes from OFF to ON again (rising edge), the count value is set to 1, and the count coil is cleared to OFF.

Precautions

1. When the count preset value (S) is less than or equal to zero, no count action occurs.

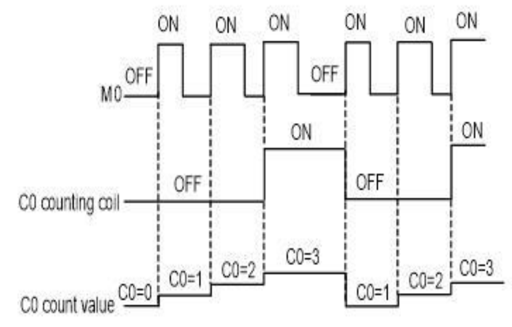
2. (D) The address of the designated 16-bit counter C should be within C0 to C199.

- **Example of use**

LD M0

CTR C0 3

- **Sample Timing Diagram**



5.5.3 DCNT: 32-bit increment and decrement count instructions

Ladder Diagram: ├─┤ ├─[DCNT *(D)* *(S)*]									**Applicable models**		**VC-B VC-S VC-P**					
									Affect the flag							
Command list: **DCNT**(D) (S)									**Step size**		**7**					
Operand	Type	Applicable devices														Index
D	DINT											C				
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

S: Source operand

- **Function Description**

1. When the input power flow changes from OFF→ON (rising edge), the count value of the designated 32-bit counter C (D) increases or decreases by 1 (the direction of count increase and decrease is determined by the corresponding SM Flag bit).

2. When it is an up-counter, when the count value is greater than or equal to the count preset value (S), the count coil is set to ON.

3. When it is a down counter, when the count value is less than or equal to the count preset value (S), the count coil is set to OFF.

4. When the count value = 2147483647, if the count is incremented again, the count value becomes -2147483648.

5. When the count value=-2147483648, if the count is decremented by one again, the count value becomes 2147483647.

- **Precautions**

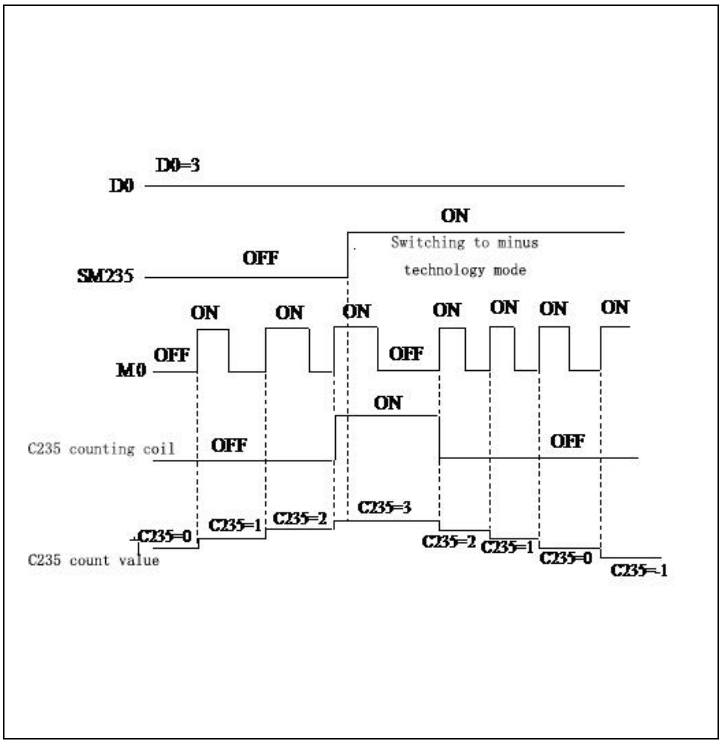
D The address of the specified C element should be between C200 and C235.

- **Example of use**

LD M0

DCNT C235 D0

- **Sample Timing Diagram**



Chapter 6 Application Instruction

6.1 Program Flow Control Instructions

6.1.1 FOR: Loop instructions

Ladder Diagram: ├─┤ ├─[FOR (S)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	
Command list: **FOR**(S)															Step size	3
Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

Operand description

S: Source operand

6.1.2 NEXT: Loop back

Ladder Diagram: ├[NEXT]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: **NEXT**	Step size	1

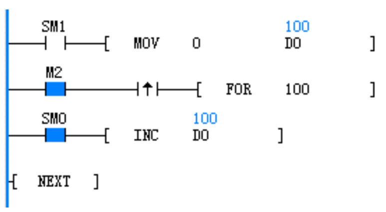
- **Function Description**

1. The FOR instruction matches NEXT into a FOR-NEXT structure.

2. When the current power flow of FOR is valid, and the number of loops (***S***) is greater than zero, the instructions in the middle of the FOR-NEXT structure are executed in a continuous loop ***S*** Second-rate. When the loop is finished ***S*** After that, continue to execute the instruction after the FOR-NEXT structure.

3. If the power flow before FOR is invalid, or the loop count (***S***) is less than or equal to zero, the instruction in the middle of the FOR-NEXT structure is not executed, and the program directly jumps to the FOR-NEXT structure and continues to execute.

- **Example of use**



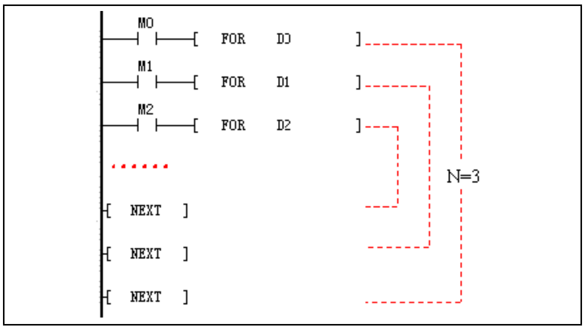

```
LD    SM1
MOV   0    D0
LD    M2
EU
FOR   100
LD    SM0
INC   D10
NEXT
```

The initial conditions of operation D0=0, M2=OFF. When M2 changes from OFF→ON, the instruction in the FOR-NEXT structure is continuously executed 100 times, and D0 is incremented 100 times. After the cycle ends, D0=100.

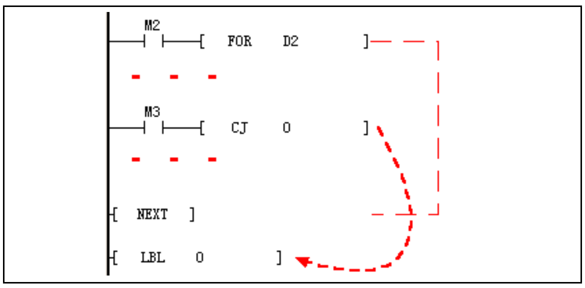
- **Precautions**

1. The FOR-NEXT instruction must be used in pairs in a program body (POU), otherwise the user program cannot be compiled correctly.

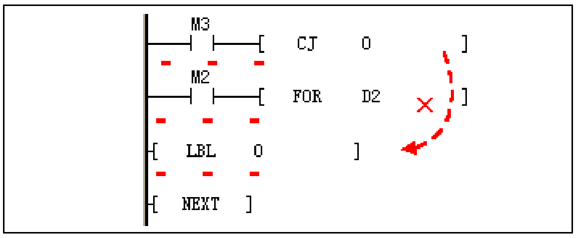
2. Multiple FOR-NEXT structure nesting is supported. VC2L series CPU units only support up to 8 layers of FOR-NEXT structure nesting. (The figure below illustrates a 3-level FOR-NEXT structure nesting)



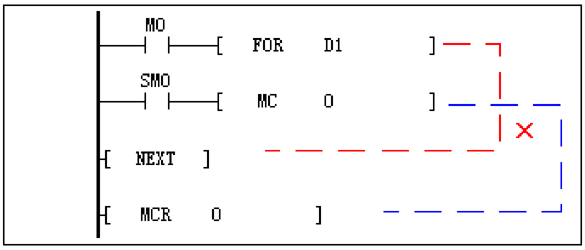
3. The conditional jump instruction (CJ) can be used in the loop body to jump out of the loop body, so as to achieve the purpose of terminating the execution of the loop body in advance, as shown in the following ladder diagram:



4. Users are prohibited from jumping into a loop body using jump statement (CJ), the following ladder diagram will not compile correctly:



5. The intersection of MC-MCR structure and FOR-NEXT structure is prohibited, the following ladder diagram will not compile correctly:



📖 Notice

The execution of the FOR-NEXT loop body is time-consuming. The more the number of loops, or the more instructions contained in the loop body, the longer the execution time will be. To prevent run time timeout errors, be careful to use WDT instructions within the body of a time-consuming loop.

6.1.3 LBL: Jump label definition instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┤[LBL (*S*)]	Affect the flag	
Command list: **LBL**(*S*)	**Step size**	**3**

Operand	Type	Applicable devices												Index
S	INT	Constant												

- **Operand Description**

S: Label value

- **Function Description**

1. Defines a label with a label value of S.

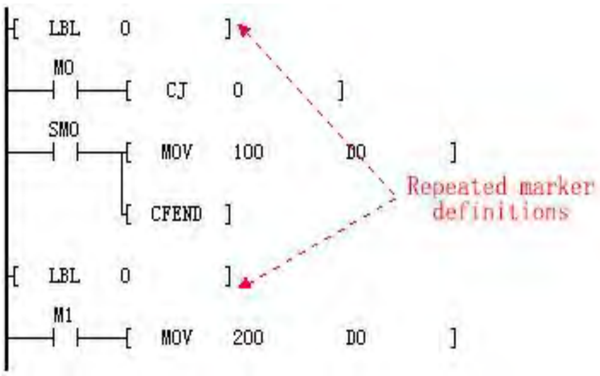
2. No substantive operation is generated, but the specific location of the jump is marked for the conditional jump instruction (CJ).

- **Precautions**

1. The range of the label value S: 0≤S≤127.

2. In a user program, it is not allowed to have two repeatedly defined labels in the same program body, otherwise the user program will fail to compile. However, repeated label definitions are allowed in different program bodies (such as different subroutines).

- **wrong program example**



6.1.4 CJ: Conditional jump instruction

Ladder Diagram: ⊢⊣ ⊢[CJ (S)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: CJ(S)	Step size	3

Operand	Type	Applicable devices													Index
S	INT	Constant													

- **Operand Description**

S: Label value

- **Function Description**

1. When the power flow is valid, the user program jumps to the number of **S** is executed at the legal label instruction.

2. If the power flow is invalid, no jump operation will occur, and the sequence will be executed CJ the last instruction.

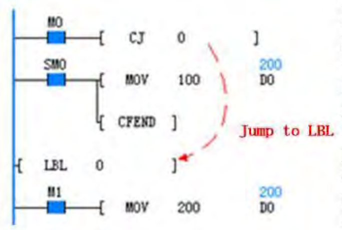
- **Precautions**

1. The label S (0≤S≤127) to be jumped by the CJ instruction should be a legal and defined label, otherwise the user program will not be able to compile correctly.

2. Jumping into a FOR-NEXT structure using the CJ instruction is not allowed.

3. You can use the CJ instruction to jump out of or into the MC-MCR structure and SFC state, but this will destroy the logic of the MC-MCR and SFC state and complicate the program. It is not recommended to use it in this way.

- **Example of use**




```
LD   M0
CJ   0
LD   SM0
MOV 100  D0
CFEND
LBL  0
LD   M1
MOV 200  D0
```

1. Initial condition M0=OFF, M1=ON, CJ 0 does not jump, D0=100. After executing CFEND, the program flow exits the main program in advance, and the instructions LD M1 and MOV 200 D0 are not executed.

2. When M0=ON, M1=ON, the instruction CJ 0 is executed, and the MOV 100 D0 and CFEND instructions are crossed. After jumping to LBL 0, execute MOV 200 D0 instruction, at this time D0=200.

6.1.5 CFEND: User main program conditional return

Ladder Diagram: ⊢⊣ ⊢[CFEND]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: CFEND	Step size	1

- **Function Description**

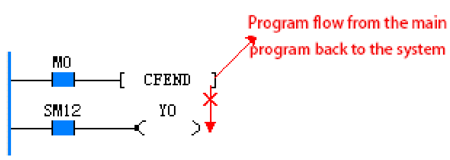
1. When the power flow of the instruction is valid, the main program returns to the system from the current scan cycle (the main program of the user program is called and executed repeatedly by the system according to the scan cycle), and the subsequent instructions in the main program are not executed.

2. When the power flow of an instruction is invalid, the instruction does not produce any action, and the subsequent instructions are executed sequentially.

- **Precautions**

The CFEND instruction must appear in the user's main program, otherwise it cannot be compiled.

- **Example of use**




```
LD   M0
CFEND
LD   SM12
OUT  Y0
```

When the program is running, when M0=OFF, the CFEND instruction does not act, and the subsequent LD SM12, OUT Y0 is executed, and Y0 periodic flickering output can be observed. When M0=ON, the CFEND instruction will act, and the program flow will return to the system from the main program in advance. After that, LD SM12 and OUT Y0 will not be executed, and the periodic flickering phenomenon of Y0 will disappear.

6.1.6 WDT: User program watchdog clear

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[WDT]	Affect the flag	
Command list: WDT	Step size	1

- **Function Description**

When the power flow is valid, this instruction will reset the timer value of the user program watchdog to zero, and the system user program watchdog will start timing again.

6.1.7 EI: Interrupt Enable

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[EI]	Affect the flag	
Command list: EI	Step size	1

- **Function Description**

1. When the power flow is valid, the interrupt is enabled.
2. When the EI instruction is valid, the interrupt request will be allowed to join the interrupt request queue, waiting for the system to respond.

6.1.8 DI: Interrupt Disable

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[DI]	Affect the flag	
Command list: DI	Step size	1

- **Function Description**

1. When the power flow is valid, the interrupt global enable Sign is disabled, that is, the global interrupt is turned off.
2. When the global interrupt enable Sign is disabled, various interrupt events cannot generate interrupt requests.

- **Precautions**

When the close interrupt request instruction takes effect, if there are still interrupt requests in the interrupt request queue that have not been processed, the remaining interrupt requests will still be responded, but new interrupt events will not generate interrupt requests.

6.1.9 CIRET: User Interrupt Program Conditional Return

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[CIRET]	Affect the flag	
Command list: CIRET	Step size	1

- **Function Description**

When the power flow is valid, the interrupt program being executed is exited in advance.

6.1.10 STOP: User program stops

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[STOP]	Affect the flag	
Command list: STOP	Step size	1

- **Function Description**

When the power flow is valid, the system will immediately stop the execution of the user program.

6.1.11 CALL: User subroutine call

Ladder Diagram:		
Ladder Diagram: CALL (SBR_NAME) (PARAME 1) (PARAME 2) (. . .)	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: **CALL** ***(subprogram name) (subprogram parameter 1) (subprogram parameter 2)***…	**Step size**	**Determined by the parameters of the subroutine**

- **Function Description**

When the power flow is valid, the subroutine with the specified name is called for execution. After the subroutine is executed, it returns to the instruction after the CALL to continue execution.

- **Precautions**

1. The subroutine called in the CALL instruction must be defined in the user program in advance. When an undefined subroutine appears in CALL, it cannot be compiled by the program.

2. The element type of the operand carried in the CALL instruction should match the data type defined in the local variable table of the subprogram, otherwise it cannot be compiled.

- **The following example illustrates an illegal match use:**

Example 1: In the local variable table of the SBR1 subroutine, the data type of operand 1 is DINT.

The following uses are illegal:

(1) CALL SBR1 Z0 (Z element cannot be used as data type DINT)

(2) CALL SBR1 C199 (C0~C199 elements cannot be used as data type DINT)

(3) CALL SBR1 K2X0 (Kn addressing 1≤n≤3, cannot be used as data type DINT)

Example 2: In the local variable table of the SBR1 subroutine, the data type of operand 1 is INT.

The following uses are illegal:

(1) CALL SBR1 C200 (C200~C255 elements cannot be used as data type INT)

(2) CALL SBR1 K2X0 (Kn addressing 4≤n≤8, cannot be used as data type INT)

3. The element type of the operand carried in the CALL instruction should match the variable type defined in the local variable table of the subprogram, otherwise it cannot be compiled.

The following example illustrates an illegal match use:

Example: In the local variable table of the SBR1 subroutine, the operand type of operand 1 is OUT or IN_OUT.

The following uses are illegal:

(1) CALL SBR1 321 (the Constant cannot be changed, so it does not match the OUT or IN_OUT type operand)

(2) CALL SBR1 K4X0 (K4X0 is only read-only, so it does not match the operand of OUT or IN_OUT type)

(3) CALL SBR1 SD0 (SD0 is only read-only, so it does not match the OUT or IN_OUT type operand)

4. The number of operands carried in the CALL instruction should match the local variable table of the subprogram, otherwise it cannot be compiled.

6.1.12 CSRET: User Subroutine Conditional Return

Ladder Diagram:		
Ladder Diagram: CSRET	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: CSRET	**Step size**	**1**

- **Function Description**

When the energy flow is valid, exit the currently executed subprogram and return to the previous subprogram.

6.2 Data Transfer Instructions

6.2.1 MOV: Word data transfer instruction

Ladder Diagram: ├─┤ ├─[MOV (S) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	
Command list:MOV (S) (D)															Step size	5
Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, the content of S is assigned to D, and the value of S remains unchanged.

- **Precautions**

1. The MOV instruction supports both signed and unsigned integers. If both operands of the instruction are devices, the data types are signed integers. If the Source operand of the instruction is a signed long integer, such as (-10, +100), the Destination operand is also a signed integer. If the Source operand is an unsigned long integer, such as (100, 45535), the Destination operand is also an unsigned integer.

2. The corresponding device C only supports C0 to C199.

- **Example of use**


```
LD   X0
MOV  D0  D10
```

When X0=ON, the content of D0 is assigned to D10, D10=500.

6.2.2 DMOV: Double word data transfer instruction

Ladder Diagram: ├─┤ ├─[DMOV (S) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	
Command list: DMOV (S) (D)															Step size	7
Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V	Z	R	√
D	DINT			KnY	KnM	KnS	KnLM		D	SD	C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

number, such as (100, 45535), the Destination operand is also an unsigned integer.

2. The corresponding soft element C only supports 32-bit C elements.

● **Function Description**

1. The DMOV instruction supports both signed and unsigned long integers. If both operands of the instruction are devices, the data types are signed integers. If the Source operand of the instruction is a signed long integer, such as (-10, +100), the Destination operand is also a signed integer. if the Source operand is an unsigned long integer

● **Example of use**

LD X0
DMOV D0 D10

When X0=ON, the content of (D0, D1) is assigned to (D10, D11), (D10, D11) = 50000.

6.2.3 RMOV: Floating point data transfer instruction

Ladder Diagram: ├──┤ ├──[RMOV (S) (D)]		**Applicable models**	**VC-B VC-S VC-P**
		Affect the flag	
Command list: **RMOV (S) (D)**		**Step size**	**7**

Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

● **Operand Description**

S: Source operand

D: Destination operand

● **Function Description**

When the power flow is valid, the content of S is assigned to D, and the value of S remains unchanged.

● **Example of use**

RMOV D0 D10

When X0=ON, the content of (D0, D1) is assigned to (D10, D11), (D10, D11) = 50000.5.

6.2.4 BMOV: Block data transfer instruction

Ladder Diagram: ├──┤ ├──[BMOV (S1) (D) (S2)]		**Applicable models**	**VC-B VC-S VC-P**
		Affect the flag	
Command list: **BMOV (S1) (D) (S2)**		**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	INT		KnX	KnY	KnM	KnS	KnLM		D	SD	C	T	V		R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

● **Operand Description**

S: Source operand, start unit of data block

D: Destination operand, starting unit of data block

S2: data block size

● **Function Description**

When the power flow is valid, the contents of the S2 units starting from the S1 unit are assigned to the S2 units starting from the D unit, and the contents of the S2 units starting from the S1 unit remain unchanged.

● **Example of use**

LD X0
BMOV D0 D100 10

When X0=ON, the contents of the 10 units starting from D0 are assigned to the 10 units starting from D100. D100=D0, D101=D1, ..., D109=D9.

6.2.5 FMOV: Data block fill instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┤├ ┤├[FMOV (S1) (D) (S2)]	Affect the flag	
Command list: FMOV (S1) (D) (S2)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand, start unit of data block

D: Destination operand, starting unit of data block

S2: data block size

- **Function Description**

When the power flow is valid, the content of the S1 unit is filled into the S2 units starting from the D unit, and the content of the S1 unit remains unchanged.

- **Precautions**

1. When S1, D, and S2 use C components, the legal range is C0 to C199.
2. S2 is greater than 0.
3. When S1 and D address Kn at the same time, Kn should be equal.

- **Example of use**

LD X0

FMOV D0 D100 10

When X0=ON, the content of D0 is filled to 10 cells starting from D100. D100=D101=……=D109=D0=500.

6.2.6 DFMOV: Data Block Double Word Fill Instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┤├ ┤├[DFMOV (S1) (D) (S2)]	Affect the flag	
Command list: DFMOV (S1) (D) (S2)	Step size	9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: start of Source operand

D: Destination operand, starting unit of data block

S2: data block size

- **Function Description**

When the power flow is valid, the content of the S1 unit is filled into the S2 units starting from the D unit, and the content of the S1 unit remains

- **Precautions**

1. When S1, D, and S2 use C components, only 32-bit C components are supported.
2. S2 is greater than 0.
3. When S1 and D address Kn at the same time, Kn should be equal.

- **Example of use**

LD X0

DFMOV D0 D10 10

unchanged.

When X0=ON, the content of (D0, D1) is filled to 10×2 cells starting from D10. (D10, D11) = (D12, D13) =…= (D28, D29) = (D0, D1) = 100000.

6.2.7 SWAP: High and low byte swap instruction

Ladder Diagram: ├─┤ ├─[SWAP (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: SWAP (D)	Step size	3

Operand	Type	Applicable devices													Index
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

D: Destination operand, refers to the word element whose high and low bytes are exchanged

- **Function Description**

When the power flow is valid, the value of the content of D after the high and low bytes are swapped is stored in the D unit.

- **Example of use**

LD X0
SWAP D0

When X0=ON, the high and low bytes of D0=0x1027 (4135) are exchanged and the value is saved to D0, D0=0x2710 (10000).

6.2.8 XCH: Word exchange instruction

Ladder Diagram: ├─┤ ├─[XCH (D1) (D2)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: XCH (D1) (D2)	Step size	5

Operand	Type	Applicable devices													Index
D1	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
D2	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

D1: Destination operand 1

D2: Destination operand 2

- **Function Description**

When the power flow is valid, the content of D1 and the content of D2 are exchanged and stored in the units D1 and D2.

- **Precautions**

When using Kn addressing mode, Kn in D1 and D2 should be the same.

- **Example of use**

LD X0
XCH D0 D10

When X0=ON, the contents of D0 and D10 are exchanged. Before execution: D0=5000, D10=1000.

After execution: D0=1000, D10=5000.

6.2.9 DXCH: Double word exchange instruction

Ladder Diagram: DXCH (D1) (D2)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: DXCH (D1) (D2)	Step size	7

Operand	Type	Applicable devices														Index
D1	DINT			KnY	KnM	KnS	KnLM		D		C	T	V		R	√
D2	DINT			KnY	KnM	KnS	KnLM		D		C	T	V		R	√

- **Operand Description**

D1: Destination operand 1;

D2: Destination operand 2

- **Function Description**

When the power flow is valid, the content of D1 and the content of D2 are exchanged and stored in the units D1 and D2.

- **Precautions**

When using Kn addressing mode, D1, D2The Kn in should be the same.

- **Example of use**

LD X0

DXCH D0 D10

When X0=ON, the contents of (D0, D1) and (D10, D11) are interchanged. Before execution: (D0, D1) = 5000000, (D10, D11) = 1000000. After execution: (D0, D1) = 1000000, (D10, D11) = 5000000.

6.2.10 PUSH: Data push instruction

Ladder Diagram: PUSH (S1) (D) (S2)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: PUSH (S1) (D) (S2)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: push value

D: The number of elements in the storage stack, and its component label is the bottom position of the stack

- **Precautions**

1. When the operation stack definition is illegal, (when the stack size is less than or equal to zero, the number of elements in the stack is less than zero. The number of elements in the stack is greater than the limit of the stack size), an illegal operation stack definition error is reported.

2. The size of the stack does not include the element at the bottom of the stack (the element specified by D).

S2: the size of the stack

- **Function Description**

1. When the power flow is valid, the value of S1 is pushed into the top of the stack with the D unit as the bottom of the stack, and the value of D is increased by 1 at the same time. At this time, the number of the top unit of the stack is: the number of D + the value of D.

2. When the D value is equal to the S2 value, there is still a push instruction to execute, the operation Carry flag (SM81) is set to 1, and the push operation is not performed.

3. S2 specifies the size of the stack, the range is greater than 0.

- **Example of use**

LD M0
PUSH D0 D100 10

1. When M0=ON, push the contents of D0 into the stack with D100 as the bottom of the stack.
2. Before execution: D0=1000, D100=8, D109=0.
3. After execution: D0=1000, D100=9, D109=1000.

6.2.11 FIFO: First in first out instruction

Ladder Diagram: FIFO (D1) (D2) (S)															
Applicable models	VC-B VC-S VC-P														
Affect the flag															
Command list: FIFO (D1) (D2) (S)															
Step size	7														
Operand	Type	Applicable devices													Index
D1	INT								D				V		R	√
D2	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D1: The number of elements in the stack, and its element number at the same time+1The element is the first element of the stack

D2:pop value storage unit

S: queue size

- **Function Description**

1. When the power flow is valid, the first value of the word stack with D1 as the head of the team (the content of the next unit after D1) is assigned to the D2 unit, and the value of D1 is reduced by 1, and the contents of the S units after D1 are moved from the back to the front. , the last cell is filled with 0.

2. When the value of D1 is equal to 0, the stack is emptied, and the Zero flag (SM80) is set to 1.

- **Precautions**

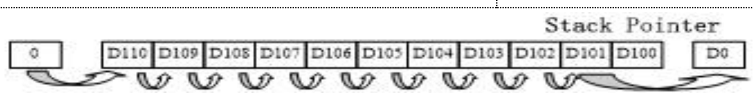
1. When the operation stack definition is illegal, (when the stack size is less than or equal to zero, the number of elements in the stack is less than zero. The number of elements in the stack is greater than the limit of the stack size), an illegal operation stack definition error is reported.

2. The size of the stack does not include the element at the bottom of the stack (the element specified by D).

3. S2 specifies the size of the stack, the range is greater than 0.

- **Example of use**

LD M0
FIFO D100 D0 10



1. When M0=ON, the content of D101 will be filled in D0, and the content of D101~D110 will move from the back to the front, and the content of D110 will be filled with 0.

2. Before execution: D0=0, D100=10, D101=1000, D102=2000, ..., D109=9000, D110=10000.

is illegal, (when the stack size is less than or equal to zero, the number of elements in the stack is less than zero. The number of elements in the stack is greater than the limit of the stack size), an illegal operation stack definition error is reported.

2. The size of the stack does not include the bottom element of the stack (D1specified element)

3. ***S*** specifies the size of the stack, the range is greater than 0.

3. After execution: D0=1000, D100=9, D101=2000, D102=3000, ..., D109=10000, D110=0.

6.2.12 LIFO: Last in first out instruction

Ladder Diagram: LIFO (D1) (D2) (S)	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: **LIFO (D1) (D2) (S)**	**Step size**	**7**

Operand	Type	Applicable devices														Index
D1	INT								D				V		R	√
D2	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

D1:The number of elements in the queue, and its element number + 1 element is the head element of the queue

D2:pop value storage unit

S: queue size

- **Function Description**

1. When the power flow is valid, the content of the top unit with D1 as the bottom of the stack is assigned to the D2 unit, and the value of D1 is decremented by 1.

2. When the value of D1 is equal to 0, the stack is emptied, and the Zero flag (SM80) is set to 1.

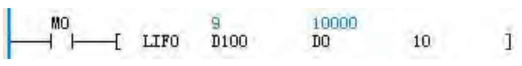
- **Precautions**

1. When the operation stack definition is illegal, (when the stack size is less than or equal to zero, the number of elements in the stack is less than zero. The number of elements in the stack is greater than the limit of the stack size), an illegal operation stack definition error is reported.

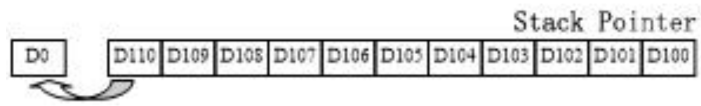
2. The size of the stack does not include the element at the bottom of the stack (the element specified by D1).

3. S specifies the size of the stack, the range is greater than 0.

- **Example of use**



LD M0
LIFO D100 D0 10



1. When M0=ON, the content of D110 is assigned to D0, and the content of units D101~D110 remains unchanged.

2. Before execution: D0=0, D100=10, D101=1000, D102=2000, ..., D109=9000, D110=10000.

3. After execution: D0=10000, D100=9, D101=1000, D102=2000, ..., D109=9000, D110=10000.

6.2.13 WSFR: String right shift instruction

Ladder Diagram: ⊢┤ ├─[WSFR (S1) (D) (S2) (S3)	Applicable models	VC-B VC-S VC-P
	Affect the flag	Carry flag Borrow flag
Command list: WSFR *(S1)* *(D)* *(S2)* (S3)	Step size	9

Oper and	Type	Applicable devices														Index
S1	INT		Kn X	KnY	KnM	KnS	KnL M		D	SD	C	T	V		R	√
D	INT			KnY	KnM	KnS	KnL M		D		C	T	V		R	√
S2	INT	Cons tant	Kn X	KnY	KnM	KnS	KnL M	KnS M	D	SD	C	T	V	Z	R	√
S3	INT	Cons tant	Kn X	KnY	KnM	KnS	KnL M	KnS M	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand

D: Destination operand, string start element

S2: The size of the destination word queue

S3: Shift right to fill in the number of words

- **Function Description**

When the power flow is valid, the contents of S2 units starting from D unit are shifted to the right by S3 units in word units, and the rightmost S3 data will be discarded. At the same time, the contents of S3 units starting with S1 unit will be moved into The left end of the string.

- **Precautions**

1. In the left and right order, the small component number is on the right, and the large component number is on the left.

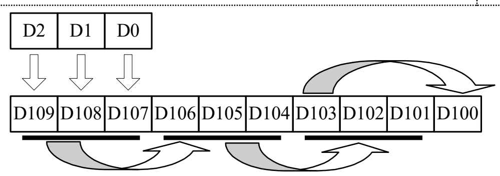
2. S2 is greater than or equal to 0, and S3 is greater than or equal to 0.

3. S2 is greater than or equal to S3.

4. When S1 and D address Kn at the same time, Kn should be equal

- **Example of use**

LD X0
WSFR D0 D100 10 3



1. When M0=ON, the contents of the 10 units starting from the D100 unit are shifted to the right by 3 units in word units, and the data of the rightmost units D102 to D100 are discarded. At the same time, the contents of the 3 cells starting at cell D0 are shifted to the left end of the string.

2. Before execution: D2=300, D1=200, D0=100. D109=10000, D108=9000, D107=8000, D106=7000, D105=6000, D104=5000, D103=4000, D102=3000, D101=2000, D100=1000.

3. After execution: the contents of D0~D2 remain unchanged. D2=300, D1=200, D0=100. D109=300, D108=200, D107=100, D106=10000, D105=9000, D104=8000, D103=7000, D102=6000, D101=5000, D100=4000.

6.2.14 WSFL: String left shift command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
⊢┤ ├─[WSFL (S1) (D) (S2) (S3)]	Affect the Flag	Zero flag Carry flag Borrow flag
Command list: WSFL *(S1)* *(D)* *(S2)* (S3)	Step size	9

Operand	Type	Applicable devices														Index
S1	INT		KnX	KnY	KnM	KnS	KnLM		D	SD	C	T	V		R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand

D: Destination operand, string start element

S2: Destination word queue size

S3: Shift right to fill in the number of characters

- **Function Description**

When the power flow is valid, move the contents of the S2 units starting from the D unit to the left by S3 units in word units, and the leftmost S3 data will be discarded. At the same time, the contents of the S3 units starting with the S1 unit will be moved into The right end of the string.

- **Precautions**

1. In the left and right order, the small component number is on the right, and the large component number is on the left.

2. S2 is greater than or equal to 0; S3 is greater than or equal to 0.

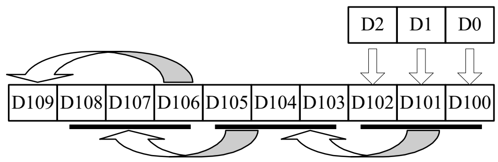
3. S2 is greater than or equal to S3.

4. When S1 and D address Kn at the same time, Kn should be equal.

- **Example of use**

LD X0

WSFL D0 D100 10 3



1. When X0=ON, the contents of the 10 units starting from the D100 unit are shifted to the left by 3 units in word units, and the data of the leftmost units D109~D107 will be discarded. At the same time, the contents of the 3 units starting from the D0 unit will be Shift into the right end of the string.

2. Before execution: D0=100, D1=200, D2=300. D109=10000, D108=9000, D107=8000, D106=7000, D105=6000, D104=5000, D103=4000, D102=3000, D101=2000, D100=1000.

3. After execution: the contents of D0~D2 remain unchanged. D2=300, D1=200, D0=100. D109=7000, D108=6000, D107=5000, D106=4000, D105=3000, D104=2000, D103=1000, D102=300, D101=200, D100=100.

6.3 Integer Arithmetic Instructions

6.3.1 ADD: Integer Addition Instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┫├─┤├─[ADD (S1) (S2) (D)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: ADD *(S1)* *(S2)* *(D)*	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***S2*:** Source operand 2

***D*:** Destination operand

Function Description

1. When the energy flow is valid, S1 plus S2, the result of the operation is given to D.

2. If the operation result (D) is greater than 32767, set the incoming flag bit (SM81). If the result is equal to 0, set the zero flag bit (SM80). If the result is less than -32768, set the debit flag bit (SM82).

- **Example of use**

LD X0
ADD D0
D1 D10

When X0=ON, D0 (1000) plus D1 (2000) result is assigned to D10,D10=3000.

6.3.2 SUB: Integer Subtraction Instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┫├─┤├─[SUB (S1) (S2) (D)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: SUB *(S1)* *(S2)* *(D)*	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

- **Example of use**

LD X0
SUB D0
D1 D10

D: Destination operand

- **Function Description**

1. When the power flow is valid, S1 is subtracted from S2, and the operation result is assigned to D.

2. When the operation result (D) is greater than 32767, the Carry flag bit (SM81) is set. When the operation result is equal to 0, the Zero flag (SM80) is set. When the operation result is less than -32768, the borrow Flag bit (SM82) is set.

When X0=ON, D0(1000) minus D1(2000) result is assigned to D10, D10=－1000.

6.3.3 MUL: Integer Multiplication Instruction

Ladder Diagram: ├─┤ ├─[MUL *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: **MUL *(S1)* *(S2)* *(D)***	**Step size**	**8**

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

- **Function Description**

When the power flow is valid, S1 is multiplied by S2, and the operation result is assigned to D.

Precautions

The operation result of the MUL instruction is 32-bit data.

- **Example of use**

LD X0

MUL D0 D1 D10

When X0=ON, the result of multiplying D0 (1000) by D1 (2000) is assigned to (D10, D11), (D10, D11) = 2000000.

6.3.4 DIV: Integer Division Instruction

Ladder Diagram: ├─┤ ├─[DIV *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: **DIV *(S1)* *(S2)* *(D)***	**Step size**	**7**
Operand	Type	Applicable devices	Index

S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

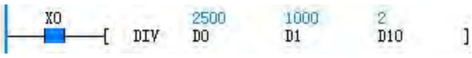
- **Function Description**

When the power flow is valid, S1 is divided by S2, and the operation result is assigned to D (D includes two units, the first unit stores the quotient value, and the second unit stores the remainder value).

Precautions

S2≠0, otherwise an error of division by 0 is reported, and the division operation is not performed.

- **Example of use**



LD X0

DIV D0

D1 D10

When X0=ON, the result of dividing D0 (2500) by D1 (1000) is assigned to (D10, D11). D10=2, D11=500.

6.3.5 SQT: Integer Arithmetic Square Root Instruction

Ladder Diagram: ├─┤ ├─[SQT *(S)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: SQT *(S) (D)*	**Step size**	**5**

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, S is squared, and the operation result is assigned to D.

2. When the operation result (D) is equal to 0, the Zero flag (SM80) is set. When the operation result is rounded off to decimals, the borrow flag (SM82) is set.

- **Precautions**

S≥0, otherwise an operand error is reported, and the square root operation is not performed.

- **Example of use**

LD X0

SQT D0 D10

When X0=ON, the result of the square root of D0 (1000) is assigned to D10, and D10=31.

6.3.6 INC: Integer plus one instruction

Ladder Diagram: ├──┤ ├──[INC (D)]										Applicable models		VC-B VC-S VC-P				
										Affect the flag						
Command list: INC (D)										Step size		3				
Operand	Type	Applicable devices														Index
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

- **Function Description**

When the power flow is valid, D is incremented by 1.

- **Precautions**

This instruction is a loop addition instruction, the range is -32768～32767; the supported range of C components is: C0～C199.

- **Example of use**

LD X0
INC D0

When X0=ON, D0 (1000) is incremented by 1, and D0=1001 after execution.

6.3.7 DEC: Integer minus one instruction

Ladder Diagram: ├──┤ ├──[DEC (D)]										Applicable models		VC-B VC-S VC-P				
										Affect the flag						
Command list: DEC (D)										Step size		3				
Operand	Type	Applicable devices														Index
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

D: Destination operand

- **Function Description**

When the power flow is valid, D is decremented by 1.

- **Precautions**

This command is cyclic subtraction, the range is -32768～32767.

- **Example of use**

LD X0
DEC D0

When X0=ON, D0 (1000) is decremented by 1, and D0=999 after execution.

.

6.3.8 VABS: Integer Absolute Value Instruction

Ladder Diagram: ├──┤ ├──[VABS (S) (D)]										Applicable models		VC-B VC-S VC-P				
										Affect the flag						
Command list: VABS *(S)* *(D)*										Step size		5				
Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand

D: Destination operand

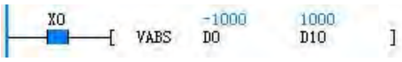
- **Function Description**

When the power flow is valid, S takes the absolute value, and the result is assigned to D.

- **Precautions**

SThe range of S should be -32767～32767; when the value of S is -32768, an illegal operand error will be reported, and the instruction will not take action.

- **Example of use**



LD X0

VABS D0 D10

When X0=ON, D0 (-1000) takes the absolute value, the result is assigned to D10, D10=1000.

6.3.9 NEG: Integer Negation Instruction

Ladder Diagram: ┣━┫ ┣━[NEG *(S)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**	
	Affect the flag		
Command list : NEG *(S)* *(D)*	**Step size**	**5**	

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S takes the opposite number, and the result is assigned to D.

- **Precautions**

S The range should be -32767～32767; when the value of S is -32768, an illegal operand error will be reported, and the instruction will not take action

- **Example of use**

LD X0

NEG D0

D10

When X0=ON, D0 (1000) takes the opposite number, the result is assigned to D10, D10=-1000.

6.3.10 DADD: Add Long Integer Instruction

Ladder Diagram: ┣━┫ ┣━[DADD *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: DADD *(S1)* *(S2)* *(D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

- **Function Description**

1. When the power flow is valid, S1 is added to S2, and the operation result is assigned to D.

2. When the operation result (D) is greater than 2147483647, the Carry flag (SM81) is set. When the operation result is equal to 0, the Zero flag (SM80) is set. When the operation result is less than -2147483648, the borrow Flag bit (SM82) is set.

- **Example of use**

LD X0

DADD D0 D2 D10

When X0=ON, the value (100000) of (D0, D1) is added to the value (200000) of (D2, D3), and the result is assigned to (D10, D11), (D10, D11) = 300000.

6.3.11 DSUB: Long Integer Subtraction Instruction

Ladder Diagram: ├─┤ ├─[DSUB *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: DSUB *(S1)* *(S2)* *(D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

- **Function Description**

1. When the power flow is valid, S1 is subtracted from S2, and the operation result is assigned to D.

2. When the operation result (D) is greater than 2147483647, the Carry flag (SM81) is set. When the operation result is equal to 0, the Zero flag (SM80) is set. When the operation result is less than -2147483648, set the borrow flag (SM82)

- **Example of use**

LD X0

DSUB D0 D2 D10

When X0=ON, the value (100000) of (D0, D1) subtracts the value (200000) of (D2, D3), and the result is assigned to (D10, D11), (D10, D11) = -100000.

6.3.12 DMUL: Long Integer Multiplication Instruction

Ladder Diagram: ├─┤ ├─[DMUL *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: DMUL *(S1)* *(S2)* *(D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

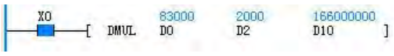
Function Description

When the power flow is valid, S1 is multiplied by S2, and the operation result is assigned to D.

- **Precautions**

The operation result of the DMUL instruction is 32-bit data, which may overflow, please pay attention to it.

- **Example of use**



LD X0

DMUL D0 D2 D10

When X0=ON, the value (83000) of (D0, D1) is multiplied by the value (2000) of (D2, D3) and the result is assigned to (D10, D11), (D10, D11) = 1660000000.

6.3.13 DDIV: Divide Long Integer Instructions

Ladder Diagram: ├─┤ ├─[DDIV *(S1)* *(S2)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: DDIV *(S1)* *(S2)* *(D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

- **Function Description**

When the power flow is valid, S1 is divided by S2, and the operation result is assigned to D (D includes 4 units, the first two units store the quotient value, and the last two units store the remainder value)

- **Precautions**

S2≠0, otherwise a division by 0 error is reported, and the division operation is not performed.

- **Example of use**

LD X0

DDIV D0 D2 D10

When X0=ON, the value (83000) of (D0, D1) is divided by (D2, D3) (2000) and the result is assigned to (D10, D11), (D12, D13). (D10, D11) = 41, (D12, D13) = 1000.

6.3.14 DSQT: long integer arithmetic square root instruction

Ladder Diagram: ├─┤ ├─[DSQT (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DSQT (S) (D)	Step size	7

Operand	Type	Applicable devices													Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, S is squared, and the operation result is assigned to D.

2. When the operation result (D) is equal to 0, the Zero flag (SM80) is set. When the operation result is rounded off to decimals, the borrow Flag bit (SM82) is set.

- **Precautions**

S≥0, otherwise an operand error is reported, and the square root operation is not performed.

- **Example of use**

LD X0

DSQT D0 D10

When X0=ON, the value (83000) of (D0, D1) is squared, and the result is assigned to (D10, D11), (D10, D11) =288.

6.3.15 DINC: increment long integer by one instruction

Ladder Diagram: ├─┤ ├─[DINC (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: DINC (D)	Step size	4

Operand	Type	Applicable devices													Index
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

D: Destination operand

Function Description

When the power flow is valid, D is incremented by 1.

- **Precautions**

1. This instruction is a cyclic addition instruction, the range is -2147483648～2147483647.

2. C components only support 32-bit C components.

- **Example of use**

LD X0

DINC D0

When X0=ON, the value (100000) of (D0, D1) is incremented by 1, after execution (D0, D1)=100001.

6.3.16 DDEC: long integer minus one instruction

Ladder Diagram: ├──┤ ├──[DDEC (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: DDEC (D)	Step size	4

Operand	Type	Applicable devices														Index
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

D: Destination operand

- **Function Description**

When the power flow is valid, D is decremented by 1.

- **Precautions**

This instruction is cyclic subtraction, the range is -2147483648～2147483647.

- **Example of use**

LD X0
DDEC D0

When X0=ON, the value (100000) of (D0, D1) is decremented by 1, and after execution (D0, D1) = 99999.

6.3.17 DVABS: Long Integer Absolute Value Instruction

Ladder Diagram: ├──┤ ├──[DVABS (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DVABS (S) (D)	Step size	7

Operand	Type	Applicable devices													Index	
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S takes the absolute value, and the result is assigned to D.

- **Precautions**

S The range should be -2147483647～2147483647; when the value of S is -2147483648, an illegal operand error is reported, and the instruction does not take action

- **Example of use**

LD X0

DVABS D0 D10

When X0=ON, the value (-100000) of (D0, D1) takes the absolute value, and the result is assigned to (D10, D11), (D10, D11)=100000.

6.3.18 DNEG: Negative Long Integer Instruction

Ladder Diagram: ├──┤ ├──[DNEG (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DNEG (S) (D)	Step size	7

Operand	Type	Applicable devices												Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S takes the opposite number, and the result is assigned to D.

- **Precautions**

The range of S should be -2147483647～2147483647; when the value of S is -2147483648, an illegal operand error is reported, and the instruction does not take action.

- **Example of use**

LD X0

DNEG D0 D10

When X0=ON, (D0, D1) (100000) takes the opposite number, and the result is assigned to (D10, D11), (D10, D11) = -100000.

6.3.19 SUM: Integer accumulation instruction

Ladder Diagram: SUM (S1) (S2) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: SUM (S1) (S2) (D)	Step size	8

Operand	Type	Applicable devices														Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand, accumulation start unit

S2: Source operand, the number of accumulated data

D: Destination operand, accumulation result

- **Function Description**

When the power flow is valid, the contents of the S2 units starting from the start unit (S1) will be accumulated, and the accumulated operation result will be assigned to the D unit.

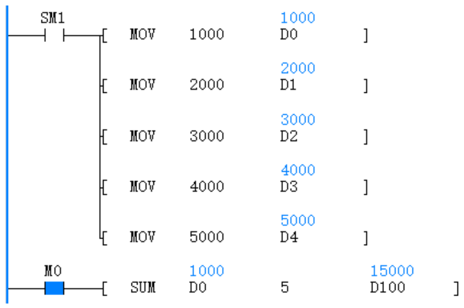
- **Precautions**

1. The operation result of the SUM instruction is 32-bit data.

2. 0≤S2≤255, otherwise an operand error is reported.

3. Since D is 32-bit data, the carry and borrow flags are always 0. The Zero flag is determined according to the final accumulation result.

- **Example of use**



LD SM1
MOV 1000 D0
MOV 2000 D1
MOV 3000 D2
MOV 4000 D3
MOV 5000 D4
LD M0
SUM D0 5 D100

When M0=ON, accumulate the data of 5 units starting from D0, and assign the result to (D100, D101). (D100, D101) =D0+……+D4=15000.

6.3.20 DSUM: Long Integer Accumulation Instruction

Ladder Diagram: DSUM (S1) (S2) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DSUM (S1) (S2) (D)	Step size	9

Operand	Type	Applicable devices														Index
S1	DINT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S1: Source operand, accumulation start unit

S2: Source operand, the number of accumulated data

- **Example of use**

LD SM0
DMOV 100000 D0
DMOV 200000 D2
DMOV 300000 D4

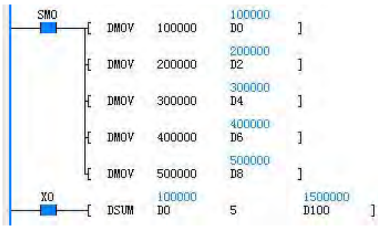
D: Destination operand, accumulation result

- **Function Description**

When the power flow is valid, the contents of the S2×2 units starting from the accumulation start unit (S1) are assigned to the D unit according to the result of the accumulation operation of the long integer data.

- **Precautions**

0≤S2≤255, otherwise an operand error is reported.



DMOV 400000 D6
DMOV 500000 D8
LD X0
DSUM D0 5 D100

When X0=ON, accumulate the long integers of 5×2 units starting from D0, and assign the result to (D100, D101). (D100, D101) = (D0, D1) +... + (D8, D9) = 1500000.

6.4 Floating-Point Arithmetic Instructions

6.4.1 RADD: Floating-point addition instruction

Ladder Diagram: ├─┤ ├─[RADD *(S1)* *(S2)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: RADD *(S1)* *(S2)* *(D)*	**Step size**	**10**

Operand	Type	Applicable devices													Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

- **Function Description**

1. When the power flow is valid, S1 is added to S2, and the operation result is assigned to D.

2. When the operation result (D) is greater than 1.701412e+038 or less than

-1.701412e+038, set the Carry flag (SM81). When the operation result is equal to 0, the Zero flag (SM80) is set.

- **Example of use**

LD X0
RAD DD0
D2 D10

When X0=ON, the value (-10000.2) of (D0, D1) is added to the value (2000.5) of (D2, D3), and the result is assigned to (D10, D11), (D10, D11) =-7999.7.

6.4.2 RSUB: Floating-point subtraction instruction

Ladder Diagram: ├─┤ ├─[RSUB *(S1)* *(S2)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: RSUB *(S1)* *(S2)* *(D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

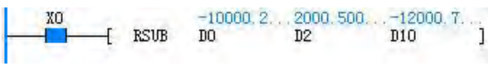
D: Destination operand

- **Function Description**

1. When the power flow is valid, S1 is subtracted from S2, and the operation result is assigned to D.

2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the Carry flag (SM81) is set. When the operation result is equal to 0, the Zero flag (SM80) is set.

- **Example of use**



LD X0

RSUB D0 D2 D10

When X0=ON, the value (-10000.2) of (D0, D1) subtracts the value (2000.5) of (D2, D3), and the result is assigned to (D10, D11), (D10, D11)=-12000.7.

6.4.3 RMUL: Floating-point multiplication instruction

Ladder Diagram: ├─┤ ├─[RMUL *(S1)* *(S2)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: RMUL *(S1)* *(S2)* *(D)*	Step size	10

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

S1: Source operand 1

S2: Source operand 2

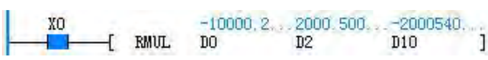
D: Destination operand

- **Function Description**

1. When the power flow is valid, S1 is multiplied by S2, and the operation result is assigned to D.

2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, set the Carry flag (SM81). When the operation result is equal to 0, the Zero flag (SM80) is set.

- **Example of use**



LD X0

RMUL D0 D2 D10

When X0=ON, the value (-10000.2) of (D0, D1) is multiplied by the value (2000.5) of (D2, D3) and the result is assigned to (D10, D11), (D10, D11) =-20005400.0 (actually the product should be -20005400.1, and 0.1 was discarded due to the measurement accuracy).

6.4.4 RDIV: Floating Point Divide Instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├─┤ ├─[RDIV (S1) (S2) (D)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: RDIV (S1) (S2) (D)	Step size	10

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

● **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

● **Function Description**

1. When the power flow is valid, S1 is divided by S2, and the operation result is assigned to D
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, set the Carry flag (SM81). When the operation result is equal to 0, set the Zero flag (SM80)

● **Precautions**

S2≠0, otherwise an error of division by 0 is reported, and the division operation is not performed.

● **Example of use**

LD X0
RDIV D0 D2 D10

When X0=ON, (D0, D1) = -10000.2 divided by (D2, D3) = 2000.5 and the result is assigned to (D10, D11). (D10,D11)=-4.998850.

6.4.5 RSQT: Floating-point arithmetic square root instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├─┤ ├─[RSQT (S) (D)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: RSQT (S) (D)	Step size	7

Operand	Type	Applicable devices														Index
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

● **Operand Description**

S: Source operand

D: Destination operand

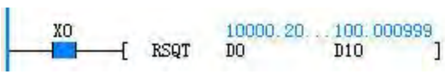
● **Function Description**

1. When the power flow is valid, S is squared, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the Zero flag (SM80) is set.

● **Precautions**

S≥0, otherwise an operand error is reported, and the square root operation is not performed.

● **Example of use**



LD X0
RSQT D0 D10

When X0=ON, the value (10000.2) of (D0, D1) is squared, and the result is assigned to (D10, D11), (D10, D11)=100.000999.

6.4.6 RVABS: Floating point absolute value instruction

Ladder Diagram: ├──┤ ├──[RVABS (S) (D)]																
Applicable models	**VC-B VC-S VC-P**															
Affect the flag																
Command list: RVABS (S) (D)	**Step size**	**7**														
Operand	Type	Applicable devices														Index
S	REAL	Constant						D				V		R		√
D	REAL							D				V		R		√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S takes the absolute value, and the result is assigned to D.

- **Example of use**

LD X0

RVABS D0 D10

When X0=ON, the value (-10000.2) of (D0, D1) takes the absolute value, and the result is assigned to (D10, D11), (D10, D11) = 10000.2.

6.4.7 RNEG: Negative floating-point number instruction

Ladder Diagram: ├──┤ ├──[RNEG (S) (D)]																
Applicable models	**VC-B VC-S VC-P**															
Affect the flag																
Command list: RNEG (S) (D)	**Step size**	**7**														
Operand	Type	Applicable devices														Index
S	REAL	Constant						D				V		R		√
D	REAL							D				V		R		√

- **Example of use**

LD X0

RNEG D0 D10

When X0=ON, (D0, D1) =10000.2 takes the opposite number, and the result is assigned to (D10, D11), (D10, D11) =-10000.2.

6.4.8 SIN: Floating-point number SIN instruction

Ladder Diagram: ├──┤ ├──[SIN (S) (D)]																
Applicable models	**VC-B VC-S VC-P**															
Affect the flag	**Zero flag Carry flag Borrow flag**															
Command list: SIN (S) (D)	**Step size**	**7**														
Operand	Type	Applicable devices														Index
S	REAL	Constant						D				V		R		√
D	REAL							D				V		R		√

- **Operand Description**

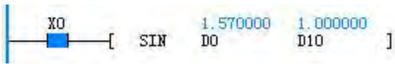
- **Example of use**

S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, find the SIN value of S (unit is radian), and assign the result to D.
2. When the operation result (D) is equal to 0, set the Zero flag bit SM80



LD X0
SIN D0 D10

When X0=ON, (D0, D1)=1.57 takes the SIN value, the result is assigned to (D10, D11), (D10, D11)=1

6.4.9 COS: floating point number COS instruction

Ladder Diagram: COS (S) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: COS (S) (D)	Step size	7

Operand	Type	Applicable devices														Index
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

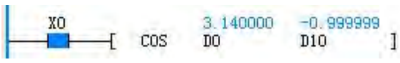
S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, find the COS value of S (unit is radian), and assign the result to D.
2. When the operation result (D) is equal to 0, the Zero flag (SM80) is set.

- **Example of use**



LD X0
COS D0 D10

When X0=ON, (D0, D1)=3.14 to calculate the COS value, the result is assigned to (D10, D11), (D10, D11)=-0.999999.

6.4.10 TAN: floating point number TAN instruction

Ladder Diagram: TAN (S) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: TAN *(S)* *(D)*	Step size	7

Operand	Type	Applicable devices														Index
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

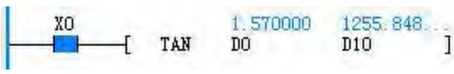
S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, find the TAN value of S (unit is radian), and assign the result to D.

- **Example of use**



LD X0
TAN D0 D10

When X0=ON, (D0, D1)=1.57 to find the TAN value, the result is assigned to (D10, D11), (D10, D11)=1255.848398.

2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the Carry flag (SM81) is set. When the operation result is equal to 0, set the Zero flag (SM80)

6.4.11 POWER: floating point number exponentiation operation

Ladder Diagram: ─┤ ├─[POWER *(S1)* *(S2)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: POWER *(S1)* *(S2)* *(D)*	Step size	10

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

● Operand Description

S1: Source operand 1

S2: Source operand 2

D: Destination operand

● Function Description

1. When the power flow is valid, take S1 to the power of S2, and assign D to the operation result.

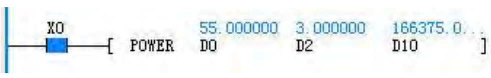
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the Carry flag (SM81) is set. When the operation result is equal to 0, the Zero flag (SM80) is set.

● Precautions

1. When S1=0 and S2≤0, the operand value error is reported and the operation is not performed.

2. When S1 < 0 and the mantissa part of S2 is not 0, the operand value error is reported, and the operation is not performed.

● Example of use



LD X0

POWER D0 D2 D10

When X0=ON, find the (D2, D3) power of (D0, D1) (that is, 55.0 to the 3.0th power), and assign the result to (D10, D11), (D10, D11) = 166375.0.

6.4.12 LN: Floating point natural logarithm instruction

Ladder Diagram: ─┤ ├─[LN *(S)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: LN *(S)* *(D)*	Step size	7

Operand	Type	Applicable devices														Index
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

● Example of use

- **Operand Description**

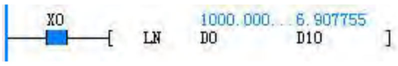
S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, find the LN value of S, and assign the result to D.

2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the Carry flag (SM81) is set. When the operation result is equal to 0, the Zero flag (SM80) is set.



LD X0
LN D0 D10

When X0=ON, (D0, D1)=1000.0 to find the LN value, the result is assigned to (D10, D11), (D10, D11)=6.907755.

6.4.13 EXP: Floating-point number natural number exponentiation instruction

Ladder Diagram: ├─┤ ├──[EXP (S) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	Zero flag Carry flag Borrow flag
Command list: EXP (S) (D)															Step size	7
Operand	Type	Applicable devices													Index	
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, find the EXP value of S, and assign the result to D.

2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the Carry flag (SM81) is set. When the operation result is equal to 0, set the Zero flag (SM80)

- **Example of use**

LD X0
EXP D0 D10

When X0=ON, (D0, D1)=10.0 to find the EXP value, the result is assigned to (D10, D11), (D10, D11)=22026.464844.

6.4.14 RSUM: Floating-point accumulation instruction

Ladder Diagram: ├─┤ ├──[RSUM (S1) (S2) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	Zero flag Carry flag Borrow flag
Command list: RSUM (S1) (S2) (D)															Step size	9
Operand	Type	Applicable devices													Index	
S1	REAL								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D				V		R	√
D	REAL								D				V		R	√

- **Operand Description**

S1: Source operand, accumulation start unit

S2: Source operand, the number of accumulated data.

D: Destination operand, accumulation result

- **Function Description**

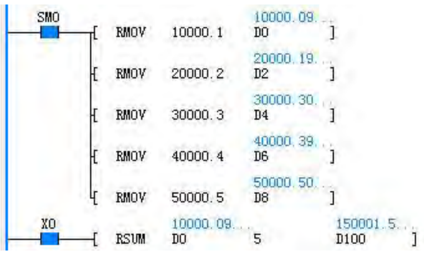
When the power flow is valid, the contents of the S2×2 units starting from the accumulation start unit (S1) are accumulated according to the floating-point data, and the result of the operation is assigned to the D unit.

- **Precautions**

1. 0≤S2≤255, otherwise an operand error is reported.

2. In case of overflow, the accumulation operation will no longer be performed

- **Example of use**



LD SM0
RMOV 10000.1 D0
RMOV 20000.2 D2
RMOV 30000.3 D4
RMOV 40000.4 D6
RMOV 50000.5 D8
LD X0
RSUM D0 5 D100

When X0=ON, the floating-point numbers of 5×2 units starting from D0 are accumulated, and the result is assigned to (D100, D101). (D100, D101) = (D0, D1) +... + (D8, D9) = 150001.5.

6.4.15 ASIN: Floating point number ASIN instruction

Ladder Diagram: ├──┤ ├──[ASIN *(S)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list:ASIN *(S)* *(D)*	**Step size**	**7**

Operand	Type	Applicable devices												Index
S	REAL	Constant						D				V	R	√
D	REAL							D				V	R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

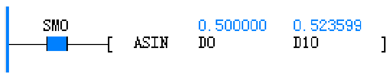
1. When the power flow is valid, find the SIN-1 value of S, and assign the result to D;

2. When the operation resul (D) is equal to 0, set the Zero flag (SM80);

- **Precautions**

When S>1 or S<-1, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**



LD SM0
ASIN D0 D10

When SM0=ON, (D0, D1) (0.500000) takes the value of SIN-1, and the result is assigned to (D10, D11), (D10, D11)=0.523599.

6.4.16 ACOS: Floating point number ACOS instruction

Ladder Diagram: ├──┤ ├──[ACOS *(S)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: ACOS *(S)* *(D)*	**Step size**	**7**

Operand	Type	Applicable devices	Index

S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

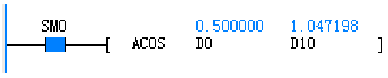
1. When the power flow is valid, find the COS-1 value of S, and assign the result to D;

2. When the operation result (D) is equal to 0, set the Zero flag (SM80);

- **Precautions**

When S>1 or S<-1, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**



LD SM0

ACOS D0 D10

When SM0=ON, (D0, D1)=0.500000 to find the value of COS-1, the result is assigned to (D10, D11), (D10, D11)=1.047198.

6.4.17 ATAN: Floating point ATAN instruction

Ladder Diagram: ATAN *(S)* *(D)*													**Applicable models**		**VC-B VC-S VC-P**
													Affect the flag		**Zero flag Carry flag Borrow flag**
Command list: ATAN *(S)* *(D)*													**Step size**		**7**
Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- **Operand Description**

S: Source operand

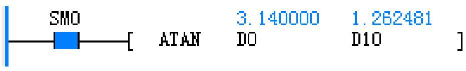
D: Destination operand

- **Function Description**

1. When the power flow is valid, find the TAN-1 value of S, and assign the result to D;

2. When the operation result (D) is equal to 0, set the Zero flag (SM80);

- **Example of use**



LD SM0

ATAN D0 D10

When SM0=ON, (D0, D1) (3.14) calculates the value of TAN-1, and assigns the result to (D10, D11), (D10, D11)=1.262481.

6.4.18 LOG: Common logarithmic operations on floating-point numbers

Ladder Diagram: LOG *(S)* *(D)*													**Applicable models**		**VC-S VC-P**
													Affect the flag		**Zero flag Carry flag Borrow flag**
Command list: LOG *(S)* *(D)*													**Step size**		**7**
Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- **Example of use**

- **Operand Description**

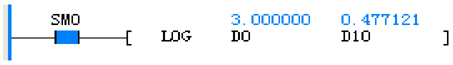
S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, find the LOG value of S, and assign the result to D. LOG is a common logarithmic operation with the base 10;

2. When the operation result (D) overflows, set the carry (overflow) Sign (SM81); when the operation result is equal to 0, set the Zero flag (SM80);



LD SM0
LOG D0 D10

When SM0=ON, the value of D0(D1) (3.0), the result is assigned to D10(D11), D10(D11) =0.477121.

6.4.19 RAD: Floating point angle->radian conversion

Ladder Diagram: ├──┤ ├──[RAD (S) (D)]	Applicable models	VC-S VC-P
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: RAD (S) (D)	**Step size**	**7**

Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- **Operand Description**

S: Source operand

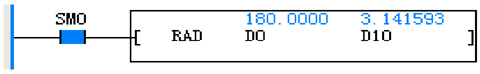
D: Destination operand

- **Function Description**

1. When the power flow is valid, convert the floating point angle value of the S unit into a radian value, and assign the result to D;

2. When the operation result is equal to 0, set the Zero flag (SM80);

- **Example of use**



LD SM0
RAD D0 D10

When SM0=ON, the value of D0(D1) (180.0), the result is assigned to D10(D11), D10(D11)=3.141593.

6.4.20 DEG: Floating point radian->angle conversion

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
├─┤ ├─[DEG *(S)* *(D)*]	**Affect the flag**	**Zero flag Carry flag Borrow flag**
Command list: DEG *(S)* *(D)*	**Step size**	**7**

Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

● **Operand Description**

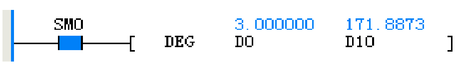
S: Source operand

D: Destination operand

● **Function Description**

1. When the power flow is valid, convert the floating point radian value of the S unit into an angle value, and assign the result to D;
2. When the operation result is equal to 0, set the Zero flag (SM80), when the operation result overflows, set the carry (overflow) Sign (SM81);

● **Example of use**



LD SM0
DEG D0 D10

When SM0=ON, the value of D0(D1) (3.0), the result is assigned to D10(D11), D10(D11)=171.8873.

6.5 Numeric Conversion Instructions

6.5.1 DTI: Long Integer Convert Integer Instruction

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
├─┤ ├─[DTI *(S)* *(D)*]	**Affect the flag**	**Zero flag Carry flag Borrow flag**
Command list: DTI *(S)* *(D)*	**Step size**	**6**

Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

● **Operand Description**

S: Source operand

D: Destination operand

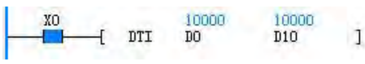
● **Function Description**

When the power flow is valid, S is converted from a long integer to an integer, and the result is assigned to D.

● **Precautions**

When S>32767 or S<-32768, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.

● **Example of use**



LD X0
DTI D0 D10

When X0=ON, (D0, D1)=10000 is converted from long integer to integer and assigned to D10. D10=10000.

6.5.2 ITD: Integer Convert Long Integer Instruction

Ladder Diagram: ├──┤ ├──[ITD (S) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	Zero flag Carry flag Borrow flag
Command list: ITD (S) (D)															Step size	6
Operand	Type	Applicable devices													Index	
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S is converted from an integer to a long integer, and the result is assigned to D.

- **Example of use**

LD X0
ITD D0 D10

When X0=ON, D0=1000 is converted from integer to long integer and assigned to D10, (D10, D11)=1000.

6.5.3 FLT: Integer to floating point instruction

Ladder Diagram: ├──┤ ├──[FLT (S) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	Zero flag Carry flag Borrow flag
Command list: FLT (S) (D)															Step size	6
Operand	Type	Applicable devices													Index	
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	REAL								D				V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S is converted from an integer to a floating-point number, and the result is assigned to D.

- **Example of use**

LD X0
FLT D0 D10

When X0=ON, D0=10005 is converted from integer to floating point number and assigned to (D10, D11), (D10, D11)=10005.0.

6.5.4 DFLT: Long Integer Convert Floating Point Number Instruction

Ladder Diagram: ├──┤ ├──[DFLT (S) (D)]															Applicable models	VC-B VC-S VC-P
															Affect the flag	Zero flag Carry flag Borrow flag
Command list: DFLT (S) (D)															Step size	7
Operand	Type	Applicable devices													Index	
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	REAL								D				V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S is converted from a long integer to a floating point number, and the result is assigned to D.

- **Example of use**

LD X0

DFLT D0 D10

When X0=ON, (D0, D1)=100000, convert from long integer to floating point number, assign it to (D10, D11), (D10, D11)=100000.0.

6.5.5 INT: Floating-point conversion integer instruction

Ladder Diagram: ├─┤ ├─[INT *(S)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: INT *(S)* *(D)*	Step size	6

Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V		R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, S is converted from a floating point number to an integer, and the result is assigned to D.

2. This instruction affects the Zero flag and the borrow flag. When the conversion result is zero, the Zero flag SM80 is set. When the result is rounded off to a decimal point, the borrow flag is set. When the result exceeds the data range of the integer data, the carry (overflow) Sign SM81 is set.

- **Precautions**

When S>32767, D=32767. When S<-32768, D=-32768, and the carry (overflow) Flag bit SM81 is set at the same time

- **Example of use**

LD X0

INT D0 D10

When X0=ON, (D0, D1)=10000.5, it is converted from floating point number to integer and assigned to D10, D10=10000.

6.5.6 DINT: Floating point number to long integer instruction

Ladder Diagram: ├─┤ ├─[DINT *(S)* *(D)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DINT *(S)* *(D)*	Step size	7

Operand	Type	Applicable devices													Index
S	REAL	Constant							D				V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

- **Precautions**

S: Source operand

D: Destination operand

- **Function Description**

1. When the power flow is valid, S is converted from a floating point number to a long integer, and the result is assigned to D.
2. When the conversion result is zero, the Zero flag SM80 is set. When the result is rounded off to a decimal point, the borrow flag is set. When the result exceeds the long integer data range, the carry (overflow) Sign is set.

When S>2147483647, D=2147483647. When S<-2147483648, D=-2147483648, and the carry (overflow) Flag bit SM81 is set at the same time.

- **Example of use**

LD X0
DINT D0 D10

When X0=ON, (D0, D1)=100000.5 is converted from a floating point number to a long integer, and assigned to (D10, D11), (D10, D11)=100000.

6.5.7 BCD: Word conversion 16-bit BCD code instruction

Ladder Diagram: ├──┤ ├──[BCD (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: BCD (S) (D)	Step size	5

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand, ≤9999

D: Destination operand

Function Description

When the power flow is valid, S is converted from an integer to a 16-bit BCD code, and the result is assigned to D.

- **Precautions**

When S>9999, the system reports an Operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**

LD X0
BCD D0 D10

When X0=ON, D0=0x0D05 (3333) is converted from integer to 16-bit BCD code and assigned to D10, D10=0x3333 (13107).

6.5.8 DBCD: Double word conversion 32-bit BCD code instruction

Ladder Diagram: ├[DBCD (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DBCD (S) (D)	Step size	7

Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand, ≤99999999

D: Destination operand

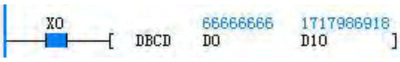
- **Function Description**

When the power flow is valid, S is converted from a long integer to a 32-bit BCD code, and the result is assigned to D.

- **Precautions**

When S>99999999, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**



LD X0

DBCD D0 D10

When X0=ON, (D0, D1)=0x3F940AA (66666666) is converted from long integer to 32-bit BCD code, and assigned to (D10, D11), (D10, D11)=0x66666666 (1717986918).

6.5.9 BIN: 16-bit BCD code conversion word command

Ladder Diagram: ├[BIN (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: BIN (S) (D)	Step size	5

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand, data format must conform to BCD code format

D: Destination operand.

- **Function Description**

When the power flow is valid, S is converted into an integer by 16-bit BCD code, and the result is assigned to D.

- **Precautions**

When the data format of S does not conform to the BCD code format, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**

LD X0

BIN D0 D10

When X0=ON, D0=0x5555 (21845) is converted into an integer by 16-bit BCD code and assigned to D10, D10=0x15B3 (5555).

6.5.10 DBIN: 32-bit BCD code conversion double word instruction

Ladder Diagram: ├─┤ ├─[DBIN (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: DBIN (S) (D)	Step size	7

Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

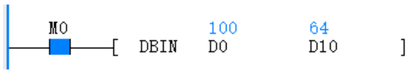
1. When the power flow is valid, S is converted into a long integer by 32-bit BCD code, and the result is assigned to D.

2. The data format of S must conform to the BCD code format

- **Precautions**

When the data format of S does not conform to the BCD code format, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**



LD M0
DBIN D0 D10

When M0=ON, (D0, D1) = 0x64 (100) is converted into a long integer by 32-bit BCD code, and assigned to (D10, D11), (D10, D11) = 0x40 (64).

6.5.11 GRY: Word conversion 16-bit gray code instruction

Ladder Diagram: [GRY (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: GRY (S) (D)	Step size	5

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

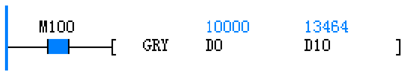
S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S is converted from an integer to a 16-bit Gray code, and the result is assigned to D.

- **Example of use**



LD M100
GRY D0 D10

When M100=ON, D0=0x2710 (10000) is converted from integer to 16-bit Gray code and assigned to D10, D10=0x3498 (13464).

6.5.12 DGRY: Double word conversion 32-bit Gray code instruction

Ladder Diagram: [DGRY (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list : DGRY (S) (D)	Step size	7

Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S is converted from a long integer to a 32-bit Gray code, and the result is assigned to D.

- **Example of use**

LD X0

DGRY D0 D10

When X0=ON, (D0, D1)=0x7A120 (500000) is converted from long integer to 32-bit Gray code, and assigned to (D10, D11), (D10, D11)=0x471B0 (291248).

6.5.13 GBIN: 16-bit Gray code conversion word command

Ladder Diagram: ├─┤ ├──[GBIN *(S)* *(D)*]										**Applicable models**		**VC-B VC-S VC-P**			
										Affect the flag		**Zero flag Carry flag Borrow flag**			
Command list: GBIN *(S) (D)*										**Step size**		**5**			
Operand	Type	Applicable devices													Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

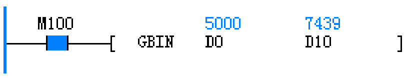
S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, S is converted into an integer by 16-bit Gray code, and the result is assigned to D.

- **Example of use**



LD M100

GBIN D0 D10

When M100=ON, D0=0x1388 (5000) is converted into an integer by 16-bit Gray code and assigned to D10, D10=0x1D0F (7439).

6.5.14 DGBIN: 32-bit Gray code conversion double word instruction

Ladder Diagram: ├─┤ ├──[DGBIN *(S)* *(D)*]										**Applicable models**		**VC-B VC-S VC-P**				
										Affect the flag		**Zero flag Carry flag Borrow flag**				
Command list: DGBIN *(S) (D)*										**Step size**		**7**				
Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- **Operand Description**

S: Source operand

D: Destination operand

- **Function Description**

- **Example of use**

LD X0

DGBIN D0 D10

When X0=ON, (D0, D1)=0xA0012 (655378) is converted into a long integer by 32-bit Gray code, and assigned to (D10, D11), (D10, D11)=0xC001C (786460).

When the power flow is valid, S is converted to a long integer by 32-bit Gray code, and the result is assigned to D.

6.5.15 SEG: Word conversion 7-segment code instruction

Ladder Diagram: ├─┤ ├─[SEG (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: SEG (S) (D)	Step size	5

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S: Source operand, S≤15

D: Destination operand

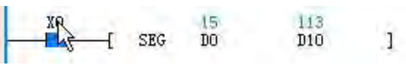
- **Function Description**

When the power flow is valid, S is converted from an integer to a 7-segment code, and the result is assigned to D.

- **Precautions**

When S>15, the system reports an Operand error, does not perform conversion, and the content of D remains unchanged.

- **Example of use**



LD X0

SEG D0 D10

When X0=ON, D0=0x0F (15) is converted from integer to 7-segment code and assigned to D10, D10=0x71 (113).

6.5.16 ASC: ASCII code conversion command

Ladder Diagram: ├─┤ ├─[ASC (S1~S8) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list:ASC (S1～S8) (D)	Step size	19

operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	INT	Constant														
S3	INT	Constant														
S4	INT	Constant														

S5	INT	Constant														
S6	INT	Constant														
S7	INT	Constant														
S8	INT	Constant														
D	INT								D		C	T	V	Z	R	√

- **Operand Description**

S1~S8: source operands (less than 8, the remaining characters are filled with 0)

Only supports characters whose ASCII codes are 0x21～0x7E (keyboard input, supplement with 0X00 if less than 8 characters)

D: Destination operand

- **Function Description**

When the power flow is valid, the strings S1 to S8 are converted into ASCII codes, and the result is assigned to the D starting element. When SM85=OFF, the high and low bytes of each D element store two ASCII code data, when SM85=ON, each D element low byte stores one ASCII code data.

- **Example of use**

LD M0
ASC 12345678 D0

When M0=ON, ASCII conversion is performed, and the data is stored in two ways:

(1) If SM85=OFF, the execution result is: D0=0x3231, D1=0x3433, D2=0x3635, D3=0x3837.

(2) If SM85=ON, the execution result is: D0=0x31, D1=0x32, D2=0x33, D3=0x34, D4=0x35, D5=0x36, D6=0x37, D7=0x38.

6.5.17 ITA: 16-bit hexadecimal number conversion ASCII code command

Ladder Diagram: ├──┤ ├──[ITA (S1) (D) (S2)]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: ITA *(S1) (D) (S2)*	**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source hexadecimal data to be converted;

D: Destination operand

S2: Number of ASCII codes (1≤***S2***≤256)

- **Function Description**

When the power flow is valid, convert the hexadecimal numbers starting from the S1 element into S2 ASCII

- **Precautions**

1. When S1 and D use Kn addressing, Kn=4.
2. When S2 is not between 1 and 256, the system reports an operand error, does not perform conversion, and the content of D remains unchanged.
3. If S1 is Constant, S2≥4, the default S2=4 processing. By default, no more operand errors are reported.

- **Example of use**

LD M0
ITA 16#9876 D20 8

codes, and assign the result to the D starting element. When SM85=OFF, the high and low bytes of each D element store two ASCII code data, when SM85=ON, each D element low byte stores one ASCII code data.

When M0=ON, ITA conversion is performed, and the data is stored in two ways:

(1)If SM85=OFF, the execution result is: D20=0x3839, D21=0x3637.

(2)If SM85=ON, the execution result is: D20=0x39, D21=0x38, D22=0x37, D23=0x36.

6.5.18 ATI: ASCII code number conversion 16-bit hexadecimal command

Ladder Diagram: ├─┤ ├─[ATI (S1) (D) (S2)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: **ATI** ***(S1) (D) (S2)***	**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source ASCII code data to be converted 0x30≤S1≤0x39 or 0x41≤S1≤0x46 (when SM85=OFF, both high and low bytes of S1 need to meet this range)

D: Destination operand

S2: Number of ASCII codes (1≤***S2***≤256)

- **Function Description**

When the power flow is valid, the S2 ASCII code data starting from the S1 element is converted into hexadecimal data, and the result is stored in the D starting element every 4 bits. When SM85=OFF, the high and low bytes of each D element store two ASCII code data; when SM85=ON, each D element low byte stores one ASCII code data.

- **Precautions**

1. When S1 and D use Kn addressing, Kn=4.

2. When S1Not at 0x30～0x39 or 0x41～0x46, or when S2 is not between 1 and 256, the system reports an operand error, does not perform the conversion, and the content of D remains unchanged.

3. If S1 is Constant, when SM85=OFF and S2≥2, the default S2=2 processing. When SM85=ON and S2≥When 1, the default S2=1 processing. By default, no more operand errors are reported.

- **Example of use**

LD M0
ATI D10 D30 4

Source data: D10=0x3938, D11=0x3736, D12=0x3534, D13=0x3332.

When M0=ON, ATI conversion is performed, and the result is as follows according to the data storage method:

(1)If SM85=OFF, the execution result is: D30=0x8967.

(2)If SM85=ON, the execution result is: D30=0x8642.

6.5.19 LCNV: Project conversion command

Ladder Diagram: ├─┤ ├─[LCNV (S1) (S2) (D) (S3)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: LCNV ***(S1) (S2) (D) (S3)***	**Step size**	**9**

Operand	Type	Applicable devices													Index
S1	INT								D				V	R	
S2	INT								D				V	R	

D	INT								D				V	R	
S3	INT	Constant							D				V	R	

- **Operand Description**

S1: The starting address of the Source operand to be converted

S2: Conversion table start address

D: Store the starting address of the conversion result

S3: The number of data to be converted(1≤***S3***≤64)

- **Function Description**

When using the analog input module to read the external analog signal, this command can be used to convert the original analog reading value into the corresponding engineering reading value.

When using a temperature or analog module for temperature or analog measurement applications, if the temperature or engineering reading value measured by the PLC deviates from the results measured by a standard thermometer or related standard instruments, this instruction can be used to make linear corrections as Correction of actual measurements.

Fill in the low point measurement in the conversion table V_{ML} , high point measurement V_{MH} and the corresponding low point standard value V_{SL} Standard value with high point V_{SH} There are four parameters in total; when performing linear transformation, the source data is calculated by the following formula to generate the corresponding target standard value. in S_n for the original input data, D_n for the conversion result data.

$$A = (V_{SL} - V_{SH})/(V_{ML} - V_{MH}) * 10000$$

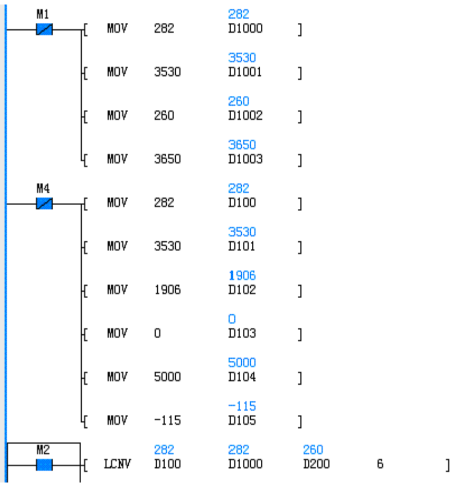
$$B = V_{SL} - (V_{ML} * A/10000)$$

$$D_n = (S_n * A/10000) + B$$

- **Precautions**

It is practical to convert the four data in the table, such as the low point measurement value should be less than the high point measurement value. Conversion results outside the integer range will be inaccurate. D_n If it is greater than 32767, it is 32767, and if it is less than -32768, it is -32768.

- **Example of use**



LDI M1
MOV 282 D1000
MOV 3530 D1001
MOV 260 D1002
MOV 3650 D1003
LDI M4
MOV 282 D100
MOV 3530 D101
MOV 1906 D102
MOV 0 D103
MOV 5000 D104
MOV -115 D105
LD M2
LCNV D100 D100 D1000 D200 6

When M2=ON, perform LCNV conversion, and according to the data storage method, the result is as follows:

D200 = 260

D201 = 3650

D202 = 1955

D203 = -34

D204 = 5184

D205 = -154

6.5.20 RLCNV: Floating point engineering conversion instruction

Ladder Diagram:	Applicable models	VC-S VC-P

├──┤ ├──[RLCNV (S1) (S2) (D) (S3)]	**Affect the flag**	**Zero flag Carry flag Borrow flag**
Command list: RLCNV *(S1) (S2) (D) (S3)*	**Step size**	**12**

Operand	Type	Applicable devices													Index
S1	REAL								D				V	R	
S2	REAL								D				V	R	
D	REAL								D				V	R	
S3	INT	Constant							D				V	R	

- **Operand Description**

S1: The starting address of the Source operand to be converted

S2: Conversion table start address

D: Store the starting address of the conversion result

S3:The number of data to be converted(1≤***S3***≤64)

- **Function Description**

When using the analog input module to read the external analog signal, this command can be used to convert the original analog reading value into the corresponding engineering reading value.

When using a temperature or analog module for temperature or analog measurement applications, if the temperature or engineering reading value measured by the PLC deviates from the results measured by a standard thermometer or related standard instruments, this instruction can be used to make linear corrections as Correction of actual measurements.

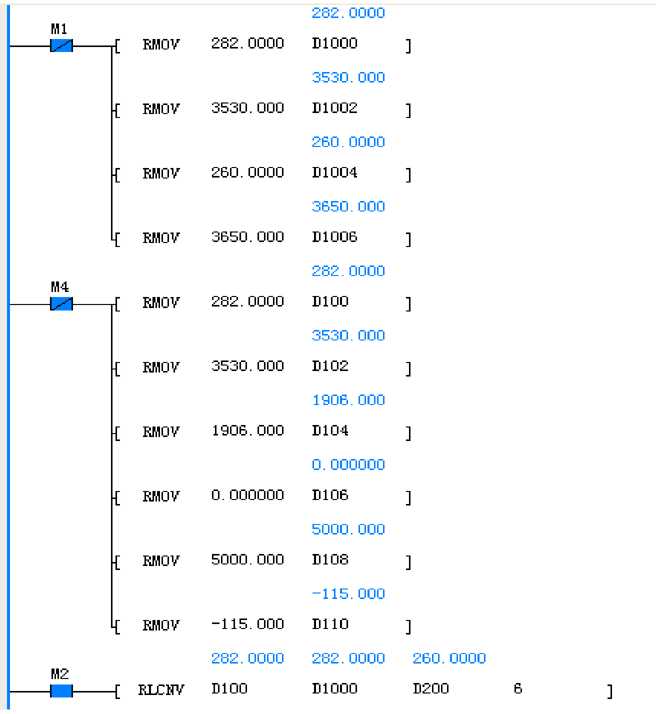
Fill in the low point measurement in the conversion table V_{ML} , high point measurement V_{MH} and the corresponding low point standard value V_{SL} Standard value with high point V_{SH} There are four parameters in total; when performing linear transformation, the source data is calculated by the following formula to generate the corresponding target standard value. in S_n for the original input data, D_n for the conversion result data.

$$A = (V_{\text{SL}} - V_{\text{SH}})/(V_{\text{ML}} - V_{\text{MH}}) * 10000$$

- **Precautions**

It is practical to convert the four data in the table, such as the low point measurement value should be less than the high point measurement value. Conversion results outside the integer range will be inaccurate. D_n If it is greater than 32767, it is 32767, and if it is less than -32768, it is -32768.

- **Example of use**



When M2=ON, perform RLCNV conversion, according to the data storage method, the result is as follows:

D200(D201) = 260

D202(D203) = 3650

D204(D205) = 1955

D206(D207) = -34.3288

D208(D209) = 5184.267

D210(D211) = -154.357

$$B = V_{SL} - (V_{ML} * A/10000)$$

$$D_n = (S_n * A/10000) + B$$

6.6 Word Logic Operations

6.6.1 WAND: Words and Instructions

Ladder Diagram: WAND (S1) (S2) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: *WAND (S1) (S2) (D)*	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

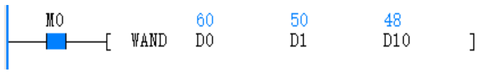
S2: Source operand 2

D: Destination operand

- **Function Description**

When the power flow is valid, S1 and S2 are bitwise (logical AND, and the result is assigned to D.

- **Example of use**



LD M0

WAND D0 D1 D10

When M0=ON, D0=2#0000000000111100 (60) and D1=2#0000000000110010 (50) bit logical AND, the result is assigned to D10, D10=2#0000000000110000 (48).

6.6.2 WOR: Word or instruction

Ladder Diagram: WOR (S1) (S2) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: WOR *(S1) (S2) (D)*	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

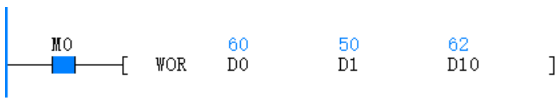
S2: Source operand 2

D: Destination operand

- **Function Description**

When the power flow is valid, S1 and S2 are logically ORed, and the result is assigned to D.

- **Example of use**



LD M0

WOR D0

D1 D10

When X0=ON, D0=2#0000000000111100(60) and D1=2#0000000000110010(50) bit logical OR, the result is assigned to D10, D10=2#0000000000111110(62).

6.6.3 WXOR: Word XOR Operation

Ladder Diagram: ├──┤ ├──[WXOR (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: WXOR (S1) (S2) (D)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

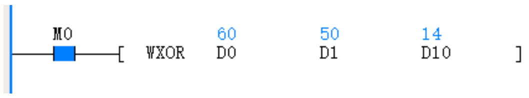
S2: Source operand 2

D: Destination operand

- **Function Description**

When the power flow is valid, S1 and S2 are logically XORed by bit, and the result is assigned to D.

- **Example of use**



LD M0
WXOR D0 D1 D10

When M0=ON, D0=2#0000000000111100(60) and D1=2#0000000000110010(50) bit logical XOR, the result is assigned to D10, D10=2#0000000000001110(14).

6.6.4 WINV: Word inversion operation

Ladder Diagram: ├──┤ ├──[WINV (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: WINV (S) (D)	Step size	5

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- **Operand Description**

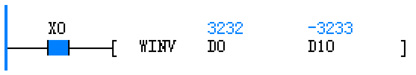
S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, the bitwise logic of S is negated, and the result is assigned to D.

- **Example of use**



LD X0
WINV D0 D10

When X0=ON, invert D0=(3232) by bit logic, and assign the result to D10, D10=(-3233).

6.6.5 DWAND: Double Word and Instruction

Ladder Diagram: ├──┤ ├──[DWAND (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: DWAND (S1) (S1) (D)	Step size	10

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

● **Operand Description**

S1: Source operand 1

S2: Source operand 2

D: Destination operand

● **Function Description**

When the energy flow is valid, S1 and S2 are bitwise logically AND, and the result is assigned to D.

● **Example of use**

LD X0

DWAND D0 D2 D10

When X0=ON, (D0, D1)=2#100101101011010000111 (1234567) and (D2, D3)=2#100101101011010000111110 (9876542) bit logical AND, the result is assigned to (D10, D11), (D10, D11)= 2#100101001010000000110 (1217542).

6.6.6 DWOR: Double word or instruction

Ladder Diagram: ├──┤ ├──[DWOR (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: DWOR *(S1) (S2) (D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

● **Operand Description**

S1: Source operand 1

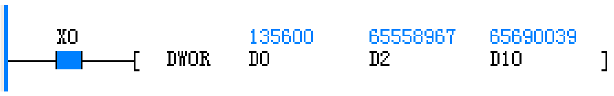
S2: Source operand 2

D: Destination operand

● **Function Description**

When the power flow is valid, S1 and S2 are logically ORed, and the result is assigned to D.

● **Example of use**



LD X0

DWORD D0 D2 D10

When X0=ON, (D0, D1)=2#100001000110110000 (135600) and (D2, D3)=2#11111010000101100110110111 (65558967) bit logical OR, the result is assigned to (D10, D11), (D10, D11)= 2#11111010100101100110110111 (65690039).

6.6.7 DWXOR: Double Word XOR Instruction

Ladder Diagram: ├──┤ ├──[DWXOR (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: DWXOR *(S1) (S2) (D)*	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

● **Operand Description**

S1: Source operand 1

S2: Source operand 2

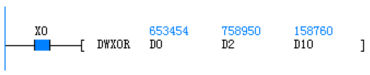
D: Destination operand

● **Function Description**

When the energy flow is valid, S1 and S2 are bitwise logically XOR,

The result is assigned to D.

● **Example of use**



LD X0

DWXOR D0 D2 D10

When X0=ON, (D0, D1)=2#10011111100010001110 (653454) and (D2,D3)=2#10111001010010100110 (758950) bitwise logical XOR, the result is assigned to (D10, D11), (D10, D11)) = 2#100110110000101000 (158760).

6.6.8 DWINV: Double word negation instruction

Ladder Diagram: ├─┤ ├──[DWINV (S) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: DWINV (S) (D)	Step size	7

Operand	Type	Applicable devices													Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V	R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V	R	√

- **Operand Description**

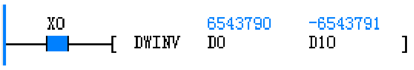
S: Source operand

D: Destination operand

- **Function Description**

When the power flow is valid, negate the bitwise logic of S, and assign the result to D

- **Example of use**



LD X0

DWINV D0 D10

When X0=ON, (D0, D1)=2#11000111101100110101110 (6543790) is inverse by bit logic, and the result is assigned to (D10, D11), (D10, D11) =2#1111111111111111111111111111111111111111100111000010011001010001 (-6543791).

6.7 Bit Shift Rotation Instruction

6.7.1 ROR: 16-bit rotate right instruction

Ladder Diagram: ├─┤ ├──[ROR (S1) (D) (S2)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Carry flag SM81
Instruction list: ROR (S1) (D) (S2)	Step size	7

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

D: Destination operand

S2: Source operand 2

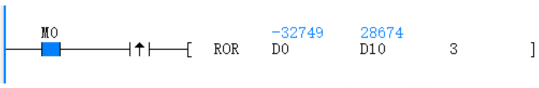
- **Function Description**

When the power flow is valid, the data of S1 is cyclically shifted to the right by S2 bits and assigned to D. At the same time, the last bit of the shift is stored in the Carry flag (SM81).

- **Precautions**

S2 range is greater than or equal to 0; when S1 is Kn

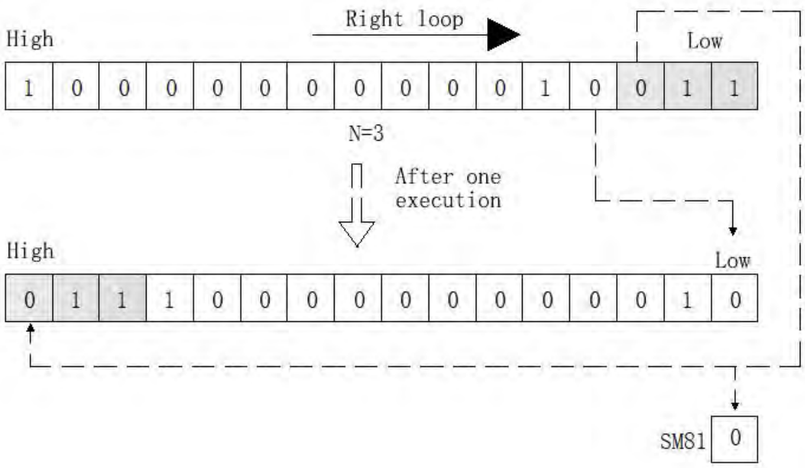
- **Example of use**



LD M0

EU

ROR D0 D10 3



addressed, Kn must be equal to 4.

When M0=ON, D0=2#1000000000010011 (-32749) cyclically shifts 3 bits to the right, the result is assigned to D10, the final bit of the shift is stored in the Carry flag, D10=2#0111 0000 0000 0010 (28674), SM81= OFF

6.7.2 ROL: 16-bit rotate left instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[ROL (S1) (D) (S2)]	Affect the flag	Carry flag SM81
Instruction list: ROL (S1) (D) (S2)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***D*:** Destination operand

***S2*:** Source operand 2

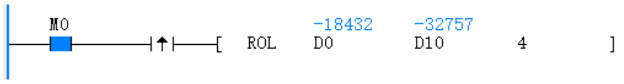
- **Function Description**

When the power flow is valid, the data of S1 is cyclically shifted to the left by S2 bits and assigned to D. At the same time, the last bit of the shift is stored in the Carry flag (SM81).

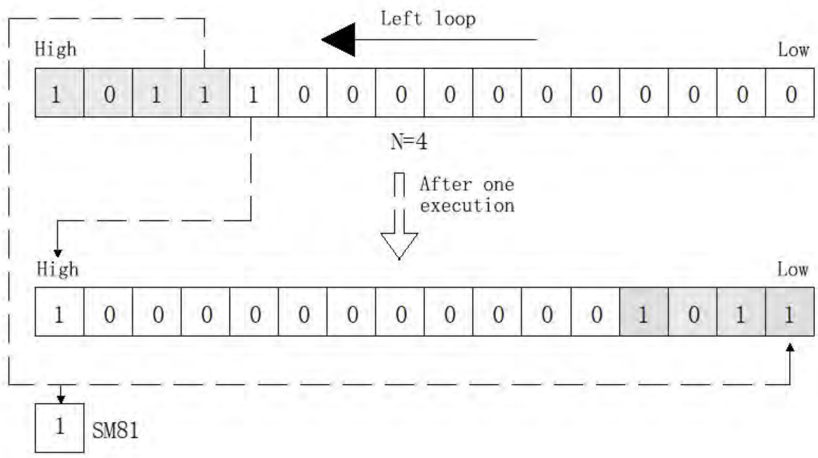
- **Precautions**

S2 range is greater than or equal to 0; when S1 is Kn addressed, Kn must be equal to 4.

- **Example of use**



LD M0
EU
ROL D0 D10 4



When X0=ON, D0=2#1011100000000000(-18432) cyclically shifts 4 bits to the left, the result is assigned to D10, the final bit of the shift is stored in the Carry flag, D10=2#1000000000001011 (-32757), SM81=OFF.

6.7.3 RCR: 16-bit rotate right instruction with carry

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[RCR (S1) (D) (S2)]	Affect the flag	Carry flag SM81
Instruction list: RCR (S1) (D) (S2)	Step size	7

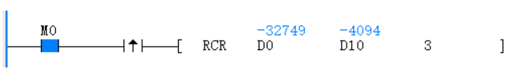
Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***D*:** Destination operand

- **Example of use**



LD M0
EU

S2: Source operand 2

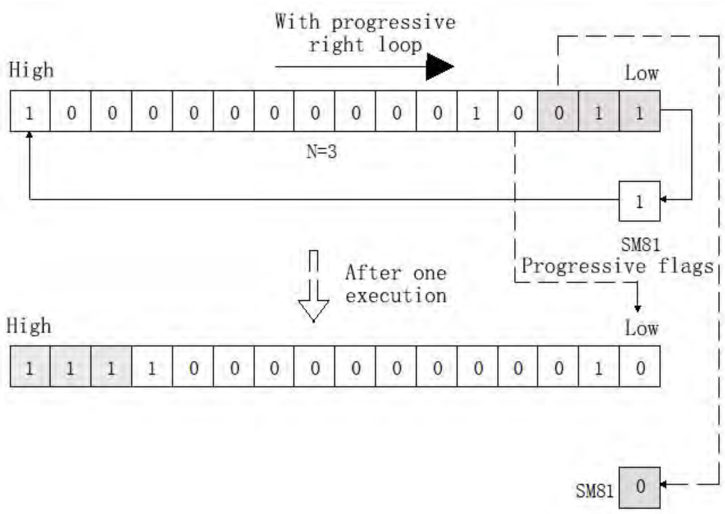
- **Function Description**

When the power flow is valid, the data of S1 is shifted to the right with the carry bit (SM81) and the result after the S2 bit is shifted to the right is assigned to D.

- **Precautions**

S2 range is greater than or equal to 0; when S1 is Kn addressed, Kn must be equal to 4.

RCR D0 D10 3



When M0=ON, D0=2#1000000000010011(-32749) with carry SM81=ON to rotate right by 3 bits, the result is assigned to D10, D10=2#1111000000000010 (-4094), SM81=OFF.

6.7.4 RCL: 16-bit rotate left instruction with carry

Ladder Diagram: RCL (S1) (D) (S2)								Applicable models		VC-B VC-S VC-P					
								Affect the flag		Carry flag SM81					
Instruction list: RCL *(S1) (D) (S2)*								**Step size**		**7**					
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

D: Destination operand

S2: Source operand 2

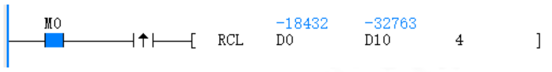
- **Function Description**

When the power flow is valid, the data of S1 is cyclically shifted to the left with the carry bit (SM81) and the result of the S2 bit is assigned to D.

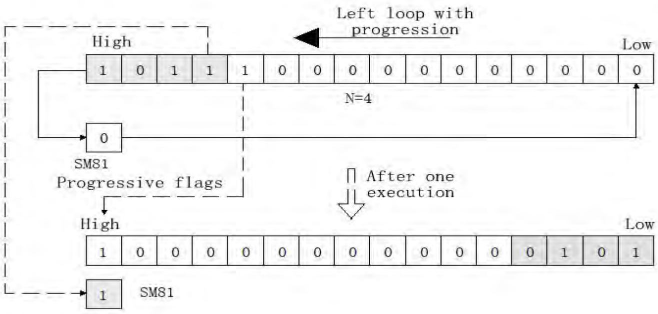
- **Precautions**

S2 range is greater than or equal to 0; when S1 is Kn addressed, Kn must be equal to 4.

- **Example of use**



LD M0
EU
RCL D0 D10 4



When M0=ON, D0=2#1011100000000000 (-18432) with carry (SM81=OFF) cyclically shifted left by 4 bits, the result is assigned to D10, D10=2#1000000000000101 (-32763), SM81=ON.

6.7.5 DROR: 32-bit rotate right instruction

Ladder Diagram: ├─┤ ├─[DROR (S1) (D) (S2)]	**Applicable models** VC-B VC-S VC-P
	Affect the flag Carry flag SM81
Instruction list: DROR (S1) (D) (S2)	**Step size** 9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

D: Destination operand

S2: Source operand 2

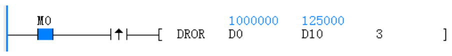
- **Function Description**

When the power flow is valid, the data of S1 is cyclically shifted to the right by S2 bits and assigned to D. At the same time, the last bit of the shift is stored in the Carry flag (SM81).

- **Precautions**

S2Range greater than or equal to 0;whenS1forWhen addressing Kn, Kn must be equal to 8.

- **Example of use**



LD M0
EU
DROR D0 D10 3

When M0=ON, D0 (D1)=2#11110100001001000000 (1000000) cyclically shifts right by 3 bits, the result is assigned to (D10, D11), and the final bit of the shift is stored in the Carry flag bit, (D10, D11)=2#1 1110100001001000 (125000), SM81=OFF.

Please refer to the ROR instruction legend

6.7.6 DROL: 32-bit rotate left instruction

Ladder Diagram: ├─┤ ├─[DROL (S1) (D) (S2)]	**Applicable models** VC-B VC-S VC-P
	Affect the flag Carry flag SM81
Instruction list: DROL (S1) (D) (S2)	**Step size** 9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

D: Destination operand

S2: Source operand 2

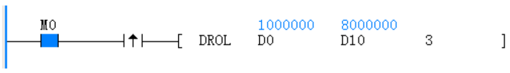
- **Function Description**

When the power flow is valid, the data of S1 is cyclically shifted to the left by S2 bits and assigned to D. At the same time, the last bit of the shift is stored in the Carry flag (SM81).

- **Precautions**

S2 range is greater than or equal to 0; when S1 is Kn addressed, Kn must be equal to 4.

- **Example of use**



LD M0
EU
DROL D0 D10 3

When M0=ON, (D0, D1)=2#11110100001001000000 (1000000) cyclically shifts right by 3 bits, the result is assigned to (D10, D11), and the final bit of the shift is stored in the Carry flag, (D10, D11)=2 #1111010 0001001000000000 (8000000), SM81=OFF.

Please refer to the ROL instruction legend

6.7.7 DRCR: 32-bit rotate right instruction with carry

Ladder Diagram: ├─┤ ├──[DRCR (S1) (D) (S2)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Carry flag SM81
Instruction List: DRCR (S1) (D) (S2)	Step size	9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***D*:** Destination operand

***S2*:** Source operand 2

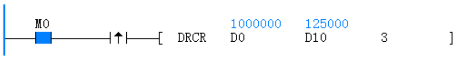
- **Function Description**

When the power flow is valid, the data of S1 is shifted to the right with the carry bit (SM81) and the result after the S2 bit is shifted to the right is assigned to D.

- **Precautions**

S2 range is greater than or equal to 0; when S1 is Kn addressed, Kn must be equal to 4.

- **Example of use**



LD M0
EU
DRCR D0 D10 3

1. When M0=ON, (D0,D1)=2#11110100001001000000 (1000000) with carry (SM81=OFF) cyclically shift right by 3 bits, the result is assigned to (D10,D11), (D10,D11)=2#1 1110100001001000 (125000), SM81=OFF.

2. Please refer to the RCR instruction legend

6.7.8 DRCL: 32-bit rotate left instruction with carry

Ladder Diagram: ├─┤ ├──[DRCL (S1) (D) (S2)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Carry flag SM81
Instruction List: DRCL (S1) (D) (S2)	Step size	9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***D*:** Destination operand

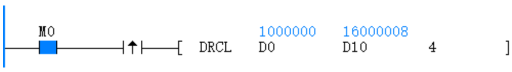
***S2*:** Source operand 2

- **Function Description**

When the power flow is valid, the data of S1 is cyclically shifted to the left with the carry bit (SM81) and the result of the S2 bit is assigned to D.

- **Precautions**

- **Example of use**



LD M0
EU
DRCL D0 D10 4

1. When M0=ON, (D0, D1)=2#11110100001001000000 (1000000) with carry (SM81=ON) cyclically shift left by 4 bits, the result is assigned to (D10,D11), (D10,D11)=2# 111101000010010 00001000 (16000008), SM81=OFF.

2. Please refer to the RCL instruction legend

S2 range is greater than or equal to 0; when S1 is Kn addressed, Kn must be equal to 4.

6.7.9 SHR: 16-bit right shift instruction

Ladder Diagram: ─┤ ├─[SHR (S1) (D) (S2)]								Applicable models		VC-B VC-S VC-P					
								Affect the flag							
Instruction list: SHR (S1) (D) (S2)								Step size		7					
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

D: Destination operand

S2: Source operand 2

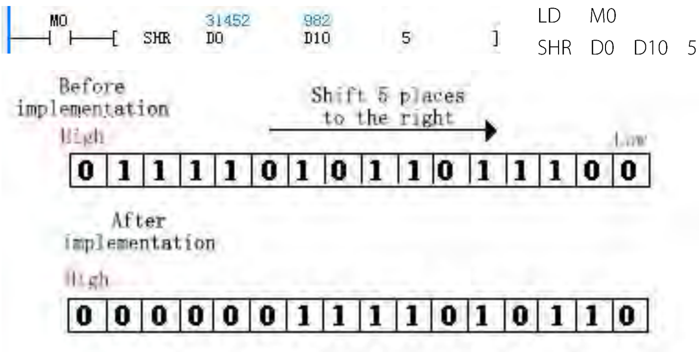
- **Function Description**

When the power flow is valid, the result of shifting the data of S1 to the right by S2 bits is assigned to D.

- **Precautions**

S2 The range is greater than or equal to 0; when S1 addresses Kn, Kn must be equal to 4.

- **Example of use**



LD M0
SHR D0 D10 5

When M0=ON, D0=2#0111101011011100 (31452) is shifted to the right by 5 bits, and the result is assigned to D10, D10=2#0000001111010110 (982).

6.7.10 SHL: 16-bit left shift instruction

Ladder Diagram: ─┤ ├─[SHL (S1) (D) (S2)]								Applicable models		VC-B VC-S VC-P					
								Affect the flag							
List of instructions: SHL (S1) (D) (S2)								Step size		7					
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

- **Example of use**

LD M0
SHL D0 D10 7

***D*:** Destination operand

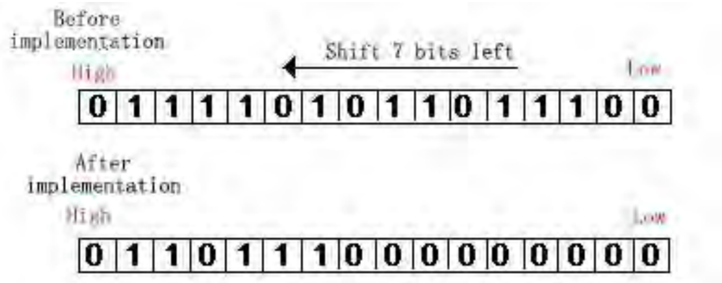
***S2*:** Source operand 2

- **Function Description**

When the power flow is valid, the result of shifting the data of S1 to the left by S2 bits is assigned to D.

- **Precautions**

S2 The range is greater than or equal to 0; when S1 addresses Kn, Kn must be equal to 4.



When M0=ON, D0=2#0111101011011100 (31452) is shifted left by 7 bits, and the result is assigned to D10, D10=2#0110111000000000 (28160).

6.7.11 DSHR: 32-bit shift right instruction

Ladder Diagram: DSHR *(S1)* *(D)* *(S2)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: DSHR *(S1)* *(D)* *(S2)*	Step size	9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***D*:** Destination operand

***S2*:** Source operand 2

- **Function Description**

When the power flow is valid, the result of shifting the data of S1 to the right by S2 bits is assigned to D.

- **Precautions**

S2 The range is greater than or equal to 0; when S1 addresses Kn, Kn must be equal to 8.

- **Example of use**

LD M0
DSHR D0 D10 10

1. When M0=ON, (D0,D1)=2#01110011100110001001110010101100 (1939381420) is shifted right by 10 bits, and the result is assigned to (D10,D11), (D10,D11)=2#00000000000111001110011000100111 (18).

2. Please refer to the SHR instruction legend

6.7.12 DSHL: 32-bit shift left instruction

Ladder Diagram: DSHL *(S1)* *(D)* *(S2)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: DSHL *(S1)* *(D)* *(S2)*	Step size	9

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

- **Example of use**

LD M0
DSHL D0 D10 15

***D*:** Destination operand

***S2*:** Source operand 2

- **Function Description**

When the power flow is valid, the result of shifting the data of S1 to the left by S2 bits is assigned to D.

- **Precautions**

S2 The range is greater than or equal to 0; when S1 addresses Kn, Kn must be equal to 8.

1. When M0=ON, (D0, D1)=2#01110011100110001001110010101100 (1939381420) is shifted to the left by 15 bits, and the result is assigned to (D10,D11),(D10,D11)=2#01001110010101100000000000000000000000000000 (134).

2. Please refer to the SHL instruction legend

6.7.13 SFTR: Bit string right shift instruction

Ladder Diagram: ├──┤ ├──[SFTR *(S1)* *(D)* *(S2)* *(S3)*	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: SFTR *(S1)* *(D)* *(S2)* (S3)	Step size	9

Operand	Type	Applicable devices														Index
S1	BOOL		X	Y	M	S	LM	SM			C	T				√
D	BOOL			Y	M	S	LM				C	T				√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

***S1*:** Source operand 1

***D*:** Destination operand

***S2*:** Source operand 2

S3: Source operand 3

- **Function Description**

When the power flow is valid, the contents of the S2 units starting from the D unit are shifted to the right by S3 units, and the S3 data at the rightmost end will be discarded. At the same time, the contents of S3 units starting with S1 unit will be shifted to the left end of the string.

- **Precautions**

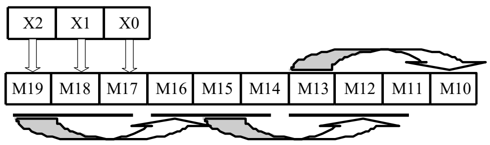
In left and right order, the large component number is on the left, and the small component number is on the right.

S2 The range is greater than or equal to zero, and the S3 range is greater than or equal to zero.

- **Example of use**

LD M0

SFTR X0 M10 10 3



1. When M0=ON, the contents of the 10 units starting from the M10 unit are shifted to the right by 3 units in units of bits, and the rightmost M10~M12 will be discarded. At the same time, the contents of the 3 cells starting at cell X0 are shifted into the left end of the bit string.

2. Before execution: X0=1, X1=0, X2=1. M10=0, M11=1, M12=1, M13=0, M14=0, M15=1, M16=0, M17=0, M18=0, M19=1.

3. After execution: the contents of X0~X2 remain unchanged. M10=0, M11=0, M12=1, M13=0, M14=0, M15=0, M16=1, M17=1, M18=0, M19=1.

6.7.14 SFTL: Bit string left shift instruction

Ladder Diagram: ├──┤ ├──[SFTL *(S1)* *(D)* *(S2)* *(S3)*	Applicable models	VC-B VC-S VC-P
	Affect the flag	

Instruction list: SFTL *(S1)* *(D)* *(S2)* (S3)									Step size			9				
Operand	Type	Applicable devices														Index
S1	BOOL		X	Y	M	S	LM	SM			C	T				√
D	BOOL			Y	M	S	LM				C	T				√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand 1

D: Destination operand

S2: Source operand 2

S3: Source operand 3

- **Precautions**

1. In left and right order, the large component number is on the left, and the small component number is on the right.

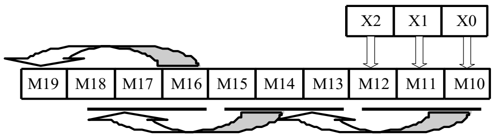
2. The S2 range is greater than or equal to zero, and the S3 range is greater than or equal to zero.

- **Function Description**

When the power flow is valid, move the contents of the S2 units starting from the D unit to the left by S3 units, the leftmost S3 data will be discarded, and at the same time, the contents of the S3 units starting with the S1 unit will be moved to the right end of the string .

- **Example of use**

LD M0
SFTL X0 M10 10 3



1. When M0=ON, the contents of the 10 units starting from the M10 unit are shifted to the left by 3 units in units of bits, and the leftmost M17~M19 will be discarded. At the same time, the contents of 3 cells starting at cell X0 are shifted into the right end of the bit string.

2. Before execution: X0=1, X1=0, X2=1. M10=0, M11=1, M12=1, M13=0, M14=0, M15=1, M16=0, M17=0, M18=0, M19=1.

3. After execution: the contents of X0~X2 remain unchanged. M10=1, M11=0, M12=1, M13=0, M14=1, M15=1, M16=0, M17=0, M18=1, M19=0.

6.8 Peripheral Instructions

6.8.1 REFF: Set input filter Constant command

Ladder Diagram: ┤ ├─[REFF (D) (S)]															**Applicable models**	**VC-B VC-S VC-P**
															Affect the flag	
Instruction List: REFF *(D)* *(S)*															**Step size**	**3**
Operand	Type	Applicable devices														Index
D	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S	BOOL		X													

- **Operand Description**

S: Input filter Constant VC-B VC-S VC-P: Setting range: 0us~64ms, the data larger than 64 is processed as 64.

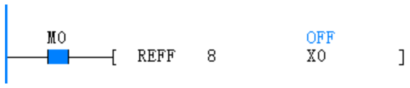
- **Function Description**

Set the input filter constants of X0~X7.

- **Precautions**

The input filter Constant is only valid for the port used as normal input, not valid for the port used as high-speed input.

- **Example of use**



LD M0
REFF 8 X0

When M0 is ON, the filter Constant time for changing the input is 8ms.

6.8.2 REF: I/O immediate refresh command

Ladder Diagram: ┤ ├─[REF *(D)* *(S)*															**Applicable models**	**VC-B VC-S VC-P**
															Affect the flag	
Instruction List: REF *(D)* *(S)*															**Step size**	**5**
Operand	Type	Applicable devices														Index
D	BOOL		X	Y												
S	INT	Constant														

- **Operand Description**

D: Start X/Y device to be refreshed

Specify the start device number to be an integer multiple of 8. Such as X0, X10, X20... or Y0, Y10, Y20..., the lowest bit is 0.

S: Number of ports to flush

The refresh points should be 8, 16, ..., 256 (multiples of 8, other values are wrong)

- **Function Description**

Usually, the input and output of the PLC are executed after the end of the user program. During operation, if you need to read the latest input state or want to update the output state immediately, you can use this instruction.

- **Precautions**

1. The number of subscripts to the input ports (Xn, Yn) should be an integer multiple of 8.

2. The number of (ports) refreshed should also be an integer multiple of 8.

3. Between FOR-NEXT instructions or between CJ instructions, REF is generally used for immediate processing.

4. When the interrupt processing with I/O action is executed, the I/O refresh is performed in the interrupt subroutine, the latest input information is acquired and the operation result is output in time, and the REF instruction is used.

5. For relay-Type output points, the response time of the output points should be considered.

- **Example of use**

LD M0
REF Y0 8

When M0 is ON, the statuses of Y0 to Y7 are output immediately and are not affected by the scan cycle.

6.9 Real Time Clock Instruction

6.9.1 TRD: Real Time Clock Read Command

Ladder Diagram: ├──┤ ├──[TRD (D)]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction List: TRD (D)	**Step size**	**3**

Operand	Type	Applicable devices														Index
D	INT								D				V		R	√

- **Operand Description**

D: Read out the starting unit stored in the system time, occupying 7 consecutive units starting from the unit designated by D

- **Function Description**

Read the time in the system and save it in the storage unit designated by D.

- **Precautions**

When a clock setting error occurs in the system, the TRD read time is unsuccessful.

- **Example of use**

LD M0
TRD D10

When M0 is ON, the system time is sent to the 7 units starting from D10. The execution result of the instruction is as follows:

	element	project	clock data		element	project
Special data register for real-time clock	SD60	year	2000～2099	————————→	D10	year
	SD61	moon	1 to 12	————————→	D11	moon
	SD62	day	1 to 31	————————→	D12	day
	SD63	Time	0～23	————————→	D13	Time
	SD64	Minute	0～59	————————→	D14	Minute
	SD65	second	0～59	————————→	D15	second
	SD66	Week	0～6	————————→	D16	Week

6.9.2 TWR: Real Time Clock Write Command

Ladder Diagram: ├──┤ ├──[TWR (S)]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction List: TWR (S)	**Step size**	**3**

Operand	Type	Applicable devices														Index
S	INT								D				V		R	√

- **Operand Description**

- **Example of use**

Change the time of the system through TWR, see the figure below:

S: The device to which the system time is written

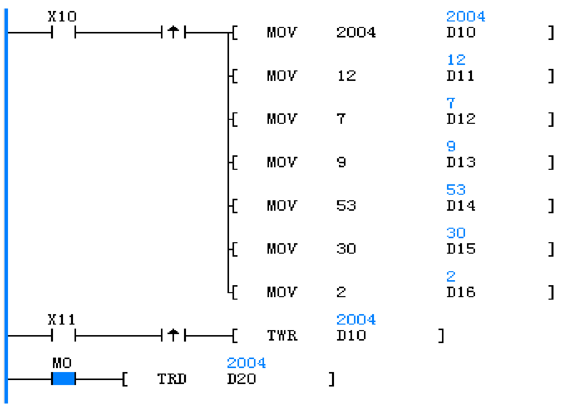
- **Function Description**

When the system time is different from the actual time, the TWR instruction can be used to change the system time.

- **Precautions**

1. The written time data must meet the requirements of the Gregorian calendar, otherwise the command execution will fail.

2. It is recommended to use edge triggering as the instruction execution condition.



LD X10
EU
MOV 2004 D10
MOV 12 D11
MOV 7 D12
MOV 9 D13
MOV 53 D14
MOV 30 D15
MOV 2 D16
LD X11
EU
TWR D10
LD M0
TRD D20

1. When the rising edge of X10 is detected, the time setting value is written to 7 consecutive units of D10.

2. When the rising edge of X11 is detected, the value of 7 consecutive units of D10 is written into the system time.

3. When M0 is ON, read the system time and store it in D20.

	Element	Project	Clock data		Element	Project
Data for clock setting	D10	Year	2000～2099	——————→	SD60	Year
	D11	Moon	1 to 12	——————→	SD61	Moon
	D12	Day	1 to 31	——————→	SD62	Day
	D13	Time	0～23	——————→	SD63	Time
	D14	Minute	0～59	——————→	SD64	Minute
	D15	Second	0～59	——————→	SD65	Second
	D16	Week	0～6	——————→	SD66	Week

6.9.3 TADD: Clock plus instruction

Ladder Diagram: ├─┤ ├─[TADD *(S1)* *(S2)* *(D)*]															Applicable models	VC-B VC-S VC-P
															Affect the flag	**Zero flag SM80 Carry flag SM81**
Instruction list: TADD *(S1)* *(S2)* *(D)*															**Step size**	**7**
Operand	Type	Applicable devices														Index
S1	INT							D	SD			V		R		√
S2	INT							D	SD			V		R		√
D	INT							D				V		R		√

- **Operand Description**

S1: Clock data 1

The time data is stored in the three storage units indicated by S1. For data that does not meet the time

- **Example of use**

S1			S2			D	
D10	23 hours	+	D20	23 hours	=	D30	23 hours
D11	59 points		D21	58 points		D31	58 points
D12	59 seconds		D22	58 seconds		D32	57 seconds

format, the system prompts an illegal error of the instruction operand value.

S2: Clock data 2

Another time data is stored in the three storage units indicated by S2. For data that does not meet the time format, the system prompts an illegal error of the instruction operand value.

D: Time result storage unit

The data processed by time plus is stored in the 3 storage units pointed to by D. The Carry flag SM81 and the Zero flag SM80 will be affected according to the result of the processing.

- **Function Description**

The data in time format is added, and the operation rules are executed according to the time format.

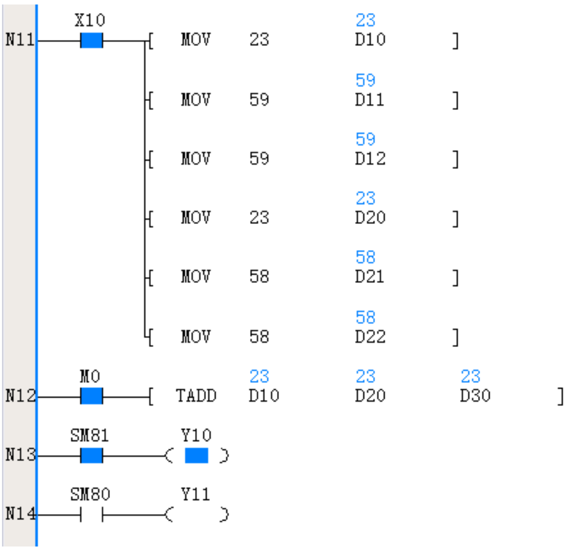
- **Precautions**

The time data involved in the operation should conform to the time format:

"Hour" setting range: 0 to 23

"Min" setting range: 0～59

"Second" setting range: 0 to 59



LD X10
MOV 23 D10
MOV 59 D11
MOV 59 D12
MOV 23 D20
MOV 58 D21
MOV 58 D22
LD M0
TADD D0 D20 D30
LD SM81
OUT Y 10
LD SM80
OUT Y 11

1. When X10 is ON, the time data is sent to 3 points starting from D10 and 3 storage units starting from D20.

2. When M0 is ON, the 3 storage units starting from D10 are added to the 3 storage units starting from D20, and the processed result is stored in the 3 storage units starting from D30.

3. The Carry flag (SM81) is ON, and the Zero flag (SM80) is OFF.

6.9.4 TSUB: Clock Subtract Instruction

Ladder Diagram: ┤├──┤[TSUB (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	Zero flag SM80 Borrow flag SM82
Instruction list: TADD (S1) (S2) (D)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT								D	SD			V		R	√
S2	INT								D	SD			V		R	√
D	INT								D				V		R	√

- **Operand Description**

S1: Clock data 1

The time data is stored in the three storage units indicated by S1. For data that does not meet the time format, the system prompts an illegal error of the instruction operand value.

S2: Clock data 2

Another time data is stored in the three storage units indicated by S2. For data that does not meet the time format, the system prompts an illegal error of the instruction operand value.

D: Time result storage unit

The data processed by time plus is stored in the 3 storage units pointed to by D. The borrow flag SM82 and the Zero flag SM80 will be affected according to the result of the processing.

- **Function Description**

Subtract the data in the time format, and the operation rules are executed according to the time format.

- **Precautions**

The time data involved in the operation should conform to the time format:

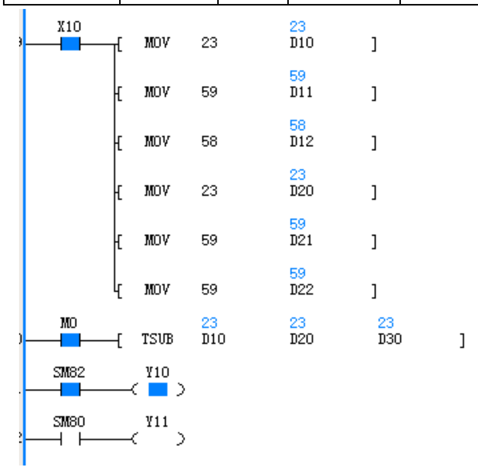
"Hour" setting range: 0 to 23

"Min" setting range: 0～59

"Second" setting range: 0 to 59

- **Example of use**

S1			S2			D	
D10	23 hours		D20	23 hours		D30	23 hours
D11	59 points	-	D21	59 points	=	D31	59 points
D12	58 seconds		D22	59 seconds		D32	59 seconds




```
LD   X10
MOV 23 D10
MOV 59 D11
MOV 58 D12
MOV 23 D20
MOV 59 D21
MOV 59 D22
LD   M0
TSUB D10 D20 D30
LD   SM82
OUT  Y 10
LD   SM80
OUT  Y11
```

1. When X10 is ON, the time data is sent to 3 points starting from D10 and 3 storage units starting from D20.

2. When M0 is ON, the 3 storage units starting from D10 are subtracted from the 3 storage units starting from D20, and the processed result is stored in the 3 storage units starting from D30.

3. The borrow flag (SM82) is ON, and the Zero flag (SM80) is OFF.

6.9.5 HOUR:Chronograph command

Ladder Diagram: ├──┤ ├──[HOUR (S) (D1) (D2)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: HOUR (S) (D1) (D2)	Step size	8

Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	INT								D				V		R	√
D2	BOOL			Y	M	S	LM									

- **Operand Description**

S: Hourly comparison data. Data range 0～32767

D1: Time storage unit

D1The data unit of D1+1 is kept for hours, and the data unit of D1+1 is kept for seconds

D2: Alarm output address

When the data of D1 is greater than or equal to the data specified by S, the alarm point becomes ON output.

- **Function Description**

The time that the input contact is in the ON state is judged in units of hours.

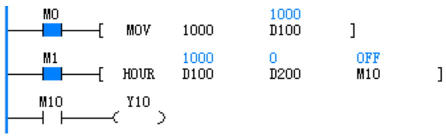
- **Precautions**

1. In order to use the current data even after the power of the PLC is cut off, please designate D1 as the device unit for power failure retention. If ordinary device units are used, the current data will be cleared when the power of the PLC is cut off or the operation is performed from RUN to STOP.

2. Even if the alarm output D2 is ON, it can continue to count.

3. The instruction hour is 16-bit integer data. When the data of the hour is greater than 32767, it starts from 0 again.

- **Example of use**




```
LD    M0
MOV   1000
D100
LD    M1
HOUR  D100
D200  M10
LD    M10
OUT   Y10
```

1. Set the comparison data of the HOUR instruction when M0 is ON.

2. When M1 is ON, HOUR adds time to the input contact.

3. When the accumulated time of ON state of M1 is greater than or equal to 1000, M10 is ON state.

6.9.6 DCMP: (=, <, >, <>, >=, <=) date comparison commands

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├──┤ ├──[DCMP= (S1) (S2) (D)] ├──┤ ├──[DCMP< (S1) (S2) (D)] ├──┤ ├──[DCMP> (S1) (S2) (D)] ├──┤ ├──[DCMP<> (S1) (S2) (D)] ├──┤ ├──[DCMP>= (S1) (S2) (D)] ├──┤ ├──[DCMP<= (S1) (S2) (D)]	Affect the flag	

Command list:	Step size	7
DCMP= (*S1*) (*S2*) (*D*) **DCMP<** (*S1*) (*S2*) (*D*) **DCMP>** (*S1*) (*S2*) (*D*) **DCMP<>** (*S1*) (*S2*) (*D*) **DCMP>=** (*S1*) (*S2*) (*D*) **DCMP<=** (*S1*) (*S2*) (*D*)		

Operand	Type	Applicable devices														Index
S1	INT								D	SD			V		R	√
S2	INT								D	SD			V		R	√
D	BOOL			Y	M	S	LM				C	T				

- **Operand Description**

S1: Date comparison data 1, occupying 3 word units starting from the specified unit of S1, the data of the 3 units must conform to the Gregorian calendar format, otherwise the system will report an operand error.

S2: Date comparison data 2, occupy the first 3 word units of the specified unit of S2, and the data of the 3 units must conform to the Gregorian calendar format, otherwise the system will report an operand error.

D: The comparison status output, the data meets the comparison conditions, D is set to ON, otherwise it is OFF.

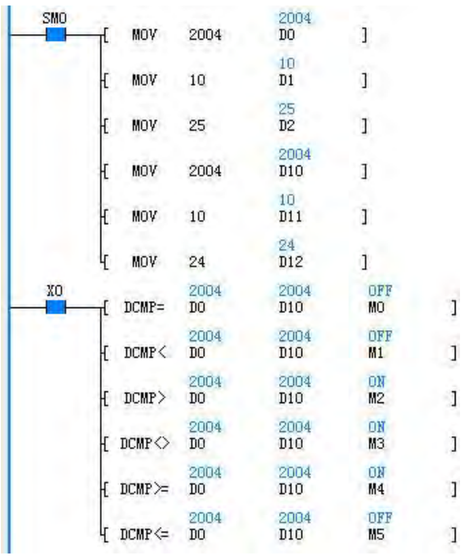
- **Function Description**

BIN comparison is performed on the date data starting with S1 and S2 respectively, and D is assigned to the result of the comparison.

- **Precautions**

The date data with S1 and S2 as the starting unit must comply with the Gregorian calendar, otherwise an operand error will be reported (for example: 2004, 9, 31 and 2003, 2, 29 and other data are not legal).

- **Example of use**



LD SM0
MOV 2004 D0
MOV 10 D1
MOV 25 D2
MOV 2004 D10
MOV 10 D11
MOV 24 D12
LD X0
DCMP= D0 D10 M0
DCMP< D0 D10 M1
DCMP> D0 D10 M2
DCMP<> D0 D10 M3
DCMP>= D0 D10 M4
DCMP<= D0 D10 M5

BIN comparison is performed on the date data starting with D0 and D10 respectively, and the result of the comparison is assigned to the target data (M0, etc.).

6.9.7 TCMP: (=, <, >, <>, >=, <=) time comparison instructions

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
[TCMP= (S1) (S2) (D)] [TCMP< (S1) (S2) (D)] [TCMP> (S1) (S2) (D)] [TCMP<> (S1) (S2) (D)] [TCMP>= (S1) (S2) (D)] [TCMP<= (S1) (S2) (D)]	Affect the flag	

Command list: DCMP= (S1) (S2) (D) DCMP< (S1) (S2) (D) DCMP> (S1) (S2) (D) DCMP<> (S1) (S2) (D) DCMP>= (S1) (S2) (D) DCMP<= (S1) (S2) (D)	**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	INT								D	SD			V		R	√
S2	INT								D	SD			V		R	√
D	BOOL			Y	M	S	LM				C	T				

- **Operand Description**

S1: Time comparison data 1

Occupies the first 3 word units of the specified unit of S1, and the data of the 3 units must conform to the 24-hour time format, otherwise the system will report an operand error.

S2: Time comparison data 2

Occupies the first 3 word units of the specified unit of S2, and the data of the 3 units must conform to the 24-hour time format, otherwise the system will report an operand error.

D: Compare status output, if the data meets the comparison condition, D is set to ON, otherwise it is OFF

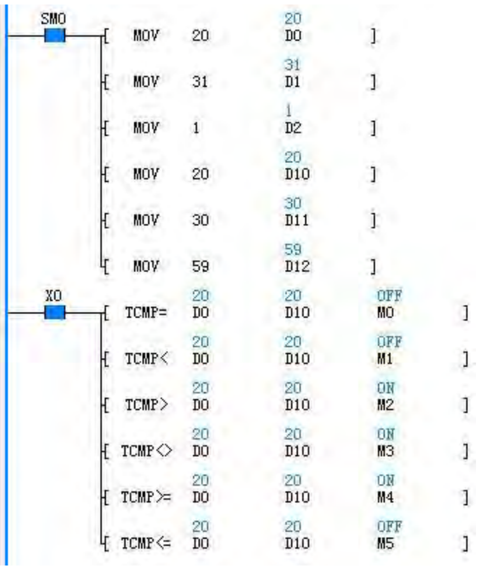
- **Function Description**

BIN comparison is performed on the time data starting with S1 and S2 respectively, and D is assigned to the result of the comparison.

- **Precautions**

The time data with S1 and S2 as the starting unit must conform to the 24-hour clock, otherwise an operand error will be reported (for example: 24, 10, 31 and 13, 59, 60 and other data are illegal)

- **Example of use**



LD SM0
MOV 20 D0
MOV 31 D1
MOV 1 D2
MOV 20 D10
MOV 30 D11
MOV 59 D12
LD X0
TCMP= D0 D10 M0
TCMP< D0 D10 M1
TCMP> D0 D10 M2
TCMP<> D0 D10 M3
TCMP>= D0 D10 M4
TCMP<= D0 D10 M5

BIN comparison is performed on the time data starting with D0 and D10 respectively, and the result of the comparison is assigned to the target data (M0, etc.).

6.9.8 HTOS: Hour, minute, second data second conversion command

Ladder Diagram: [HTOS (S) (D)]	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: HTOS (S) (D)	**Step size**	**5**

Operand	Type	Applicable devices												Index
S	INT		KnX	KnY	KnM	KnS	T	C	D	SD				R	√
D	INT			KnY	KnM	KnS	T	C	D	SD				R	√

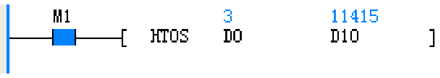
- **Operand Description**
- **Example of use**

S: The start number of the device where the time data before conversion is stored.

D: Save the converted time data device number.

- **Function Description**

After converting the time data (hour, minute, second) of S-S+2 into seconds, save the result in D.



LD M1
HTOS D0 D10

1. When M1=ON, convert the time data of hour, minute and second at the beginning of unit D0 into seconds, and save the result in D10. When D0=3, D1=10, D2=15, D10=11415.

6.9.9 STOH: Hour, minute, second conversion command for second data

Ladder Diagram: ├──┤ ├──[STOH (S) (D)]															**Applicable models**	**VC-S VC-P**
															Affect the flag	
List of instructions: STOH *(S) (D)*															**Step size**	**5**
Operand	Type	Applicable devices												Index		
S	INT		KnX	KnY	KnM	KnS	T	C	D	SD				R	√	
D	INT			KnY	KnM	KnS	T	C	D	SD				R	√	

- **Operand Description**

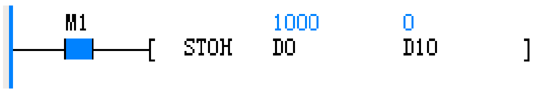
S: The device number where the time data before conversion is stored.

D: The start number of the device where the converted time data is stored.

- **Function Description**

Convert the second data of S into hours, minutes, and seconds, and save the results in D, D+1, and D+2.

- **Example of use**



LD M1
STOH D0 D10

1. When M1=ON, the second data in D0 is converted into hours, minutes and seconds, and the results are stored in 3 units starting from D10. When D0=1000, D10=0, D11=16, D12=40

6.10 High-Speed IO Instructions

6.10.1 HCNT: High-speed counter drive command

Ladder Diagram: ├──┤ ├──[HCNT *(D)* *(S)*]																**Applicable models**	**VC-B VC-S VC-P**
																Affect the flag	
Instruction list: HCNT *(D) (S)*																**Step size**	**7**
Operand	Type	Applicable devices													Index		
D	DINT										C						
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√	

- **Operand Description**

D: Specify the counter number, the settable range: C236～C263

S: Specify the comparison Constant, which is 32-bit signed data, the data range -2147483648～2147483647

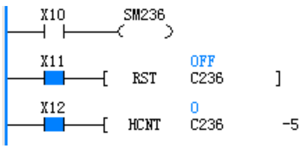
- **Function Description**

Drive the specified hardware high-speed counter. All high-speed counters can only perform hardware high-speed counting under the condition of continuous driving. At the same time, according to S, the action of the normally open contact of the high-speed counter is judged.

- **Precautions**

There are hardware conflicts in HCNT instruction, SPD instruction, external input interrupt and pulse capture. Pay attention to the usage conditions of all high-speed IOs in the system. Please refer to Chapter 8 High-speed Input Function Instructions for use.

- **Example of use**




```
LD    X10
OUT   SM236
LD    X11
RST   C236
LD    X12
HCNT  C236-5
```

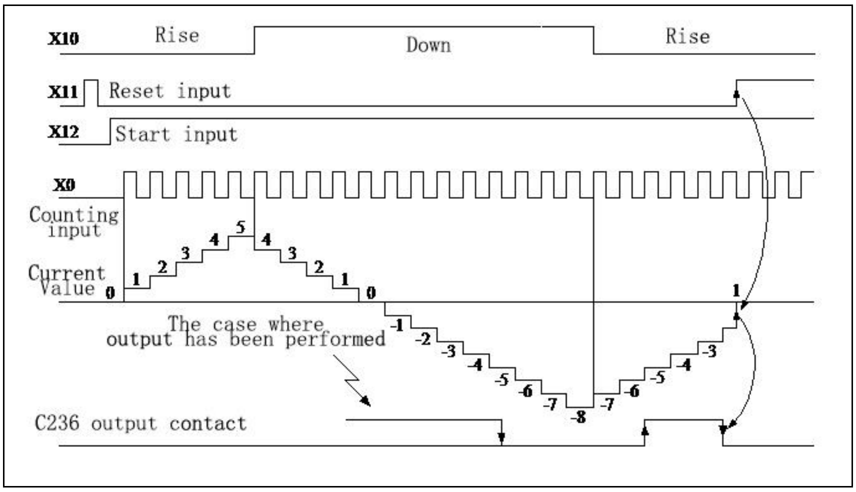
1. When X12 changes from OFF to ON, the hardware counter corresponding to C236 is initialized, X0 is the pulse input terminal of C236, and C236 counts the external pulse of X0. When X12 is OFF, X0 is a general input point, and C236 cannot count the external pulses of X0.

2. Action on the contact: When the current value of the counter C236 increases from -6 to -5, the contact of C236 is set. When the current value of the counter C236 decreases from -5 to -6, the contact of C236 is reset.

3. When X11 is ON, the RST instruction is executed, the data of C236 is cleared, and the contact of C236 is disconnected.

4. The data for the high-speed tachometer and its contact state are set by the user in the system block in the upper-level software in the event of a power failure.

The sequence operation of the operation instance of the program is as follows:



6.10.2 DHSCS: High Speed Count Compare Set Instruction

Ladder Diagram: DHSCS (S1) (S2) (D)	**Applicable models**	**VC-B VC-S VC-P**	
	Affect the flag		
Instruction List: DHSCS (S1) (S2) (D)	**Step size**	**10**	

Operand	Type	Applicable devices													Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT										C					
D	BOOL			Y	M	S										

- **Operand Description**

S1: The data to be compared by the high-speed counter is 32-bit DINT data, the data range is -2147483648～2147483647

S2: High-speed counter, the applicable range of high-speed counter is C236～C263

D: Output bit component object, set the output immediately for Y, M, S and not be affected by the scan cycle

- **Function Description**

1. The high-speed counter only counts in interrupt mode according to the count input OFF→ON under the condition of the HCNT drive instruction. When the value of the high-speed counter is equal to S1 in the DHSCS instruction, the bit element specified by D is immediately set. If it is a Y element, Y The element is output immediately.

2. This instruction can be used when it is desired to output the comparison result to the outside immediately after the comparison setting of the current value of the high-speed counter is set.

- **Precautions**

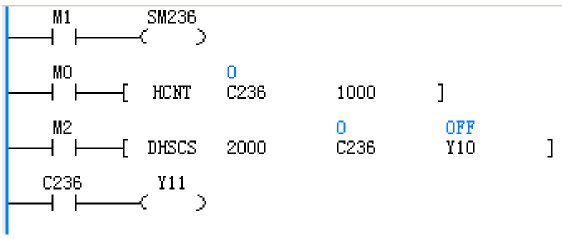
1. The DHSCS instruction action must be used in conjunction with the HCNT instruction. Only the high-speed counter driven by the HCNT can the DHSCS be used.

2. The DHSCS instruction operates on the comparison result when the pulse is input. Therefore, even if the high-speed counter value is changed with DMOV or MOV instruction, DHSCS will not operate.

3. DHSCS (DHSCI, DHSCR, DHSZ, DHSP, DHST) can be used multiple times like ordinary instructions, but the number of simultaneous driving of these instructions is limited to a total of 8 instructions or less. More than 8 valid instructions are not executed, and the valid instructions are determined according to the effective order of the instructions.

4. The maximum allowable frequency of the PLC high-speed counter. For example, when using DHSCS, DHSCI, DHSCR, DHSZ, DHSP, DHST commands, it will be limited by the maximum response frequency and integrated frequency. For details, refer to Chapter 8 High-speed Input Function User Guide.

- **Example of use**




```
LD      M1
OUT     SM236
LD      M0
HCNT    C236  1000
LD      M2
DHSCS   2000  C236  Y10
LD      C236
OUT     Y11
```

1. When M0 is ON, C236 counts X0 from OFF→ON in an interrupted way (the input frequency of X0 refers to the instruction of high-speed IO), when C236 changes from 999→1000, the C236 contact is set, and when 1001→1000, the C236 contacts Click reset. When the contact point of C236 drives Y11, the execution of Y11 is determined by the scan of the user program.

2. When M2 is ON, when the DHSCS high-speed command meets the high-speed command requirements mentioned in the precautions, Y10 is output immediately when C236 reaches 2000, and is not affected by the scan cycle.

3. When M1 is ON, SM236 is driven and the C236 counter is decremented. When M1 is OFF, SM236 is not driven, and the C236 counter counts up.

6.10.3 DHSCI: High-speed counting compare interrupt trigger instruction

Ladder Diagram: DHSCI (S1) (S2) (S3)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: DHSCI *(S1)* *(S2)* *(S3)*	Step size	10

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT										C					
S3	INT	Constant														

- **Operand Description**

S1: The data to be compared by the high-speed counter is 32-bit DINT data, the data range is -2147483648～2147483647

S2: High-speed counter, the applicable range of high-speed counter is C236～C263

S3: interrupt number. Interrupt number range: 33～40

- **Function Description**

The high-speed counter only counts in interrupt mode according to the count input OFF→ON under the condition of the HCNT drive instruction. When the value of the high-speed counter is equal to S1 in the DHSCI instruction, it enters the interrupt subroutine designated by S3. The user can write the program to be executed immediately in the interrupt subroutine.

- **Precautions**

1. The DHSCI command action must be used in conjunction with the HCNT command. Only the high-speed counter driven by the HCNT can the DHSCI be really used.

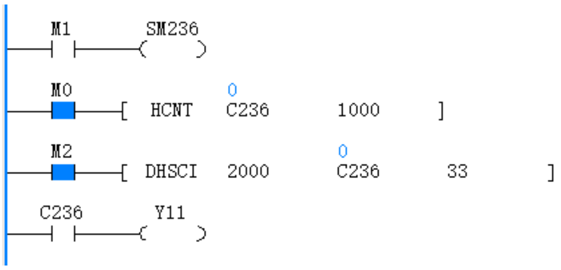
2. The DHSCI instruction acts on the comparison result when the pulse is input. Therefore, even if the high-speed counter value is changed by DMOV or MOV instruction, DHSCI will not operate.

3. DHSCI (DHSCS, DHSCR, DHSZ, DHSP, DHST) can be used multiple times like ordinary instructions, but the number of simultaneous driving of these instructions is limited to a total of 8 instructions or less. More than 8 valid instructions are not executed, and the valid instructions are determined according to the effective order of the instructions.

4. The maximum allowable frequency of the PLC high-speed counter. For example, when using DHSCS, DHSCI, DHSCR, DHSZ, DHSP, DHST commands, it will be limited by the maximum response frequency and integrated frequency. For details, refer to Chapter 8 High-speed Input Function User Guide.

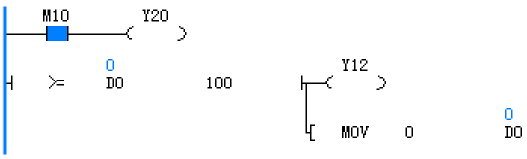
● **Example of use**

The user main program is as follows:



LD M1
OUT SM236
LD M0
DHSCI 2000 C236 33
LD C236
OUT Y11

The interrupt program with user interrupt number 33 is as follows:



LD M10
OUT Y20
LD>= D0 100
OUT Y12
MOV 0 D0

1. When M0 is ON, C236 counts X0 from OFF→ON in an interrupted way (the input frequency of X0 refers to the instruction of high-speed IO), when C236 changes from 999→1000, the C236 contact is set, and when 1001→1000, the C236 contacts Click reset. When the contact point of C236 drives Y11, the execution of Y11 is determined by the scan of the user program.

2. When M2 is ON, when the DHSCI high-speed instruction meets the high-speed instruction requirements mentioned in the precautions, when C236 to 2000, the interrupt subroutine whose interrupt number is 33 responds immediately and executes the user program in the interrupt program.

3. When M1 is ON, SM236 is driven, and the C236 counter counts down. When M1 is OFF, SM236 is not driven, and the C236 counter counts up.

4. When C236 has pulse input, when C236 is 2000, it enters the interrupt program whose interrupt number is 33. When M10 is ON, Y20 is driven, but the output execution of Y20 is related to the scan cycle of the user program. At the same time, it is also judged that when the data of D0 is greater than 100, Y12 is driven and the data of D0 is cleared.

6.10.4 DHSPI: High-speed output through position comparison interrupt trigger instruction

Ladder Diagram: DHSPI (S1) (S2) (S3)	Applicable models	VC-S VC-P
	Affect the flag	
Instruction list: DHSPI (S1) (S2) (S3)	Step size	10

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT									SD						
S3	INT	Constant														

● **Operand Description**

S1: The data to be compared by the high-speed output position element is 32-bit DINT data, the data range is -2147483648～2147483647

S2: High-speed output position element, the scope of application is the current position of the output shaft

S3: Interrupt number. The range of interrupt numbers is: 45~52

● **Function Description**

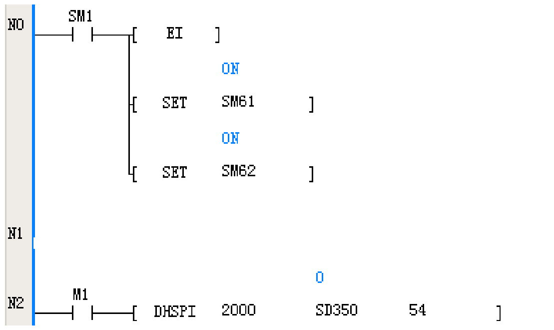
When the value of the high-speed output position element is equal to S1 in the DHSPI instruction, enter the interrupt subroutine designated by S3; the user can write the program to be executed in the interrupt subroutine.

● **Precautions**

1. When writing to an SD device, the passing position interrupt is not triggered. After writing, if the desired interrupt position is passed again, a position interrupt is triggered.

● **Example of use**

The user main program is as follows:



You can select the interrupt number of the interrupt subroutine as 54 or other high-speed output passing position interrupt source, and then write the program you want to execute when passing the position in the interrupt subroutine.

6.10.5 DHSCR: High-speed count comparison reset instruction

Ladder Diagram: DHSCR *(S1)* *(S2)* *(D)*										**Applicable models**		**VC-B VC-S VC-P**				
										Affect the flag						
Instruction list: DHSCR *(S1)* *(S2)* *(D)*										**Step size**		**10**				
Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT										C					
D	BOOL			Y	M	S					C					

● **Operand Description**

S1: The data to be compared by the high-speed counter is 32-bit DINT data, the data range is -2147483648～2147483647

S2: High-speed counter, the applicable range of high-speed counter is C236～C263

D: The output bit component object, reset the output immediately for Y, M, S, C and is not affected by the scan cycle. The C element can only be S2 itself

● **Function Description**

The high-speed counter only counts in interrupt mode according to the count input OFF→ON under the condition of the HCNT drive instruction. When the value of the high-speed counter is equal to S1 in the DHSCR instruction, the bit element specified by D is reset immediately. If it is a Y element, the Y element output immediately. Use the DHSCR high-speed comparison reset instruction when you want to output the comparison result to the outside immediately after the comparison reset of the current value of the high-speed counter.

● **Precautions**

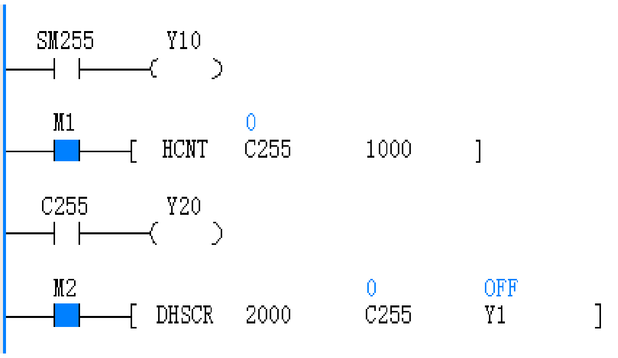
1. The DHSCR instruction action must be used in conjunction with the HCNT instruction. Only the high-speed counter driven by the HCNT can the DHSCR be really used.

2. The DHSCR instruction acts on the comparison result when the pulse is input. Therefore, even if the high-speed counter value DHSCR is changed using DMOV or MOV instruction, etc., there will be no action.

3. DHSCR (DHSCI, DHSCS, DHSZ, DHSP, DHST) can be used multiple times like ordinary instructions, but the number of simultaneous driving of these instructions is limited to a total of 8 instructions or less. More than 8 valid instructions are not executed, and the valid instructions are determined according to the effective order of the instructions.

4. The maximum allowable frequency of the PLC high-speed counter. For example, when using DHSCS, DHSCI, DHSCR, DHSZ, DHSP, DHST commands, it will be limited by the maximum response frequency and integrated frequency. For details, refer to Chapter 8 High-speed Input Function User Guide.

● **Example of use**




```
LD    SM255
OUT   Y  10
LD    M1
HCNT  C255 1000
LD    C255
OUT   Y20
LD    M2
DHSCR 2000 C255 Y1
```

1. When M1 and X7 are ON at the same time, C255 counts the phase difference between X4 and X5 in an interrupted manner (the input frequency of the phase difference refers to the instruction of high-speed IO). When the contact drives Y20, the execution of Y20 is determined by the scan cycle of the user program. (Note: C255 is a high-speed counter with a start bit. The start bit is X7)

2. When M2 is ON, when the DHSCR high-speed command meets the high-speed command requirements mentioned in the precautions, Y1 is output immediately when C255 reaches 2000, and is not affected by the scan cycle.

3. When the input pulse of X3 is ahead of X4, SM255 is ON; when the input pulse of X4 is ahead of X3, SM255 is OFF.

4. When X7 (start signal of C255) is OFF, the C255 counter cannot count.

5. When M1 and X7 are ON at the same time, if X5 is ON, the C255 counter is cleared to 0, and the C255 auxiliary contact is also cleared.

6.10.6 DHSZ: High-speed counting interval comparison instruction

Ladder Diagram: DHSZ *(S1)* *(S2)* *(S3)* *(D)*	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
List of instructions: DHSZ *(S1)* *(S2)* *(S3)* *(D)*	**Step size**	**13**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S3	DINT										C					
D	BOOL			Y	M	S										

- **Operand Description**

S1:The data 1 to be compared by the high-speed counter is 32-bit DINT data, and the data range is -2147483648～2147483647

S2:The data 2 to be compared by the high-speed counter is 32-bit DINT data, and the data range is -2147483648～2147483647

S3:High-speed counter, the applicable range of high-speed counter is C236～C263

D: Output bit element object, the processing of Y, M, S is not affected by the scan cycle

- **Function Description**

1. The high-speed counter only counts in interrupt mode according to the count input OFF→ON under the condition of the HCNT drive instruction.

2. When the value of the high-speed counter is less than S1 in the instruction: the bit element specified by D is set, the bit element specified by D+1 is reset, and the bit element specified by D+2 is reset.

3. When the value of the high-speed counter is greater than or equal to S1 and less than or equal to S2: the bit element designated by D is reset, the bit element designated by D+1 is set, and the bit element designated by D+2 is reset.

4. When the value of the high-speed counter is greater than S2 in the DHSZ instruction: the bit element specified by D is reset, the bit element specified by D+1 is reset, and the bit element+2 specified by D is set.

5. If it is a Y element, the Y element will output the corresponding state immediately, and the output action has nothing to do with the program scan cycle.

- **Precautions**

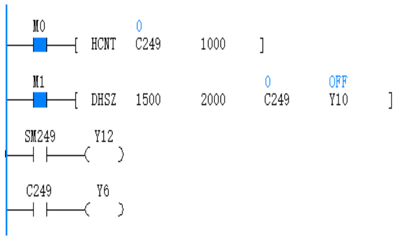
1. The action of the DHSZ command must be used in conjunction with the HCNT command. Only the high-speed counter driven by the HCNT can the DHSZ be used.

2. The DHSZ instruction operates on the comparison result when the pulse is input. Therefore, even if the high-speed counter value is changed with DMOV or MOV instruction, etc., DHSZ will not operate.

3. DHSCZ (DHSCI, DHSCS, DHSCR, DHSP, DHST) can be used multiple times like ordinary instructions, but the number of simultaneous driving of these instructions is limited to a total of 8 instructions or less. More than 8 valid instructions are not executed, and the valid instructions are determined according to the effective order of the instructions.

4. The maximum allowable frequency of the PLC high-speed counter. For example, when using DHSCS, DHSCI, DHSCR, DHSZ, DHSP, DHST commands, it will be limited by the maximum response frequency and integrated frequency. For details, refer to Chapter 8 High-speed Input Function User Guide.

● **Example of use**




```
LD    M0
HCNT  C249  1000
LD    M1
DHSZ  1500  2000  C249  Y10
LD    SM249
OUT   Y12
LD    C249
OUT   Y6
```

1. When M0 is ON, C249 counts up from OFF→ON to X2, and C249 counts down from OFF→ON to X3. (For the two-phase input frequency, refer to the instruction of high-speed IO). When C249 changes from 999 to 1000, the contact of C249 is set. Bit, C249 contact reset from 1001→1000. When the contact point of C249 drives Y6, the execution of Y6 is determined by the scan of the user program.

2. When M1 is ON, when the DHSZ high-speed command meets the high-speed command requirements mentioned in the precautions, the status of Y10, Y11 and Y12 are as follows:

(1) C249<1500: Y10: ON; Y11, Y12: OFF.

(2) 2000≥C249≥1500: Y10, Y12: OFF; Y11: ON.

(3) C249>2000: Y10, Y11: OFF; Y12: ON.

The outputs of Y10, Y11, and Y12 are not affected by the scan period.

3. When M0 is ON, if X2 counts up from OFF→ON, SM249 is reset. If X3 counts down from OFF→ON, SM249 is set.

6.10.7 DHST: High-speed counting table comparison instruction

Ladder Diagram: DHST (S1) (S2) (S3)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: DHST (S1) (S2) (S3)	Step size	10

Operand	Type	Applicable devices														Index
S1	DINT								D						R	
S2	INT	Constant														
S3	DINT										C					

● **Operand Description**

S1: Data start unit for table comparison (D element start number). The three D elements connected with the serial numbers are used to designate the data to be compared by the high-speed counter, the serial numbers of the Y elements and their corresponding output states. these four serial number connected to D The components are collectively called a record.

S2: The number of records to compare, the data range is 1 to 128

S3: High-speed counter, applicable to C236～C263

● **Function Description**

1. The high-speed counter only counts in an interrupt mode according to the count input OFF→ON under the condition of the HCNT drive instruction.

2.When the value of the high-speed counter is equal to the current data to be compared and recorded, the corresponding data is output according to the recorded data. Y Component status, the output object can only be Y element.

3.The output action has nothing to do with the scan cycle, the current record specifies the Y The element will immediately output the specified state.

4.When you want the user program to compare data andYWhen the component performs an immediate output operation, using the DHST table to compare output instructions.

● **Precautions**

1. The DHST instruction action must be used in conjunction with the HCNT instruction. Only the high-speed counter driven by the HCNT can DHST be executed correctly.

2. The DHST instruction operates on the comparison result when the pulse is input. Therefore, even if the high-speed counter value is changed with DMOV or MOV instruction, etc., DHST will not operate.

3. DHST (DHSCI, DHSCS, DHSCR, DHSP, DHSZ) can be used multiple times like ordinary instructions, but the number of simultaneous driving of these instructions is limited to a total of 8 instructions or less. If there are more than 8 commands, the command will not be executed, and the valid commands will be determined according to the valid order of the commands. In the user's command, if DHSP is a valid command, DHST will not be executed. Conversely, if DHST is a valid instruction, DHSP will not be executed. Only one instruction (DHST or DHSP) can be valid in the user program at the same time.

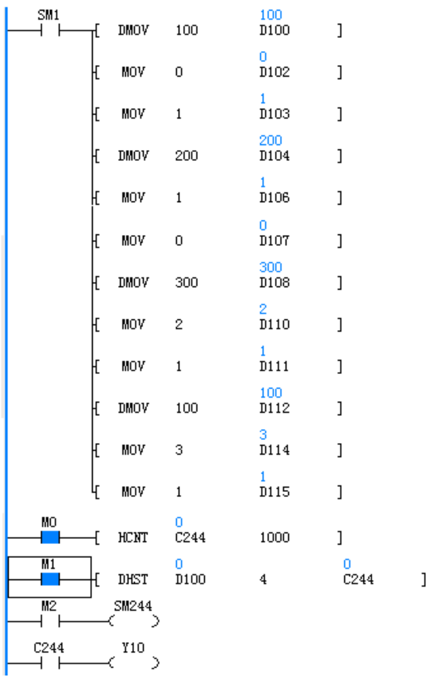
4. The maximum allowable frequency of the PLC high-speed counter. For example, when using DHSCS, DHSCI, DHSCR, DHSZ, DHSP, DHST commands, it will be limited by the maximum response frequency and integrated frequency. specific referenceChapter 8High Speed Input

● **Example of use**

The tabular data is shown in the following table:

Compare data		Output y number	Set/reset	Operating procedures
High	Low			
D100=0	D101=100	D102=0	D103=1	1↓
D104=0	D105=200	D106=1	D107=0	2↓
D108=0	D109=300	D110=2	D111=1	3↓
D112=0	D113=300	D114=3	D115=1	4↓ Back to 1

The ladder diagram is as follows:




```
LD    SM1
DMOV  100  D100
MOV   0    D102
MOV   1    D103
DMOV  200  D104
MOV   1    D106
MOV   0    D107
DMOV  300  D108
MOV   2    D110
MOV   1    D111
DMOV  100  D112
MOV   3    D114
MOV   1    D115
LD    M0
HCNT  C244  1000
LD    M1
DHST  D100  4  C244
LD    M2
OUT   SM244
LD    C244
OUT   Y10
```

1. In the first scan cycle of the user program, initial values are assigned to D100→D115 to generate the table to be compared.

2. When M0 is ON, C244 counts X0 from OFF→ON, (refer to the instruction of high-speed IO for input frequency), when C244 changes from 999→1000, the C244 contact is set, and when it changes from 1001→1000, the C244 contact is reset. When the contact point of C244 drives Y10, the execution of Y10 is determined by the scan cycle of the user program.

3. When M1 is ON, when the DHST high-speed command meets the high-speed command requirements mentioned in the precautions, it starts from the record number 1 of the table, and enters the comparison of the record number 2 after the record number 1 is completed. After completion, enter the next record comparison. When the last record comparison is completed, it returns to the first record comparison, and SM83 is set at the same time. SD90 represents the current record number to be compared,

SD88 and SD89 represent the current data to be compared. The result of the comparison is output immediately and is not affected by the scan cycle.

4. When M2 is ON, SM244 is ON, C244 is down counting, if M2 is OFF, SM244 is OFF, C244 is up counting.

6.10.8 DHSP: High-speed counting table comparison pulse output command

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
├──┤ ├──[DHSP *(S1)* *(S2)* *(S3)*]	**Affect the flag**	
Instruction list: DHSP ***(S1) (S2) (S3)***	**Step size**	**10**

Operand	Type	Applicable devices													Index
S1	DINT								D					R	
S2	INT	Constant													
S3	DINT										C				

● **Operand Description**

S1: Data start unit for table comparison (D element start number). The three D elements followed by serial numbers are used to specify the data to be compared by the high-speed counter, and Output to SD86 and SD87 data. these four serial number connected to D. The components are collectively called a record.

S2: The number of records to compare, the data range is 1 to 128

S3: High-speed counter, applicable to C236～C263

● **Function Description**

1. The high-speed counter only counts in interrupt mode according to the count input OFF→ON under the condition of the HCNT drive instruction.

2. When the value of the high-speed counter is equal to the currently recorded comparison data, SD86 and SD87 are changed according to the currently recorded output data.

3. When you want the user program to decide the assignment of high-speed output or other data according to a certain table, use the DHSP table comparison output instruction. For example, SD86 and SD87 (double word) can be specified as the output frequency operand of the PLSY instruction, so that the PLSY output frequency can be adjusted according to the table comparison result.

● **Precautions**

1. The DHSP instruction action must be used in conjunction with the HCNT instruction. Only the high-speed counter driven by the HCNT can the DHSP be executed correctly.

2. When DHSP and PLSY are used together, the data sent to SD86 and SD87 should meet the frequency output of PLSY. Requires specific reference to the PLSY instruction.

3. When the comparison wants to stop at the last row, the data sent to SD86 and SD87 of the last table is set to 0. In this case, other DHST and DHSP instructions are invalid, but the DHSP instruction at this time does not occupy the total number of other high-speed instructions.

4. The DHSP instruction operates the comparison result when the pulse is input. Therefore, even if the high-speed counter value is changed using DMOV or MOV instruction, etc., DHSP will not operate.

5. DHSP (DHSCI, DHSCS, DHSCR, DHST, DHSZ) can be used multiple times like ordinary instructions, but the number of simultaneous driving of these instructions is limited to a total of 8 instructions or less. If there are more than 8 instructions, the validity of the instructions will be determined according to their effective order.

6. In the user's command, if DHSP is a valid command, DHST will not be executed. Conversely, if DHST is a valid instruction, DHSP will not be executed. Only one instruction (DHST or DHSP) in the user program is valid at the same time, and the others are invalid.

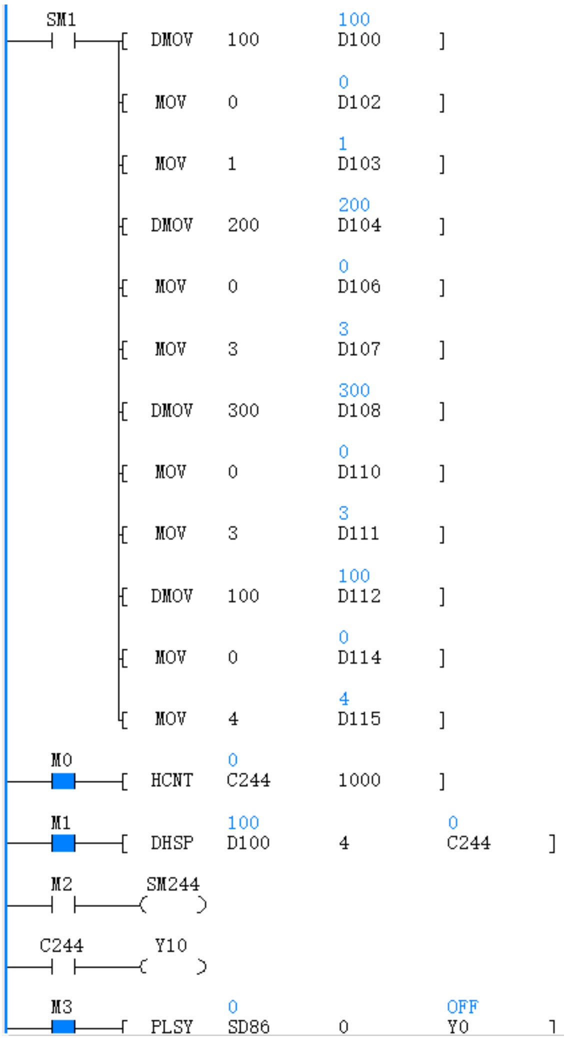
7. Pay attention to the maximum allowable frequency of the PLC high-speed counter. For example, when using DHSCS, DHSCI, DHSCR, DHSZ, DHSP, DHST commands, it will be limited by the maximum response frequency and integrated frequency. specific referenceChapter 8High Speed Input.

● **Example of use**

The tabular data is shown in the following table:

Compare data		Output to SD86 and SD87 data		Operating procedures
High	Low	High	Low	
D100=0	D101=100	D102=0	D103=1	1 ↓
D104=0	D105=200	D106=0	D107=2	2 ↓
D108=0	D109=300	D110=0	D111=3	3 ↓
D112=0	D113=100	D114=0	D115=4	4 ↓ Return from 1

The ladder diagram is as follows:



LD SM1
DMOV 100 D100
MOV 0 D102
MOV 1 D103
DMOV 200 D104
MOV 0 D106
MOV 2 D107
DMOV 300 D108
MOV 0 D110
MOV 3 D111
DMOV 100 D112
MOV 0 D114
MOV 4 D115
LD M0
HCNT C244 1000
LD M1
DHSP D100 4 C244
LD M2
OUT SM244
LD C244
OUT Y10
LD M3
PLSY SD86 0 Y0

1. In the first cycle of user program scan, initial value is assigned to D100→D115 to generate table data to be compared.

2. When M0 is ON, C244 counts X0 from OFF→ON, (refer to the instruction of high-speed IO for input frequency), when C244 changes from 999→1000, the C244 contact is set, and when it changes from 1001→1000, the C244 contact is reset. When the contact point of C244 drives Y10, the execution of Y10 is determined by the scan cycle of the user program.

3. When M1 is ON, when the DHSP high-speed command meets the high-speed command requirements mentioned in the precautions, it starts from the record number 1 of the table, and enters the comparison of the record number 2 after the record number 1 is completed. After completion, enter the next record comparison. When the last record comparison is completed, it returns to the first record comparison, and SM83 is set at the same time. SD90 represents the current record number to be compared, SD88 and SD89 represent the current data to be compared. The output operands for the results of the comparison are respectively placed in the SD86 and SD87 units, which are not affected by the scan cycle. I want the comparison to set the data sent to SD86 and SD87 to 0 for the last grid in the table when the last row stops.

4. When M2 is ON, SM244 is ON, C244 is down counting, if M2 is OFF, SM244 is OFF, C244 is up counting.

6.10.9 SPD: Frequency measurement command

Ladder Diagram: SPD (S1) (S2) (D)															
Applicable models	VC-B VC-S VC-P														
Affect the flag															
Instruction list: SPD (S1) (S2) (D)															
Step size	7														
Operand	Type	Applicable devices													Index
S1	BOOL		X												
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT								D				V		R	√

● **Operand Description**

S1: Input point, settable range: X0～X7

S2: Unit time of input point detection, in ms, operand S2>0

D: Detect pulse data storage unit, when the count exceeds 65535, automatic overflow processing

● **Function Description**

Detect the number of pulses input to X0~X7 within the specified time (ms), and store the result in the specified device unit.

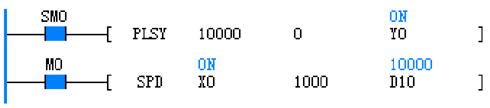
● **Precautions**

1. There are hardware conflicts between SPD and HCNT, external input interrupt, and pulse capture. For details, please refer to Chapter 8 High-speed Input Function User Guide

2. For VC-B VC-S VC-P SPD input points are X0~X7

3. The maximum pulse input frequency of SPD is 10kHz, and there may be errors in detection exceeding 10kHz.

● **Example of use**



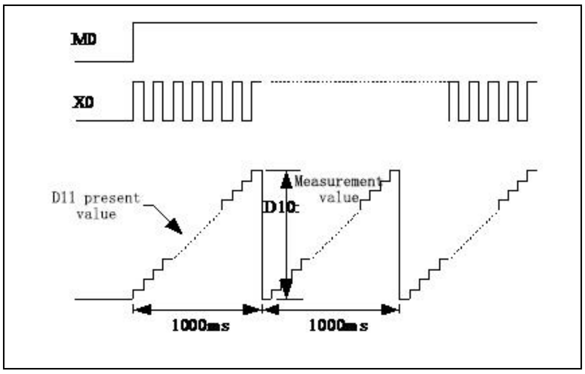
LD SM0

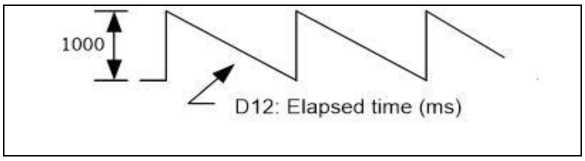
PLSY 10000 0 Y0

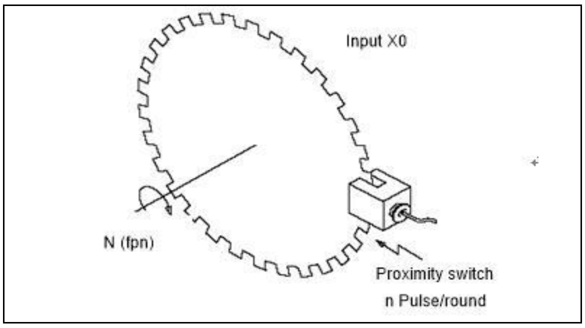
LD M0

SPD X0 1000 D10

The sequence operation of the operation instance of the program is as follows:







1. When M0 is ON, the input pulse specified by X0 is counted within 1000ms, and the counting result is stored in the storage unit of D10, where D11 is the current value of the count within 1000ms, and D12 is the running time within 1000ms.

2. The data of D10 is proportional to the rotation speed in the above figure.

3. Each time OFF→ON of X0 is counted and stored in D10 every 1000ms.

6.10.10 PLSY: High-speed pulse output command

Instruction type	Command name	Reference chapter
high speed command	PLSY pulse output	☞ For details, please refer to Chapter 11 11.3.1

6.10.11 DPLSR: 32-bit variable speed pulse output command with acceleration and deceleration

Instruction type	Command name	Reference chapter
high speed command	DPLSR with acceleration and deceleration variable speed pulse output command	☞ For details, please refer to Chapter 11 11.2.6

6.10.12 PLSR: 16-bit counting pulse output command with acceleration and deceleration

Instruction type	Command name	Reference chapter
high speed command	PLSR with acceleration and deceleration variable speed pulse output command	☞ For details, please refer to Chapter 11 11.2.5

6.10.13 PLS: Multi-speed pulse output command

Instruction type	Command name	Reference chapter
high speed command	PLS multi-speed pulse output command	☞ For details, please refer to Chapter 11 11.2.7

6.10.14 PWM: Pulse output command

Instruction type	Command name	Reference chapter
high speed command	PWM pulse width modulation command	☞ For details, please refer to Chapter 11 11.3.3

6.10.15 HTOUCH: Read position capture command

Instruction type	Command name	Reference chapter
high speed command	HTOUCH read position capture instruction	☞ For details, please refer to Chapter 11 11.3.4

6.11 Control Calculation Instructions

6.11.1 PID: Function command

Ladder Diagram: PID (S1) (S2) (S3) (D)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: PID (S1) (S2) (S3) (D)	Step size	9

Operand	Type	Applicable devices													Index
S1	INT								D					R	√
S2	INT								D					R	√
S3	INT								D					R	√
D	INT								D					R	√

- **Operand Description**

D: When the program is executed, the operation result (MV) is output

S1: Set target value (SV)

S2: Current measured value (PV)

S3: The sampling time (Ts) ranges from 1 to 32767 (ms). Time values shorter than the operation cycle cannot be executed.

S3+1: Action, alarm and upper and lower limit function setting words

bit	Setting values and their meanings	
	0	1
0	Positive feedback	Inverse feedback
1	Input change alarm is invalid	Input change alarm is valid
2	The output change amount alarm is invalid	Output change alarm is valid
3 ~ 4	reserve	
5	The upper and lower limit settings of the output value are invalid	The upper and lower limit settings of the output value are valid
6 ~ 15	reserve	

S3+2: Input filter Constant (α) in the range of 0～99[%], when it is 0, there is no input filter function.

S3+3: Proportional gain (Kp) range from 1 to 32767[%].

S3+4: Integral time (TI) in the range of 0 to 32767 (×100ms), when it is 0, it is treated as ∞ (no integration).

S3+5: Differential gain (KD) range is 0～100[%], when it is 0, there is no differential gain.

S3+6: Differential time (TD) in the range of 0 to 32767 (×10ms), when it is 0, there is no differential processing.

S3+7~S3+14: PID operation internal data storage register.

S3+15: PID input change amount (increase side) alarm setting value 0～32767 (when BIT1=1 of S3+1).

S3+16: PID input change amount (minus side) alarm setting value 0～32767 (when BIT1=1 of S3+1).

S3+17: PID output change amount (increase side) alarm setting value 0～32767 (when BIT2=1 and BIT5=0 of S3+1).

Output upper limit set value -32768～32767 (when BIT2=0 and BIT5=1 of S3+1).

S3+18: PID output change amount (minus side) alarm setting value 0～32767 (when BIT2=1 and BIT5=0 of S3+1).

Output lower limit set value -32768～32767 (when BIT2=0 and BIT5=1 of S3+1).

S3+19: PID alarm output.

BIT0: Input delta (increasing side) overflow.

BIT1: Input delta (minus side) overflow.

BIT2: The output variation (increasing side) overflows.

BIT3: Output variation (minus side) overflow.

Among them, S3~S3+6 are the operands set by the user. S3+15~S3+19 are the operands selected and set by the user. The user can use the PID instruction wizard in the background software to set each operand.

- **Function Description**

1. When the power flow is valid and the sampling time is reached, the PID operation is performed.

2. The PID instruction can be executed multiple times at the same time (the number of loops is unlimited), but it should be noted that the S1, S2, S3 or D soft element numbers used in the operation should not be overwritten repeatedly.

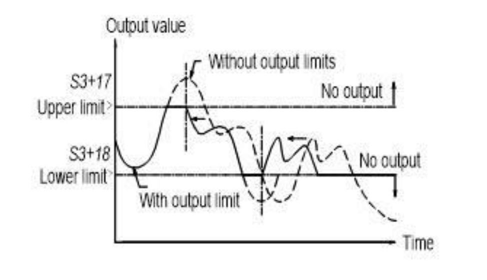
3. PID instruction can be used in timed interrupt subroutine, general subroutine and main program. In this case, before executing the PID instruction, the operand setting unit and the internal processing data unit after clearing S3+7 should be confirmed before use.

4. Entering a filter Constant has the effect of smoothing changes in the measured value.

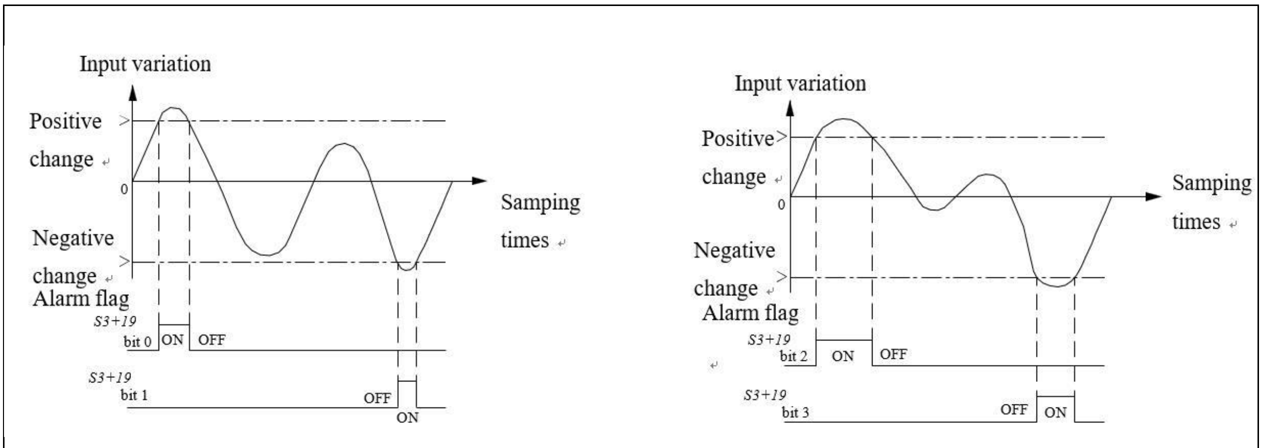
5. Derivative gain has the effect of smoothing out sharp changes in the output value.

6. Action direction: Set the positive action (positive feedback) and reverse action (negative feedback) modes of the system through BIT0 of S3+1.

7. Output upper and lower limit setting: When the output upper and lower limits are set valid (BIT5=ON and BIT2=OFF of S3+1), it can restrain the integral term of PID control from being too large. At this time, the output value is as shown in the following figure:



8. Alarm setting: When the upper and lower limits of the set output are valid (BIT1=ON, BIT2=ON and BIT5=OFF of S3+1), the PID instruction compares the input and output variation with the set values in S3+15 to S3+18 units, and when the set input and output variation is exceeded, the corresponding function bits of the PID alarm output S3+19 units are set immediately after the PID instruction is executed. The corresponding function bits of the PID alarm output S3 + 19 unit are set immediately after the PID instruction is executed. This allows the user to monitor the input variation amount and output variation amount. The output values are shown in the figure below:



9. The basic operation formula of the PID instruction:

Action direction	PID operation formula
Positive action	$\Delta MV = KP\left\{(EV_n - EV_{n-1}) + \frac{T_S}{T_I}EV_n + D_n\right\}$ $EV_n = PV_{nf-1} - SV$ $D_n = \frac{T_D}{T_S + \alpha_D * T_D}(PV_{nf} + PV_{nf-2} - 2PV_{nf-1}) + \frac{\alpha_D * T_D}{T_S + \alpha_D * T_D} * D_{n-1}$ $MV_n = \sum \Delta MV$
Reverse action	$\Delta MV = KP\left\{(EV_n - EV_{n-1}) + \frac{T_S}{T_I}EV_n + D_n\right\}$ $EV_n = SV - PV_{nf-1}$ $D_n = \frac{T_D}{T_S + \alpha_D * T_D}(2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{\alpha_D * T_D}{T_S + \alpha_D * T_D} * D_{n-1}$ $MV_n = \sum \Delta MV$

Symbol descriptions are shown in the table below:

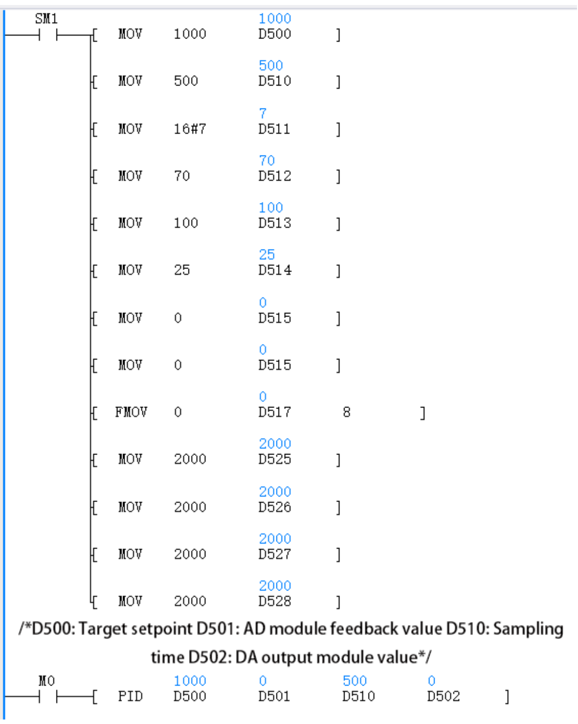
Symbol	Illustrate		Symbol	Illustrate
EV_n	This sampling deviation		D_n	This differential term
EV_{n-1}	Deviation 1 cycle ago		D_{n-1}	Derivative term 1 cycle ago

SV	Target value	KP	Proportional gain
PV_{nf}	This sampling value (after filtering)	T_S	The sampling period
PV_{nf-1}	Sampled value 1 cycle ago (after filtering)	T_I	Integration time
PV_{nf-2}	Sampled value 2 cycles ago (after filtering)	T_D	Differential time
ΔMV	Output change	α_D	Differential gain
MV	The amount of operations this time		

● **Example of use**

```
// PID initialization procedure, if the control operand remains unchanged, it can be executed only once
LD   SM1      //The initializer only needs to be run once
MOV  1000  D500 //set target value
MOV  500  D510  //Sampling time (Ts) The range is 1 to 32767 (ms) but compared to the operation period
         //Short time values cannot be executed
MOV  7  D511   //Action direction
MOV  70  D512  //Input filter Constant (α) range 0～99[%] When it is 0, there is no input filter function
MOV  100  D513  //Proportional gain (Kp) range 1～32767[%]
MOV  25  D514  //The integral time (TI) range is 0 to 32767 (×100ms), when it is 0, it is used as
         //∞ processing (no integration)
MOV  0  D515   //Differential gain (KD) range is 0～100[%], when it is 0, there is no differential gain
MOV  63  D516  //Differential time (TD) range is 0～32767 (×10ms), when it is 0, there is no differentiation
          //deal with
FMOV  0  D517  8//Clear PID operation intermediate data storage area
MOV  2000  D525  //Input change amount (increase side) alarm setting value 0～32767
MOV  2000  D526  //Input change amount (minus side) alarm setting value 0～32767
MOV  2000  D527  //Output change amount (increase side) alarm setting value 0～32767
MOV  2000  D528  //Output change amount (minus side) alarm setting value 0～32767
//PID instruction execute operation
LD   M0      //User-controlled PID operation program
PID  D500 D501 D510 D502 //PID Command: PID S1 S2 S3 D
```

The relevant ladder diagrams of the above instructions are as follows:



When the main module starts to run the first scan cycle, the PID operands are initialized, and the PID operands are not initialized in subsequent scan cycles. When M0=ON, the current measured value is read from the external A/D module (other ways in practical application) and stored in D501. Execute PID operation. The operation result will be converted into an analog signal through an external D/A module (in practical applications, other methods can be used, and added to the controlled system.

● **Precautions**

1. For D, please specify the data register area that is kept in non-stop. If it is specified in the data register area held during shutdown, please be sure to clear it to 0 (LD SM0 MOV 0 D****) when running for the first time.

2. The PID instruction needs to occupy 20 data registers starting from S3.

3. The maximum error of the sampling time TS is -(1 scan period + 1ms) to +(1 scan period). When the TS value is small, it will affect the PID effect. It is recommended to use the PID instruction in the timed interrupt.

4. When the upper and lower limits of PID output are set to be valid, if the upper limit is smaller than the lower limit, the

system reports an operand error and does not execute PID operation.

5. When setting the input and output variation alarm is valid, the set value of S3+15 to S3+18 units cannot be negative, otherwise the system will report an operand error and will not execute PID operation.

6. When BIT2 and BIT5 of S3+1 are ON at the same time, the system will consider the setting invalid (equivalent to BIT2 and BIT5 being OFF at the same time), and will not perform upper and lower limit limit or change over-value alarm.

7. When the set value of the PID control operands (S3~S3+6 units) is not within the valid range, the system will report an operand error, and the PID operation will not be performed.

8. If the sampling time is less than or equal to 1 scan period, data overflow or result overflow occurs during the operation, no alarm will be issued, and the PID operation will continue.

9. Before the PID instruction is executed for the first time, each operand needs to be initialized first. If each operand does not change during operation, and the control operand unit will not be overwritten by other programs, the initialization program can be executed only once. If the data in the PID operation intermediate data storage area is rewritten during the PID operation, the operation result will be incorrect.

6.11.2 RAMP: Ramp signal output command

Ladder Diagram: ├─┤ ├──[RAMP *(S1)* *(S2)* *(D1)* *(S3)* *(D2)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction list: RAMP ***(S1) (S2) (D1) (S3) (D2)***	**Step size**	**12**

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	INT								D				V		R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D2	BOOL			Y	M	S	LM				C	T				

● **Operand Description**

S1: Starting value

S2: End point value

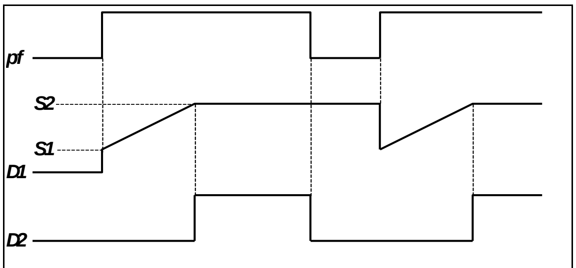
D1: Output value

S3: Number of steps (S3>0, otherwise an operand error is reported, and no operation is performed)

D2: Output state 0

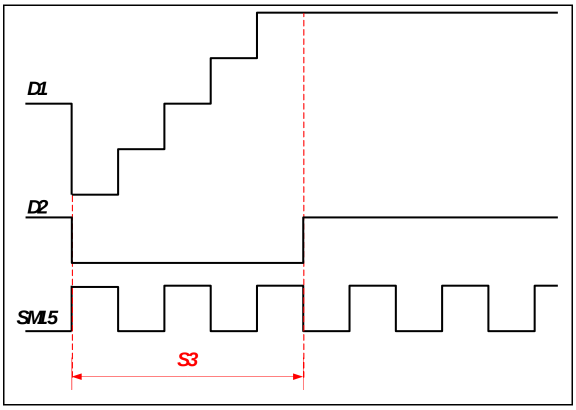
● **Function Description**

When the rising edge of energy flow appears and stays ON, each scan cycle, the increment and current output value are determined by the height of the ramp wave and the number of moving scans, and after reaching S2, the output value (D1) stays in the current state and the state output position becomes ON. If the falling edge of energy flow appears, the output state (D2) is OFF and the output value (D1) stays in the current state until the rising edge of energy flow appears again when the output value (D1) is initialized to the value of S1 and continues to generate the next ramp operation as shown in the following figure:



pf: power flow status

The execution process of the ramp command is broken down as follows (S3=5):



● **Precautions**

1. When the calculation Step size is not divisible, the "rounding" method is adopted

2. The instruction only generates ramp data once for each rising edge.

3. When S1=S2, D1=S2, D2=ON.

4. The total number of RAMP, HACKLE, TRIANGLE instructions in the program cannot exceed 100.

● **Example of use**

//The first scan cycle of power-on comes to initialize the register

LD SM1

MOV 0 D0

MOV 2000 D1

//X0=ON, execute ramp function instruction

LD X0

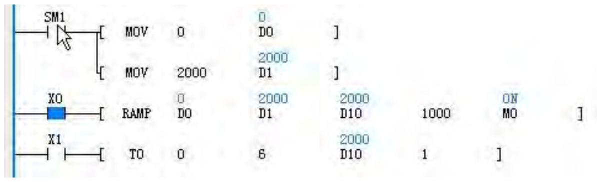
RAMP D0 D1 D10 1000 M0

//X1=ON, send the output value of the ramp function to the external DA module to generate the ramp waveform

LD X1

TO 0 6 D10 1

The ladder diagram of the above instructions is as follows:



1. When X0=ON, D10 increases by 2 (2000/1000) in each scan cycle (D10=D0=0 when the first cycle comes). When D10=D1=2000, D10 will not change, and M0=ON at the same time. During the generation of the ramp function, if the power flow has a falling edge, the output state (D2) is OFF, and the output value (D1) remains in the current state until the next rising edge, D10=D0, and a new ramping process starts again.

2. Users can convert data into analog waveforms through external special modules.

6.11.3 HACKLE: Sawtooth wave signal output command

Ladder Diagram: HACKLE (S1) (S2) (D1) (S3) (D2)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: HACKLE *(S1) (S2) (D1) (S3) (D2)*	Step size	12

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	INT								D				V		R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D2	BOOL			Y	M	S	LM				C	T				

● **Operand Description**

S1: Starting value

S2: End value

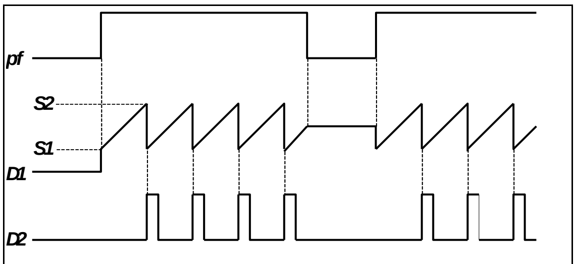
D1: Output value

S3: Number of steps (S3>0, otherwise an operand error is reported, and no operation is performed)

D2: output status

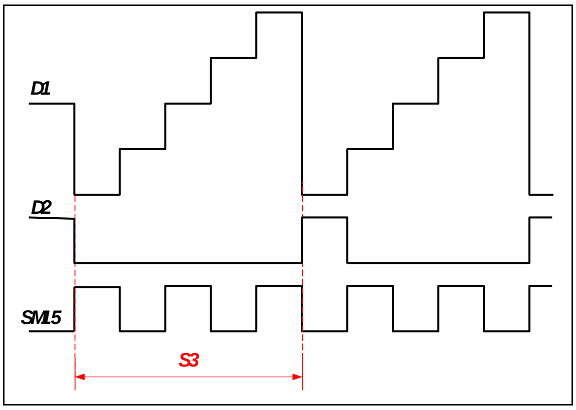
● **Function Description**

When the power flow is valid, the increment and the current output value (D1) are determined according to the height and steps of the sawtooth wave in each scanning period. When the output value reaches S2, it is initialized to the value of S1, and the status output bit (D2) is turned ON. If the power flow continues to be ON in the next scan cycle, turn the status output bit (D2) OFF to continue to generate the next sawtooth wave. During the generation process of the sawtooth wave function, if the power flow has a falling edge, the output state (D2) is OFF, and the output value (D1) remains in the current state, until the power flow has a rising edge again, the output value (D1) is initialized to the value of S1 and continue to generate the next sawtooth operation. As shown below:



pf: power flow status

The execution process of the sawtooth wave instruction is decomposed as follows (S3=5):



● **Precautions**

1. When the calculation Step size is not divisible, the "rounding" method is adopted

As long as the power flow is valid, the instruction will generate a series of continuous sawtooth data

2. When S1=S2, D1=S2, D2=ON (count pulse is not generated)

3. The total number of RAMP, HACKLE, TRIANGLE instructions in the program cannot exceed 100.

● **Example of use**

//The first scan cycle of power-on comes to initialize the register

LD SM1

MOV 0 D0

MOV 2000 D1

//X0=ON, execute sawtooth wave function instruction

LD X0

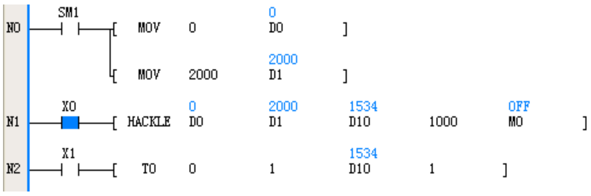
HACKLE D0 D1 D10 1000 M0

//X1=ON, send the output value of the ramp function to the external DA module to generate a sawtooth waveform

LD X1

TO 0 1 D10 1

The ladder diagram of the above instructions is as follows:



1. When X0=ON, D10 increases by 2 (2000/1000) in each scan cycle (D10=D0=0 when the first cycle comes). When D10=D1=2000, M0=ON. In the next scan cycle, if X0 remains ON, D10=D0=0, and M0=OFF at the same time, the next sawtooth wave process starts. If there is a falling edge in the power flow during operation, the output state (D2) will be OFF, and the output value (D1) will keep the current state, until the rising edge of the power flow occurs again, the output value (D1) will be initialized to the value of S1, and the output value (D1) will be reset to the value of S1. Start a new sawtooth process.

2. Users can convert data into analog wave forms through external special modules.

6.11.4 TRIANGLE: Triangular wave signal output command

Ladder Diagram: TRIANGLE (S1) (S2) (D1) (S3) (D2)	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: TRIANGLE (S1) (S2) (D1) (S3) (D2)	Step size	12

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	INT								D				V		R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D2	BOOL			Y	M	S	LM				C	T				

● **Operand Description**

S1: Starting value

S2: End value

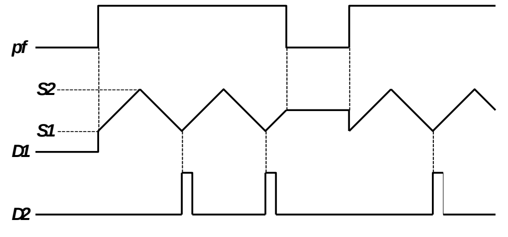
D1: Output value

S3: Number of steps (S3>0, otherwise an operand error is reported, and no operation is performed)

D2: output status

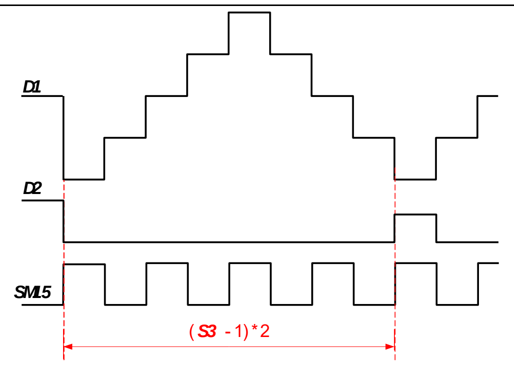
● **Function Description**

When the power flow is valid, the increment and the current output value (D1) are determined according to the height and steps of the triangular wave in each scanning period. When the output value reaches S2, the first half of the triangle wave has been completed, and the incremental direction of the output value is changed to continue to generate the second half of the slope. When the output value (D1) reaches the S1 value again, the status output bit (D2) is turned ON. If the power flow continues to be ON in the next scan cycle, set the status output bit (D2) to OFF, and continue to generate the next triangular wave. During the generation process of the triangular wave function, if the power flow has a falling edge, the output state (D2) is OFF, and the output value (D1) remains in the current state, until the power flow has a rising edge again, the output value (D1) is initialized to the value of S1, start over with a new triangle wave process, as shown in the following figure:



pf: power flow status

The triangular wave instruction execution process is decomposed as follows (S3=5):



● **Precautions**

1. When the calculation Step size is not divisible, the "rounding" method is adopted.

2. As long as the power flow is valid, the instruction will generate a series of continuous triangle wave data.

3. When S1=S2, D1=S2, D2=ON (count pulse is not generated).

4. The period of the triangular wave=(S3-1)×2.

5. The total number of RAMP, HACKLE, TRIANGLE instructions in the program cannot exceed 100.

● **Example of use**

//The first scan cycle of power-on comes to initialize the register

LD SM1

MOV 0 D0

MOV 2000 D1

//X0=ON, execute the triangular wave function instruction

LD X0

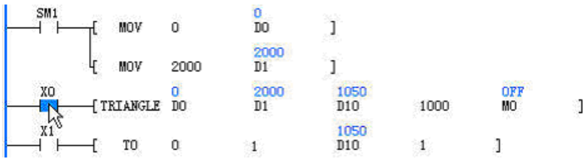
TRIANGLE D0 D1 D10 1000 M0

//X1=ON, send the output value of the ramp function to the external DA module to generate a triangular waveform

LD X1

TO 0 1 D10 1

The ladder diagram of the above instructions is as follows:



1. When X0=ON, D10 increases by 2 (2000/1000) in each scan cycle (D10=D0=0 when the first cycle comes). When D10=D1=2000, the half-slope of the triangular wave is completed, and after that, D10 is reduced by 2 for each scanning period. When D10=D0=0, the complete triangular wave is completed, and M0=ON. In the next scan cycle, if X0 remains ON, M0=OFF, and the next triangular wave process starts. If there is a falling edge in the power flow during operation, the output state (D2) will be OFF, and the output value (D1) will keep the current state, until the rising edge of the power flow occurs again, the output value (D1) will be initialized to the value of S1, and the output value (D1) will be reset to the value of S1. Start a new triangle wave process.

2. Users can convert data into analog waveforms through external special modules.

6.11.5 ALT: Alternate output command

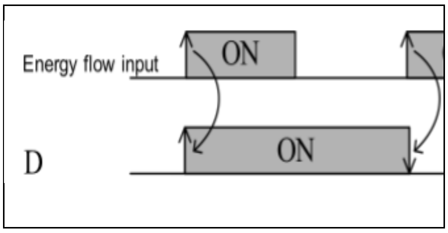
Ladder Diagram: ALT (D)										Applicable models			VC-B VC-S VC-P			
										Affect the flag			Zero flag Carry flag Borrow flag			
Instruction List: ALT (D)										Step size			11			
Operand	Type	Applicable devices														Index
D	BOOL			Y	M	S										

● **Operand Description**

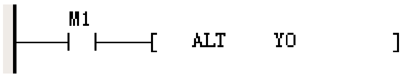
D: Alternate output element address

● **Function Description**

When the power flow is valid, each scan cycle, the device reverses action, as shown in the figure below.



- **Example of use**



6.12 Communication Command

6.12.1 Modbus: Master communication command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
MODBUS (S1) (S2) (S3)	Affect the flag	
Command list: Modbus (S1) (S2) (S3)	**Step size**	**8**

Operand	Type	Applicable devices													Index
S1	INT	Constant													
S2	INT	D	V											R	
S3	INT	D												R	√

- **Operand Description**

S1: Designated communication channel

S2: Start address of sending data

S3: Receive data start address

- **Function Description**

1. As the master station, when the input conditions are met, the data saved from S2 is sent out, and then the data is received and saved to the address unit starting from S3.

2. As a slave station, no command control is required to receive and send data.

3. Executed on rising edge.

- **Precautions**

1. Modbus sends data, no matter whether the user sets it to RTU mode or ASCII mode, it only needs to save the data in RTU form from S2, and it is not necessary to save the start character, end character and checksum. During the sending process, the required start character, end character and check code are added automatically.

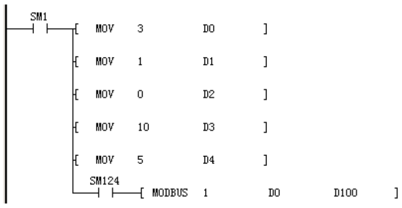
2. The data sent does not need to set the length of the data to be sent, the system will automatically send the data according to the internal set length of the system according to the command, as shown in the following figure:

S2	Slave address
S2+1	Function code
S2+2	Data 1
	… …
S2+N+1	Data N

3. For Modbus reception, no matter the user sets the RTU mode or the ASCII mode, the data is stored in the RTU form, that is, the user sets the ASCII mode, and the system automatically converts it to hexadecimal, removes the start and end characters, and saves it in the Data area starting from S3.

4. The data sent and received are stored in the low byte of the word element, and the high byte is not used.

- **Example of use**




```
LD      SM1
MOV     3   D0
MOV     1   D1
MOV     0   D2
MOV     10  D3
MOV     5   D4
AND     SM124
Modbus  1   D0  D100
```

1. Put the data sent by the Modbus command in the element starting from D0.

2. Save the received data to the element starting from D100.

3. After Modbus receives the data, it first performs CRC check, address check, and function code check. If there is an error, set the error Sign (SM136), and record the specific error information in the special register SD139.

4. The SM114 and SM124 serial port idle Signs can also be used in MODBUS to identify the communication status of MODBUS.

For detailed application examples, seeChapter 10Guidelines for the use of communication functions.

6.12.2 XMT: Free port send command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┤├ ┤├─[XMT (S1) (S2) (S3)]	Affect the flag	
Instruction list: XMT (S1) (S2) (S3)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	INT	D	V												R	
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	R	

- **Operand Description**

S1: The specified communication channel, the value range of VC-B VC-S VC-P is 1, 2

S2: Send data start address

S3: Bytes sent

- **Function Description**

When the power flow is turned on and the communication conditions are met, the data is sent according to the channel and address specified by the user.

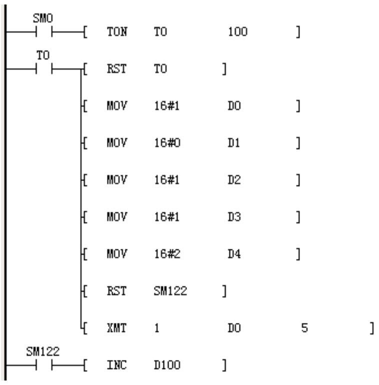
- **Precautions**

1. The size of the communication frame: The communication frame depends on the selected component Type (D or V), and the end character of the transmission frame does not exceed D7999 or V63.
2. In the case of downtime, transmission is aborted.

- **Special register**

1. SM110/SM120: Send enable Sign, this bit is set when the XMT instruction is used, and this bit is cleared when the transmission is over. When this bit is cleared, the current transmission is terminated.
2. SM112/SM122: Sending completion Sign, when it is judged that the sending is complete, the sending completion Sign is set
3. SM114/SM124: Idle Sign, set when the serial port has no communication task. It can be used as a detection bit for communication.
4. For detailed application examples, see Chapter 10 Guidelines for the use of communication functions.

- **Example of use**



LD SM0
TON T0 100
LD T0
RST T0
MOV 16# 1 D0
MOV 16#0 D1
MOV 16#1 D2
MOV 16#1 D3
MOV 16#2 D4
RST SM122
XMT 1 D0 5
LD SM122
INC D100

The routine is to send a frame of data every 10s.

Use serial port 1 to send the following data:

01	00	01	01	02

1. First, set the communication port 1 as a free port in the system block, and then set the baud rate, parity, data bits, stop bits, etc.
2. Write the data to be sent into the send buffer area, and the VC2L sends only the low byte of the word element.
3. Clear the send completion Sign (SM122) before sending data.
4. When the transmission is completed, the transmission completion Sign (SM122) is set.

6.12.3 RCV: Free port receive command

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
┤├ ┤├─[RCV (S1) (D) (S2)]	Affect the flag	

Instruction list: RCV *(S1) (D) (S2)*										**Step size**		**7**			
Operand	Type	Applicable devices													Index
S1	INT	Constant													
D	INT	D	V											R	
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	R

- **Operand Description**

S1: The specified communication channel, the value range of VC-B VC-S VC-P is 1, 2,

D: The starting address where the received data is stored

S2: Maximum number of bytes received

Function Description

When the power flow is turned on and the communication conditions are met, the data is received according to the channel and address specified by the user.

- **Precautions**

1. The size of the communication frame: The communication frame depends on the selected component Type (D or V), and the end character of the received frame does not exceed D7999 or V63.

2. When stopped, reception is aborted.

3. S1The value range is 0, 1, 2.

- **Special register**

SM111 (SM121): Receive enable Sign, this bit is set when the RCV instruction is used, and this bit is cleared when the reception is over. When this bit is cleared, the current reception is terminated.

SM113 (SM123): reception completion Sign, when reception is completed, the reception completion Sign is set.

SM114 (SM124): Idle Sign, set when the serial port has no communication task, it can be used as a communication detection bit.

SD111 (SD121): Start character, which can be set in communication.

SD112 (SD122): End character, which can be set in communication.

SD113 (SD123): Time-out time between characters, that is, the maximum interval time between receiving two characters, which can be set in the communication.

SD114 (SD124): Frame timeout time, the time from the start of power flow to the end of reception, can be set in the communication.

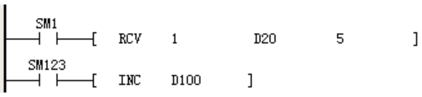
SD115 (SD125): Receiving completion information code, the data bits are defined as follows:

User terminated reception Sign	Received the specified end word Sign	Received max characters Sign	Inter character timeout Sign	(frame) receive timeout Sign	Parity error Sign	Reserved, user can ignore
No. 0	1st place	2nd	3rd	4th place	5th	6th to 15th

SD116 (SD126): The character currently received.

SD117 (SD127): The total number of characters currently received

- **Example of use**



LD SM1
RCV 1 D20 5
LD SM123
INC D100

1. When the power flow is turned on, the RCV instruction will continue to be valid. If you only want to receive it once, you can use the rising edge or a valid special register such as SM1 as the power flow input.

2. For detailed application examples, seeChapter 10Communication Function User Guide**.**

6.12.4 MODRW: MODBUS read and write command

Ladder Diagram: ├─┤ ├──[MODRW (S1) (S2) (S3) (S4) (S5) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: MODRW (S1) (S2) (S3) (S4) (S5) (D)	Step size	14

Operand	Type	Applicable devices													Index
S1	INT	Constant													
S2	INT	Constant	D	V										R	√
S3	INT	Constant	D	V										R	√
S4	INT	Constant	D	V										R	√
S5	INT	Constant	D	V										R	√
D	INT		D											R	√

- **Operand Description**

S1: Designated communication channel VC-B VC-S VC-P can designate channel 0, 1, 2

S2: Address (the slave can set the address to 1~247, and the broadcast address is suitable for writing components).

S3: Function code.

S4: Read and write element start address.

S5: The number of read and write components. The number of read and write components of VC-B VC-S VC-P is limited by the maximum frame length of RTU (256), see the table below.

Function code	Name	Number of components	Number of D components required
01	Read coil	1~2000	(s5+15)/16
02	Read discrete input	1～2000	(s5+15)/16
03	Read register	1~125	S5
04	Read input register	1~125	S5
05	Write a single coil	0 (fixed)	1
06	Write a single register	0 (fixed)	1
15	Write multiple coils	1~1968	(s5+15)/16
16	Write multiple registers	1~123	S5

VC-B VC-S VC-P the number of read and write elements (S5≤16), word elements and bit elements are up to 16, and all bit elements are stored as a word.

Both word elements and bit elements are up to 16. All bit elements are stored as a word.

D1: Read and write element storage address. Refer to the table above for the number of components required for VC.

- **Function Description**

When the power flow is valid, the message is sent and the return data is received.

- **Precautions**

1. The maximum number of components is 16.

2. A maximum of 16 elements can be read, and the small address is stored in the low-order bits, and 1 byte stores 16 bits.

3. The returned exception code is the same as the modbus command.

- **Example of use**

For detailed application examples, seeChapter 10Communication Function User Guide

6.12.5 CANNMT state switching command

Ladder Diagram: ┤├──[CANNMT S1]	Applicable models	VC-S VC-P
	Affect the flag	
Command list: CANNMT *(S1)*	Step size	3

Operand	Type	Applicable devices													Index
S1	WORD		D												√

- **Operand Description**

S1: Switching state, the value range is 1-4, respectively expressed as;

1 reset CANopen communication;

2 reset the CANopen node;

3 Switch to preprocessing mode;

4Switch to run mode.

- **Function Description**

When the power flow is valid, a message is sent to make the CANopen network enter the specified state.

- **Precautions**

When the instruction is being executed, when encountering PLC RUN to STOP, this instruction may not be executed.

- **Example of use**

LD	M0
CANNMT	D0

When M0=ON, D0 inputs different values corresponding to the CANopen network switching to different states.

SM440 =1: The execution of the CANopen instruction is completed (=1 is completed, =0 for the rest).

SM441=1: CANopen command execution error (=1 command error, =0 no error).

SM442 =1: The CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

6.12.6 CANSDORD read command

Ladder Diagram: ┤├──[CANSDORD S1 S2 S3 S4 D1]	Applicable models	VC-S VC-P
	Affect the flag	
Command list: CANSDORD *(S1) (S2) (S3) (S4) (D1)*	Step size	12

Operand	Type	Applicable devices													Index
S1	INT	Constant	D												√
S2	INT	Constant	D												√
S3	INT	Constant	D												√
S4	INT	Constant	D												√
D1	DINT	D													√

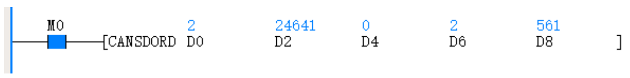
- **Operand Description**

S1: Device address range 1-126.

S2: SDO Index.

S3: SDO subIndex.

- **Example of use**



S4: Read data length, (1, 2, 4 respectively, BYTE, WORD, DWORD).

D1: The data storage address read back, (for BYTE, WORD only occupies 16 bits and is stored in the lower 16 bits).

- **Function Description**

When the power flow is valid, send a message to read the Index data of the specified node.

- **Precautions**

When the instruction is being executed, when encountering PLC RUN to STOP, the instruction may not be executed. Please make sure that the Index and sub-Index read are valid, otherwise an error will be returned.

When M0=NO, read slave 2, the Index is 16#6041, the sub-Index is 0, the data of length WORD, 16#231, is stored in the (D8/D9) register.

SM440 =1: The execution of the CANopen instruction is completed (=1 is completed, =0 for the rest).
SM441=1: CANopen command execution error (=1 command error, =0 no error).

SM442 =1: The CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

6.12.7 CANSDOWR write command

Ladder Diagram: [CANSDOWR S1 S2 S3 S4 D1]	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: CANSDOWR(S1)(S2)(S3)(S4)(D1)	**Step size**	**12**

Operand	Type	Applicable devices													Index
S1	INT	Constant	D												√
S2	TIN	Constant	D												√
S3	TIN	Constant	D												√
S4	TIN	Constant	D												√
D1	DINT	D													√

- **Operand Description**

S1: Device address range 1-126.

S2: SDO Index.

S3: SDO sub Index.

S4: Write data length, (1, 2, 4 respectively, BYTE, WORD, DWORD).

D1: The written data storage address, (for BYTE, WORD only occupies 16 bits and is stored in the lower 16 bits).

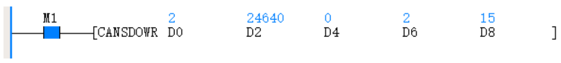
- **Function Description**

When the power flow is valid, send a message and write the Index data of the specified node.

- **Precautions**

When the command is being executed, when encountering PLC RUN to STOP, it may cause the command to fail to be executed. Please make sure that the written Index and sub-Index are valid, otherwise an error will be returned.

- **Example of use**



When M1=NO, write the length WORD data to the slave station 2, the Index is 16#6040, the sub-Index is 0, the written data 16#F is stored in the (D8/D9) register.

SM440 =1: The execution of the CANopen instruction is completed (=1 is completed, =0 for the rest).
SM441=1: CANopen command execution error (=1 command error, =0 no error).

SM442 =1: The CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

6.12.8 CANXMT: CAN free port send command

Ladder Diagram: ┤├──[CANXMT S1 S2 S3 S4]	Applicable models	VC-S VC-P
	Affect the flag	
Command list: CANXMT *(S1)* *(S2)* *(S3)* (S4)	Step size	11

Operand	Type	Applicable devices														Index
S1	DINT	Constant							D						R	√
S2	BOOL				M	S										√
S3	INT								D						R	√
S4	INT								D						R	√

- **Operand Description**

S1: CANID CAN message identifier to be sent

S2: EXT_FRAME: ON: The CAN identifier is marked as an extended frame (29 bits) OFF: The CAN identifier is marked as a standard frame (11 bits)

S3: CAN message data length to be sent (0-8)

S4: CAN message data to be sent (occupies S3 elements consecutive from S4)

- **Function Description**

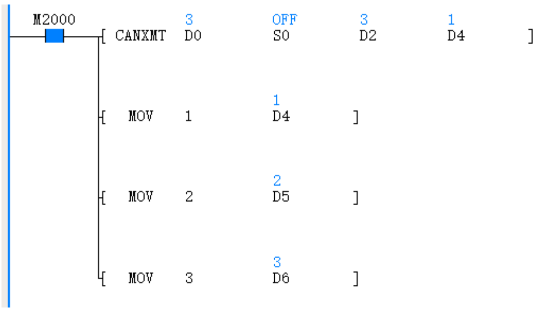
When the power flow is turned on and the CAN port is in an idle state, the CAN message data with S1 as the identifier, the data start address as S5, and the length as S4 is sent.

- **Precautions**

1. After the power flow is turned on, the data will only be sent once. If the data needs to be resent, the power flow must be turned off and turned on again.
2. CANXMT Use up to 32 instructions
3. When multiple CANXMT instruction streams are valid at the same time, they will be sent one by one

Special components	Function description	R/W
SM444	CANXMT command send complete Sign	R/W
SM445	CANXMT command send error Sign	R/W
SD464	CANXMT/CANRCV error code	R

- **Example of use**



The CAN configuration selects the free port protocol. When M2000=ON, it sends 3 data starting from D4, and stores the sent content in the 3 registers starting from D4.

SM440 =1: The execution of the CANopen instruction is completed (=1 is completed, =0 for the rest).

SM441=1: CANopen command execution error (=1 command error, =0 no error).

SM442 =1: The CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

6.12.9 CANRCV: CAN free port receive command

Ladder Diagram: ┤├──[CANRCV S1 S2 S3 S4]	Applicable models	VC-S VC-P
	Affect the flag	
Command list: CANRCV *(S1)* *(S2)* *(S3)* (S4)	Step size	10

Operand	Type	Applicable devices														Index
D1	DINT								D						R	√

D2	BOOL				M	S										√
D3	INT								D						R	√
D4	INT								D						R	√

- **Operand Description**

S1: CANID Received CAN message identifier

S2: EXT_FRAME: ON: The CAN identifier is marked as an extended frame (29 bits) OFF: The CAN identifier is marked as a standard frame (11 bits)

S3: Received CAN message data length (0-8)

S4: Received CAN message data (occupies D3 elements consecutive from D4)

- **Function Description**

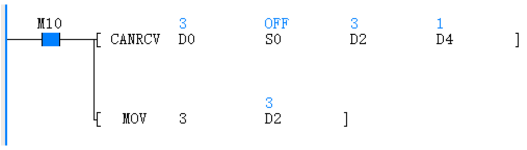
When there are multiple CANRCV instructions, when running to a valid instruction of energy flow, if any CAN port has received data, the received data will be transmitted to the output element of this instruction.

- **Precautions**

1. When SM446 is ON, it means that the CAN port has received a message, and the user needs to remove the data as soon as possible to avoid being overwritten by subsequent data, resulting in data loss. After the data is retrieved, the user needs to reset the SM446 for the next reception.

Special components	Function description	R/W
SM446	CANRCV instruction reception completion Sign	R/W
SM447	CANRCV instruction receive error Sign	R/W
SD464	CANXMT/CANRCVerror code	R

- **Example of use**



The CAN configuration selects the free port protocol. When M10=ON, it accepts 3 data starting from D4, and stores the received content in the 3 registers starting from D4.

SM440 =1: The execution of the CANopen instruction is completed (=1 is completed, =0 for the rest).

SM441=1: CANopen command execution error (=1 command error, =0 no error).

SM442 =1: The CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

6.13 Check Command

6.13.1 CCITT: Check command

Ladder Diagram: ┤├ ┤[CCITT (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: CCITT (S1) (S2) (D)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT								D				V		R	√

● **Operand Description**

S1: The starting unit of the data to be verified

S2: The number of data to be verified (S2≥0, otherwise an operand error is reported)

D: Check result

● **Function Description**

1. Perform the CCITT check operation on the S2 data starting from the starting unit (S1), and assign the result to the D unit.

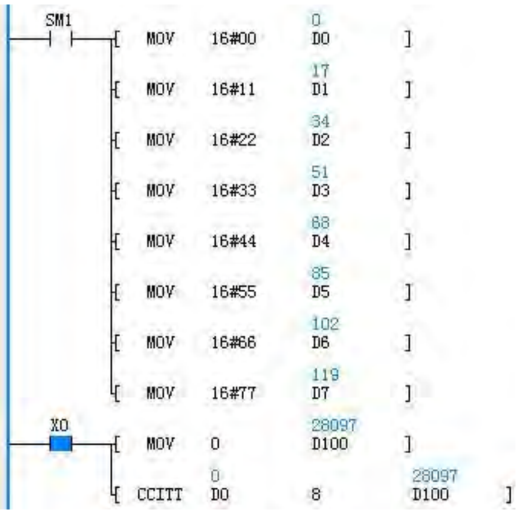
2. The polynomial of the CCITT check algorithm is: X^16+X^12+X^5+1

● **Precautions**

1.Every time an instruction is executed, the system will ***D*** The content of is brought into the operation, so D must be initialized before execution.

2. S2 unit starts Data storage in the check data area, the default is byte mode, that is, the high byte is ignored as zero, the check result is 16 bits.

● **Example of use**




```
LD   SM1
MOV  16#00 D0
MOV  16#11 D1
MOV  16#22 D2
MOV  16#33 D3
MOV  16#44 D4
MOV  16#55 D5
MOV  16#66 D6
MOV  16#77 D7
LD   X0
MOV  0  D100
CCITT  D0 8 D100
```

When X0=ON, the 8 data starting from D0 will be checked by CCITT, and the result will be assigned to D100.

6.13.2 CRC16: Check command

Ladder Diagram: ┤├ ┤[CRC16 (S1) (S2) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: CRC16 (S1) (S2) (D)	Step size	7

Operand	Type	Applicable devices														Index
S1	INT								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT								D				V		R	√

● **Operand Description**

● **Example of use**

S1: The starting unit of the data to be verified

S2: The number of data to be verified (S2≥0, otherwise an operand error is reported)

D: Check result

- **Function Description**

1. The S2 data starting from the starting unit (S1) is subjected to CRC16 check operation, and the result is assigned to the D unit.

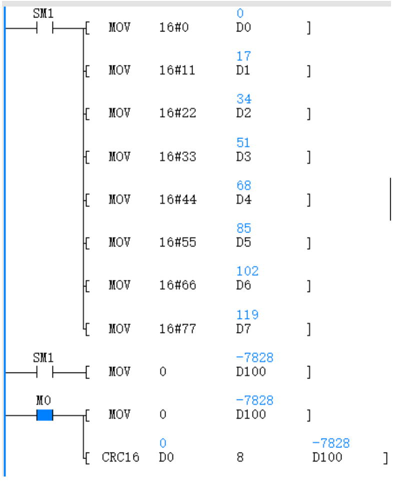
2. The polynomial of the CRC16 check algorithm is: X^16+X^15+X^2+1

- **Precautions**

1. D content will bring the content before the instruction execution into the operation every time the instruction is executed, and D should be initialized before execution.

2. If standard Mod bus CRC check is used, please assign 16#FFFF to the initial value of D element (checksum). And the high and low bytes (high 8 bits, low 8 bits) need to be exchanged.

3.The data storage in the check data area at the beginning of S2 unit is byte mode by default, that is, the high byte is ignored as zero, and the check result is 16 bits.



```
LD    SM1
MOV   16#00 D0
MOV   16#11 D1
MOV   16#22 D2
MOV   16#33 D3
MOV   16#44 D4
MOV   16#55 D5
MOV   16#66 D6
MOV   16#77 D7
LD    X0
MOV   0   D100
CRC16 D0 8 D100
```

When M0=ON, CRC16 check operation is performed on the 8 data starting from D0, and the result is assigned to D100.

6.13.3 LRC: Check command

Ladder Diagram: LRC (S1) (S2) (D)									Applicable models		VC-B VC-S VC-P					
									Affect the flag							
Instruction list: LRC (S1) (S2) (D)									Step size		7					
Operand	Type	Applicable devices														Index
S1	INT								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT								D				V		R	√

- **Operand Description**

S1: The starting unit of the data to be verified

S2: The number of data to be verified, (S2≥0, otherwise an operand error is reported)

D: Check result

- **Function Description**

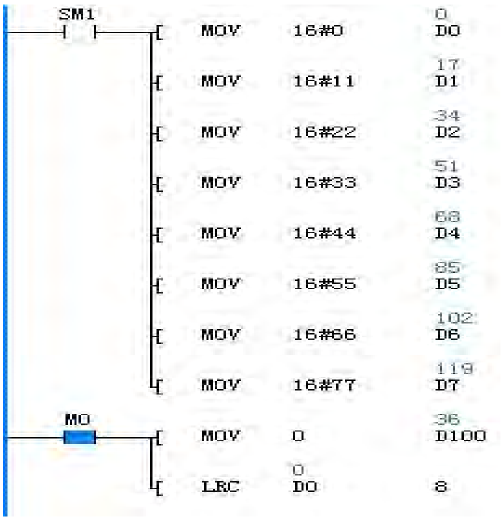
Perform the LRC check operation on the S2 data starting from the starting unit (S1), and assign the result to the D unit.

- **Precautions**

1. D content will bring the content before the instruction execution into the operation every time the instruction is executed, and D should be initialized before execution.

2. The data storage in the check data area at the beginning of S2 unit is byte mode by default, that is, the high byte is ignored as zero, the check result is 8 bits, and it is stored in the low byte of D.

● **Example of use**



When X0=ON, the data of 8 cells starting from D0 is assigned to D100 after LRC checksum operation.

LD SM1
MOV 16#00 D0
MOV 16#11 D1
MOV 16#22 D2
MOV 16#33 D3
MOV 16#44 D4
MOV 16#55 D5
MOV 16#66 D6
MOV 16#77 D7
LD M0
MOV 0 D100
LRC D0 8 D100

6.14 Enhanced Bit Handling Instructions

6.14.1 ZRST: Batch Bit Clear Instruction

Ladder Diagram: ├─┤ ├─[ZRST *(D)* *(S)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction list: ZRST *(D)* *(S)*	**Step size**	**5**

Operand	Type	Applicable devices														Index
D	BOOL			Y	M	S	LM				C	T				√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

● **Operand Description**

D: Destination operand

S: Source operand

● **Function Description**

When the power flow is valid, the S consecutive bit element units starting from the D unit are cleared to zero.

● **Precautions**

1. When the cleared bit element is C, the counter value in the C element will also be cleared.

2. When the cleared bit element is T, the timer value in the T element will also be cleared.

● **Example of use**

LD SM0
ZRST M10 10

When SM0=ON, clear all data of 10 units of M10, M11, M12...M19 starting from M10.

6.14.2 ZSET: atch position setting command

Ladder Diagram: ├─┤ ├─[ZSET *(D)* *(S)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Command list: ZSET *(D)* *(S)*	**Step size**	**5**

Operand	Type	Applicable devices	Index

D	BOOL			Y	M	S	LM				C	T				√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

● **Operand Description**

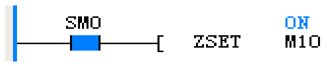
D: Destination operand

S: Source operand

● **Function Description**

When the power flow is valid, set S consecutive bit elements starting from D to 1.

● **Example of use**



LD SM0
ZSET M10
10

When SM0=ON, set all the data of 10 units of M10, M11, M12...M19 starting from M10 to 1.

6.14.3 DECO: Decode instruction

Ladder Diagram: ├─┤ ├──[DECO (S) (D)]		**Applicable models**	**VC-B VC-S VC-P**													
		Affect the flag														
List of instructions: DECO (S) (D)		**Step size**	**5**													
Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

6.14.4 ENCO: Encoding Command

Ladder Diagram: ├─┤ ├──[ENCO (S) (D)]		**Applicable models**	**VC-B VC-S VC-P**													
		Affect the flag														
List of instructions: ENCO (S) (D)		**Step size**	**5**													
Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D		C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

● **Operand Description**

S: Source operand;

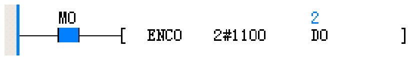
D: Destination operand

● **Function Description**

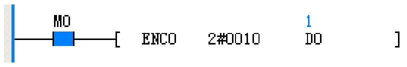
When the power flow is valid, the bit number of the "1" bit in the word element S will be written into D.

● **Precautions**

When "1" appears in several bits of S, the smallest position number will be written into D. As shown in the following figure:



● **Example of use**



LD M0
ENCO 2#0010 D0

When the power flow is valid, the operand 1 is 2#0010, and the first bit is "1", so the result is 1, which is written to D0.

6.14.5 BITS: ON bit statistics instruction in word

Ladder Diagram: ├─┤ ├──[BITS (S) (D)]		**Applicable models**	**VC-B VC-S VC-P**													
		Affect the flag														
Instruction list: BITS (S) (D)		**Step size**	**5**													
Operand	Type	Applicable devices														Index
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

● **Operand Description**

S: Source operand;

D: Destination operand

● **Function Description**

When the power flow is valid, the number of 1 bit in the operand S is counted, and the statistical result is stored in the operand D.

● **Example of use**

LD SM0
BITS 16#F0F0 D1

The power flow is valid. In the BITS instruction, S is the Constant 16#F0F0, 8 bits are "1" (ON state), and the calculation result is 8 and stored in D (D1).

LD SM0
BITS 16#F0F0 D1

6.14.6 DBITS: ON bit statistics instruction in double word

Ladder Diagram: ├─┤ ├──[DBITS (S) (D)]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction List: DBITS (S) (D)	**Step size**	**6**

Operand	Type	Applicable devices														Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

● **Operand Description**

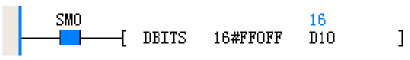
S: Source operand;

D: Destination operand

● **Function Description**

When the power flow is valid, the number of "1" bits in the double word S is counted, and the statistical result is stored in D.

● **Example of use**



LD SM0
DBITS 16#FF0FF D10

The power flow is valid. In the DBITS instruction, S is the Constant 16#FF0FF, 16 bits are "1" (ON state), and the calculation result is 16, which is stored in D (D10).

6.14.7 BON: ON bit judgment command in word

Ladder Diagram: ├─┤ ├──[BON (S1) (D) (S2)]	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Instruction list: BON (S1) (D) (S2)	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V			√
D	BOOL		Y	M	S										
S2	INT								D				V		R	

● **Operand Description**

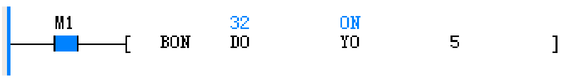
S: Source operand;

D: Destination operand

● **Function Description**

When the power flow is valid, the state of the S2 bit in the statistics word S1 is output to D.

● **Example of use**



LD M1
BON D0 Y0 5

The power flow is valid. In the BON instruction, S1 is a Constant D0, and the state of the 5th bit (ON) is output to D (Y0).

6.15 Word Contact Command

6.15.1 BLD: Word bit contact LD instruction

Ladder Diagram: ├──┤ BLD (S1) (S2) ├─( )	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: BLD (S1) (S2)	Step size	5

Operand	Type	Applicable devices														Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand

S2: Specify the bit, 0≤S2≤15, otherwise an operand error is reported.

- **Function Description**

The state of the S2 bit of the content of the S1 unit is used to drive the subsequent operation.

- **Example of use**

BLD D0 5
OUT Y0

Take the state (ON) of BIT5 of D0 (1000: 2#0000001111101000) to determine the state of the rear element Y0.

6.15.2 BLDI: Word bit contact LDI instruction

Ladder Diagram: ├──┤ BLDI (S1) (S2) ├─( )	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: BLDI (S1) (S2)	Step size	5

Operand	Type	Applicable devices														Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

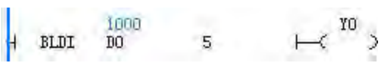
S1: Source operand

S2: Specify the bit, 0≤S2≤15, otherwise an operand error is reported.

- **Function Description**

The logical negation of the S2 bit state of the content of the S1 unit is used to drive the subsequent operation.

- **Example of use**



BLDI D0 5
OUT Y0

Take the logical negation (OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) to determine the output state of the rear element Y0.

6.15.3 BAND: Word bit contact AND instruction

Ladder Diagram Note: ├──┤ BLD (S1) (S2) ├─( )	Applicable models	VC-B VC-S VC-P
Note: Because the logical relationship has been reflected in the graphics, the BAND instruction is displayed as BLD in the ladder diagram	Affect the flag	
Instruction list: BAND (S1) (S2)	Step size	5

Operand	Type	Applicable devices	Index

S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

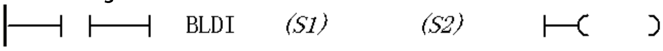
The state of the S2 bit of the content of the S1 unit is taken in series with other nodes to drive the subsequent operation.

- **Example of use**

LD X0
BAND D0 5
OUT Y0

take D0(1000: 2#0000001111101000) The state (ON) of BIT5 is connected in series with other nodes (X0=ON) to determine the output state of the rear element Y0.

6.15.4 BANI: Word bit contact ANI instruction

Ladder Diagram Note:			**Applicable models**	**VC-B VC-S VC-P**
 Note: Because the logical relationship has been reflected in the graphics, the BANI instruction is displayed as BLDI in the ladder diagram			**Affect the flag**	
Instruction list: BANI (S1) (S2)			**Step size**	**5**

Operand	Type	Applicable devices														Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

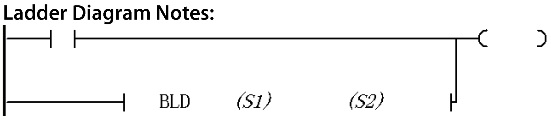
The logical negation of the state of the S2 bit of the content of the S1 unit is connected in series with other nodes to drive the subsequent operation.

- **Example of use**

LD X0
BANI D0 5
OUT Y0

Take the logical negation (OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) and other nodes (X0=ON) in series to determine the output state of the rear element Y0.

6.15.5 BOR: Word bit contact OR instruction

Ladder Diagram Notes:			**Applicable models**	**VC-B VC-S VC-P**
 Note: Because the logical relationship has been reflected in the graphics, the BOR instruction is displayed as BLD in the ladder diagram			**Affect the flag**	
Instruction list: BOR (S1) (S2)			**Step size**	**5**

Operand	Type	Applicable devices														Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

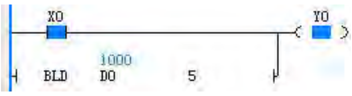
S1: Source operand

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

Take the state of the S2 bit of the content of the S1 unit in parallel with other nodes to drive the subsequent operation

- **Example of use**



LD X0
BOR D0 5
OUT Y0

Take the state (ON) of BIT5 of D0 (1000: 2#0000001111101000) in parallel with other nodes (X0=ON) to determine the output state of the rear element Y0.

6.15.6 BORI: Word bit contact ORI instruction

Ladder Diagram[Note]: BLDI (S1) (S2) Note: Because the logical relationship has been reflected in the graphics, the BORI instruction is displayed as BLDI in the ladder diagram	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction list: BORI *(S1) (S2)*	**Step size**	**5**

Operand	Type	Applicable devices													Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

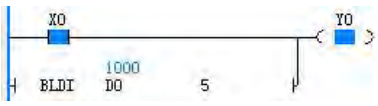
S1: Source operand

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

The logical negation of the S2 bit state of the content of the S1 unit is connected in parallel with other nodes to drive the subsequent operation.

- **Example of use**



LD X0
BORI D0 5
OUT Y0

Take the logical negation (OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) and connect it in parallel with other nodes (X0=ON) to determine the output state of the rear element Y0.

6.15.7 BOUT: Word bit coil output command

Ladder Diagram: BOUT (D) (S)	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction list: BOUT *(D) (S)*	**Step size**	**5**

Operand	Type	Applicable devices													Index
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Source operand

- **Example of use**

LD X0
BOUT D0 4

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

Assign the current power flow state to the S bit of D.

Assign the current power flow state (X0=ON) to BIT4 of D0 (1000:2#0000001111101000). After execution D0=1016 (2#0000001111111000)

6.15.8 BSET: Word bit coil set command

Ladder Diagram: ├─┤ ├─[BSET (D) (S)]														**Applicable models**		**VC-B VC-S VC-P**
														Affect the flag		
Instruction List: BSET (D) (S)														**Step size**		**5**
Operand	Type	Applicable devices														Index
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

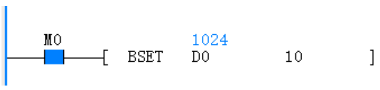
D: Destination operand

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

Set the S bit of the D element.

- **Example of use**



LD M0
BSET D0 10

When M0 is ON, set BIT10 of D0 (D0=0: 2#0000000000000000). After execution D0=1024 (2#0000010000000000)

6.15.9 BRST: Word bit coil clear command

Ladder Diagram: ├─┤ ├─[BRST (D) (S)]														**Applicable models**		**VC-B VC-S VC-P**
														Affect the flag		
Instruction List: BRST (D) (S)														**Step size**		**5**
Operand	Type	Applicable devices														Index
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

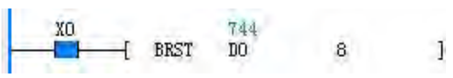
D: Destination operand

S2: Specified bit (0≤S2≤15, otherwise an operand error is reported)

- **Function Description**

Clear the S bit of the D element.

- **Example of use**



LD X0
BRST D0 8

When the power flow is valid, clear BIT8 of D0 (1000: 2#0000001111101000). After execution D0=744 (2#0000001011101000)

6.16 Compare Contact Instructions

6.16.1 LD (=, <, >, <>, >=, <=): Integer comparison contact instruction

Ladder Diagram: = (S1) (S2) < (S1) (S2) > (S1) (S2) <> (S1) (S2) >= (S1) (S2) <= (S1) (S2)	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction list: LD= (S1) (S2) LD< (S1) (S2) LD> (S1) (S2) LD<> (S1) (S2) LD>= (S1) (S2) LD<= (S1) (S2)	**Step size**	5

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

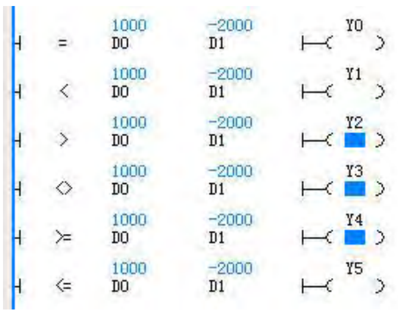
S1: Compare parameter 1

S2: Compare parameter 2

- **Function Description**

BIN comparison is performed on the contents of S1 and S2 units, and the result of the comparison is used to drive the subsequent operation.

- **Example of use**




```
LD=   D0  D1
OUT   Y0
LD<   D0  D1
OUT   Y1
LD>   D0  D1
OUT   2
LD<>  D0  D1
OUT   Y3
LD>=  D0  D1
OUT   Y4
LD<=  D0  D1
OUT   Y5
```

BIN comparison is performed on the data of D0 and D1, and the result of the comparison determines the output state of the latter element.

6.16.2 AND (=, <, >, <>, >=, <=): Integer compares contacts with instructions

Ladder Diagram: = (S1) (S2) < (S1) (S2) > (S1) (S2) <> (S1) (S2) >= (S1) (S2) <= (S1) (S2)	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	

Instruction list:	Step size	5
AND＝ （S1） （S2） AND＜ （S1） （S2） AND＞ （S1） （S2） AND＜＞ （S1） （S2） AND＞＝ （S1） （S2） AND＜＝ （S1） （S2）		

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

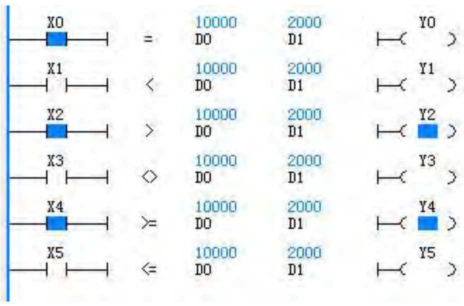
S1: Compare parameter 1

S2: Compare parameter 2

- **Function Description**

The contents of S1 and S2 units are compared with BIN, and the result of the comparison is connected in series with other nodes to drive the back-end operation.

- **Example of use**

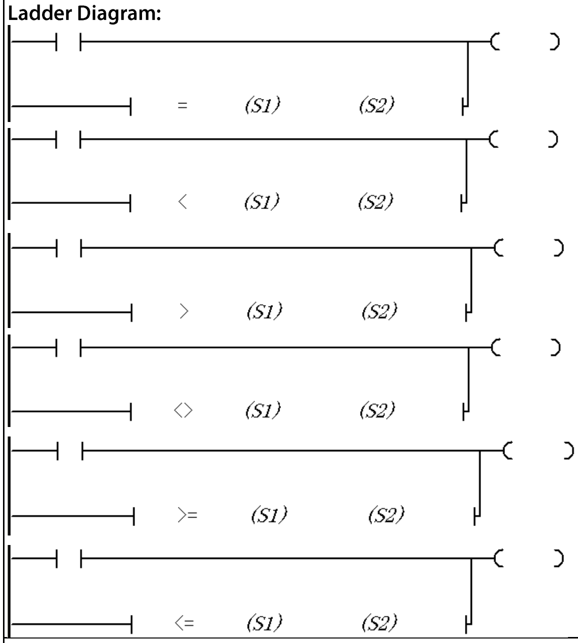


LD X0
AND= D0 D1
OUT Y0
LD X1
AND< D0 D1
OUT Y1
LD X2
AND> D0 D1
OUT Y2
LD X3
AND<> D0 D1
OUT Y3
LD X4
AND>= D0 D1
OUT Y4
LD X5
AND<= D0 D1
OUT Y5

BIN comparison is performed on the data of D0 and D1, and the result of the comparison is connected in series with other nodes to determine the output state of the latter element.

6.16.3 OR (=, <, >, <>, >=, <=): Integer comparison contacts or instructions

Ladder Diagram:



Applicable models	**VC-B VC-S VC-P**
Affect the flag	

Instruction list:

OR= (S1) (S2)

OR< (S1) (S2)

OR> (S1) (S2)

OR<> (S1) (S2)

OR>= (S1) (S2)

OR<= (S1) (S2)

Step size	**5**

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

● **Operand Description**

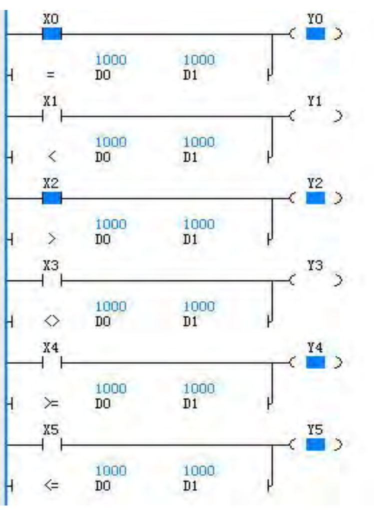
S1: Compare parameter 1

S2: Compare parameter 2

● **Function Description**

Compare the contents of S1 and S2 units, and the result of the comparison is connected in parallel with other nodes to drive the back-end operation.

● **Example of use**




```
LD X0
OR= D0 D1
OUT Y0
LD X1
OR< D0 D1
OUT Y1
LD X2
OR> D0 D1
OUT Y2
LD X3
OR<> D0 D2
OUT Y3
LD X4
OR>= D0 D1
OUT Y4
LD X5
OR<= D0 D1
OUT Y5
```

The data of D0 and D1 are compared, and the result of the comparison is connected in parallel with other nodes to determine the output state of the latter element

.

6.16.4 LDD (=, <, >, <>, >=, <=): long integer comparison contact instruction

<table>
<tr><td rowspan="2">Ladder Diagram:
D= (S1) (S2)
D< (S1) (S2)
D> (S1) (S2)
D<> (S1) (S2)
D>= (S1) (S2)
D<= (S1) (S2)</td><td>Applicable models</td><td>VC-B VC-S VC-P</td></tr>
<tr><td>Affect the flag</td><td></td></tr>
<tr><td>Instruction list:
LDD= (S1) (S2)
LDD< (S1) (S2)
LDD> (S1) (S2)
LDD<> (S1) (S2)
LDD>= (S1) (S2)
LDD<= (S1) (S2)</td><td>Step size</td><td>7</td></tr>
</table>

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

- **Operand Description**

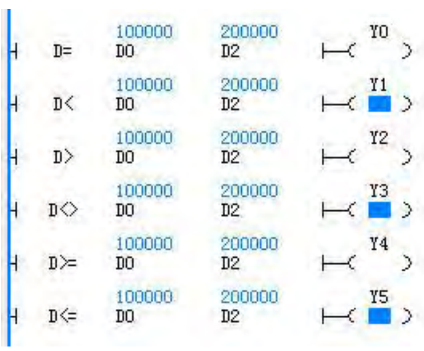
S1: Compare parameter 1

S2: Compare parameter 2

- **Function Description**

Compare the contents of S1 and S2 units, and the result of the comparison is used to drive the subsequent operation.

- **Example of use**



LDD= D0 D2
OUT Y0
LDD< D0 D2
OUT Y1
LDD> D0 D2
OUT Y2
LDD<> D0 D2
OUT Y3
LDD>= D0 D2
OUT Y4
LDD<= D0 D2
OUT Y5

Compare (D0, D1) and (D2, D3), and the comparison result determines the output state of the latter element.

6.16.5 ANDD (=, <, >, <>, >=, <=): Long integer compare contacts with instructions

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
D= (S1) (S2) D< (S1) (S2) D> (S1) (S2) D<> (S1) (S2) D>= (S1) (S2) D<= (S1) (S2)	Affect the flag	
Instruction list: **ANDD= (S1) (S2)** **ANDD< (S1) (S2)** **ANDD> (S1) (S2)** **ANDD<> (S1) (S2)** **ANDD>= (S1) (S2)** **ANDD<= (S1) (S2)**	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

- **Operand Description**

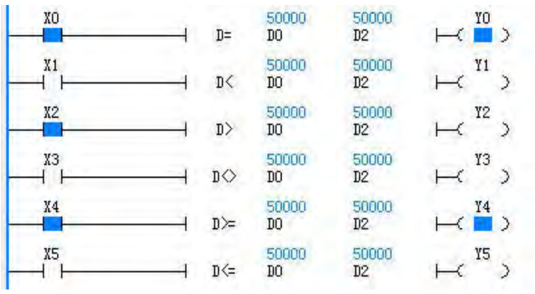
S1: Compare parameter 1

S2: Compare parameter 2

- **Function Description**

Compare the contents of the S1 and S2 units, and the result of the comparison is connected in series with other nodes to drive the back-end operation.

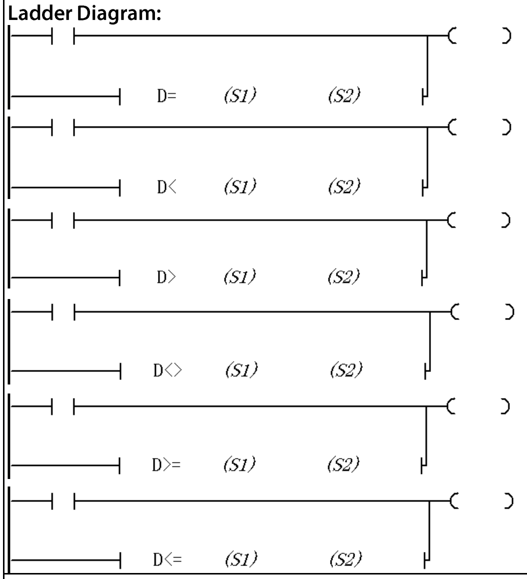
- **Example of use**



LD X0
ANDD= D0 D2
OUT Y0
LD X1
ANDD< D0 D2
OUT Y1
LD X2
ANDD> D0 D2
OUT Y2
LD X3
ANDD<> D0 D2
OUT Y3
LD X4
ANDD>= D0 D2
OUT Y4
LD X5
ANDD<= D0 D2
OUT Y5

Comparing (D0, D1) and (D2, D3), the result of the comparison is connected in series with other nodes to determine the output state of the latter element.

6.16.6 ORD (=, <, >, <>, >=, <=): Long integer comparison contact or instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction list: ORD= (S1) (S2) ORD< (S1) (S2) ORD> (S1) (S2) ORD<> (S1) (S2) ORD>= (S1) (S2) ORD<= (S1) (S2)	Step size	7

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

● **Operand Description**

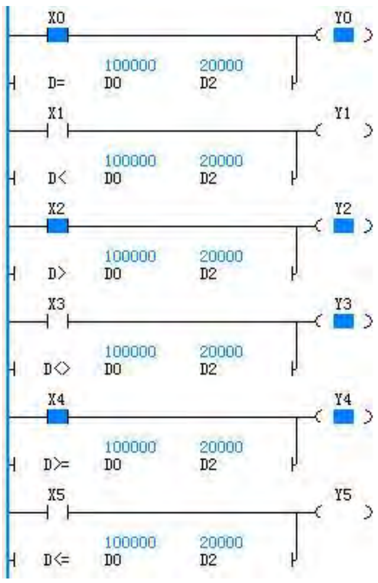
S1: Compare parameter 1

S2: Compare parameter 2

● **Function Description**

Compare the contents of S1 and S2 units, and the result of the comparison is connected in parallel with other nodes to drive the back-end operation.

● **Example of use**




```
LD      X0
ORD=    D0 D2
OUT  Y0
LD      X1
ORD<    D0 D2
OUT     Y1
LD      X2
ORD>    D0 D2
OUT     Y2
LD      X3
ORD<>   D0 D2
OUT     Y3
LD      X4
ORD>=   D0 D2
OUT     Y4
LD      X5
ORD<=   D0 D2
OUT     Y5
```

Compare (D0, D1) and (D2, D3), and the result of the comparison is connected in parallel with other nodes to determine the output state of the components in the rear section.

6.16.7 LDR (=, <, >, <>, >=, <=): Floating point comparison contact instruction

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
R= (S1) (S2) ( ) R< (S1) (S2) ( ) R> (S1) (S2) ( ) R<> (S1) (S2) ( ) R>= (S1) (S2) ( ) R<= (S1) (S2) ( )	Affect the flag	
Instruction list: LDR= (S1) (S2) LDR< (S1) (S2) LDR> (S1) (S2) LDR<> (S1) (S2) LDR>= (S1) (S2) LDR<= (S1) (S2)	Step size	7

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	RAEL	Constant							D				V		R	√

● **Operand Description**

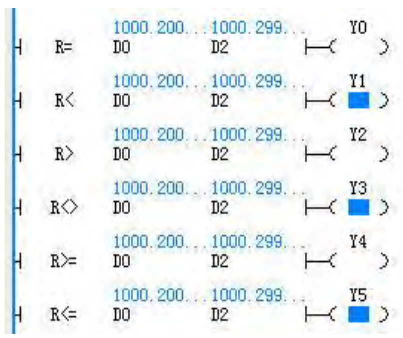
S1: Compare parameter 1

S2: Compare parameter 2

● **Function Description**

Compare the contents of S1 and S2 units, and the result of the comparison is used to drive the subsequent operation.

● **Example of use**




```
LDR=    D0 D2
OUT     Y0
LDR<    D0 D2
OUT     Y1
LDR>    D0 D2
OUT     Y2
LDR<>   D0 D2
OUT     Y3
LDR>=   D0 D2
OUT     Y4
LDR<=   D0 D2
OUT     Y5
```

Compare (D0, D1) and (D2, D3), and the result of the comparison determines the output state of the latter element.

6.16.8 ANDR (=, <, >, <>, >=, <=): Floating point comparison contacts and instructions

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
R= (S1) (S2) ( ) R< (S1) (S2) ( ) R> (S1) (S2) ( ) R<> (S1) (S2) ( ) R>= (S1) (S2) ( ) R<= (S1) (S2) ( )	Affect the flag	

Instruction list: ANDR= (S1) (S2) ANDR< (S1) (S2) ANDR> (S1) (S2) ANDR<> (S1) (S2) ANDR>= (S1) (S2) ANDR<= (S1) (S2)	**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√

● **Operand Description**

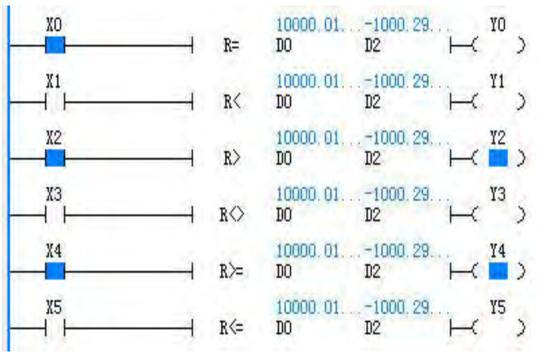
S1: Compare parameter 1

S2: Compare parameter 2

● **Function Description**

Compare the contents of the S1 and S2 units, and the result of the comparison is connected in series with other nodes to drive the back-end operation.

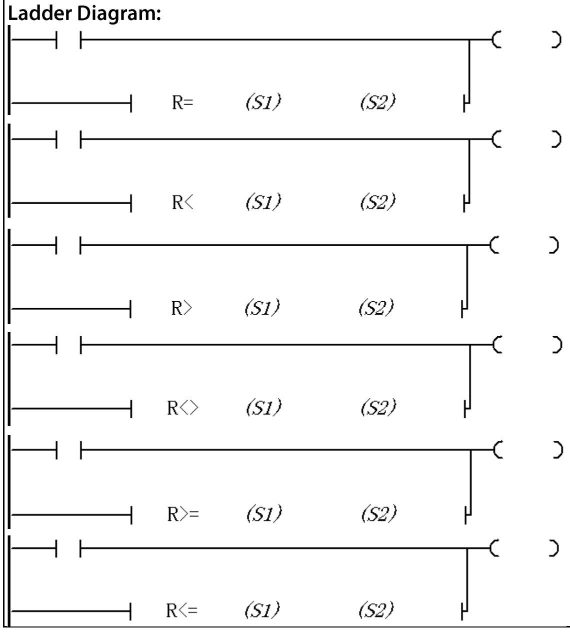
● **Example of use**




```
LD X0
ANDR= D0 D2
OUT Y0
LD X1
ANDR< D0 D2
OUT Y1
LD X2
ANDR> D0 D2
OUT Y2
LD X3
ANDR<> Y3
LD X4
ANDR>= D0 D2
OUT Y4
LD X5
ANDR<= D0 D2
OUT Y5
```

Comparing (D0, D1) and (D2, D3), the result of the comparison is connected in series with other nodes to determine the output state of the latter element.

6.16.9 ORR (=, <, >, <>, >=, <=): Floating point comparison contact or instruction

Ladder Diagram:	**Applicable models**	**VC-B VC-S VC-P**
	**Affect the flag**	

Instruction list:	Step size	7
ORR＝ （S1） （S2） ORR＜ （S1） （S2） ORR＞ （S1） （S2） ORR＜＞ （S1） （S2） ORR＞＝ （S1） （S2） ORR＜＝ （S1） （S2）		

Operand	Type	Applicable devices														Index
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√

● **Operand Description**

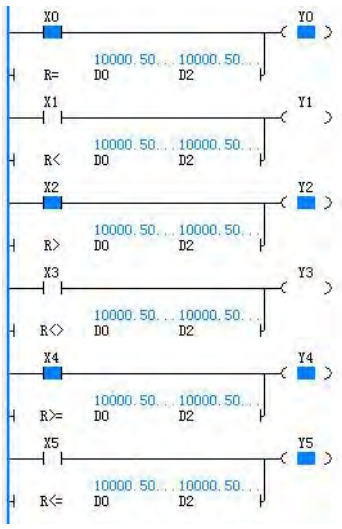
S1: Compare parameter 1

S2: Compare parameter 2

● **Function Description**

Compare the contents of S1 and S2 units, and the result of the comparison is connected in parallel with other nodes to drive the back-end operation.

● **Example of use**




```
LD      X0
ORR=    D0 D2
OUT     Y0
LD      X1
ORR<    D0 D2
OUT     Y1
LD      X2
ORR>    D0 D2
OUT     Y2
LD      X3
ORR<>   D0 D2
OUT     Y3
LD      X4
ORR>=   D0 D2
OUT     Y4
LD      X5
ORR<=   D0 D2
OUT     Y5
```

Compare (D0, D1) and (D2, D3), and the result of the comparison is connected in parallel with other nodes to determine the output state of the components in the rear section.

6.16.10 LDZ (=, <, >, <>, >=, <=): Integer absolute value comparison contact instruction

Ladder Diagram:	Applicable models	VC-S VC-P
Z= S1 S2 S3 Z< S1 S2 S3 Z> S1 S2 S3 Z<> S1 S2 S3 Z>= S1 S2 S3 Z<= S1 S2 S3	Affect the flag	
Instruction list: LDZ= (S1) (S2) (S3) LDZ< (S1) (S2) (S3) LDZ> (S1) (S2) (S3) LDZ<> (S1) (S2) (S3) LDZ>= (S1) (S2) (S3) LDZ<= (S1) (S2) (S3)	Step size	7

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Minuend

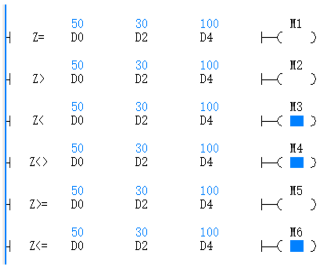
S2: Subtraction

S3: Comparison value

- **Function Description**

Compare the absolute value of the result of the subtraction of S1 and S2 with the absolute value of S3, turn the contact ON or OFF according to the comparison result, and the node directly connected to the left bus.

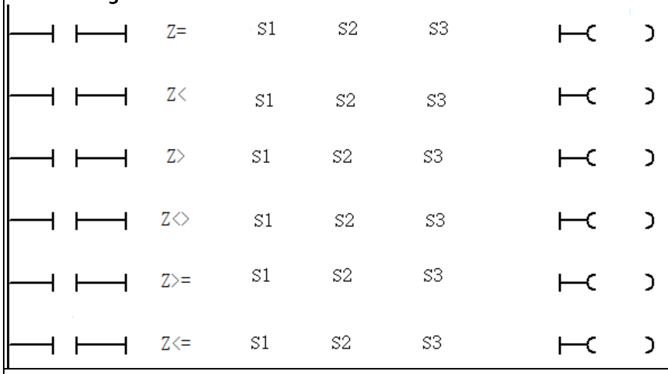
16-bit instructions	Conduction conditions	Non-conducting conditions
LDZ=	|S1-S2|=|S3|	|S1-S2|<>|S3|
LDZ<	|S1-S2|<|S3|	|S1-S2|>=|S3|
LDZ>	|S1-S2|>|S3|	|S1-S2|<=|S3|
LDZ<>	|S1-S2|<>|S3|	|S1-S2|=|S3|
LDZ>=	|S1-S2|>=|S3|	|S1-S2|<|S3|
LDZ<=	|S1-S2|<=|S3|	|S1-S2|>|S3|



LDZ= D0 D2 D4
OUT M1
LDZ> D0 D2 D4
OUT M2
LDZ< D0 D2 D4
OUT M3
LDZ<> D0 D2 D4
OUT M4
LDZ>= D0 D2 D4
OUT M5
LDZ<= D0 D2 D4
OUT M6

The absolute value of the result after subtracting the D2 register value from the D0 register value is compared with the absolute value of the D4 register value, and the contact is turned ON or OFF according to the comparison result.

6.16.11 ANDZ (=, <, >, <>, >=, <=): Integer absolute value comparison of contacts and instructions

Ladder Diagram: 	Applicable models	VC-S VC-P
	Affect the flag	
Instruction list: ANDZ= (S1) (S2) (S3) **ANDZ < (S1) (S2) (S3)** **ANDZ> (S1) (S2) (S3)** **ANDZ<> (S1) (S2) (S3)** **ANDZ>= (S1) (S2) (S3)** **ANDZ <= (S1) (S2) (S3)**	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Minuend

S2: Subtraction

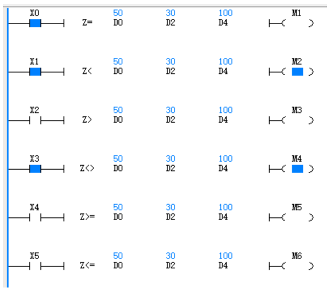
S3: Comparison value

- **Function Description**

The absolute value of the result after the subtraction of S1 and S2 is compared with the absolute value of S3, and the comparison result is connected in series with other nodes for driving the latter stage operation.

16-Bit Instructions	Turn-On Condition	Non-Conducting Condition
ANDZ =	|S1-S2|=|S3|	|S1-S2|<>|S3|
ANDZ <	|S1-S2|<|S3|	|S1-S2|>=|S3|
ANDZ >	|S1-S2|>|S3|	|S1-S2|<=|S3|
ANDZ <>	|S1-S2|<>|S3|	|S1-S2|=|S3|
ANDZ >=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ANDZ <=	|S1-S2|<=|S3|	|S1-S2|>|S3|

- **Example of use**

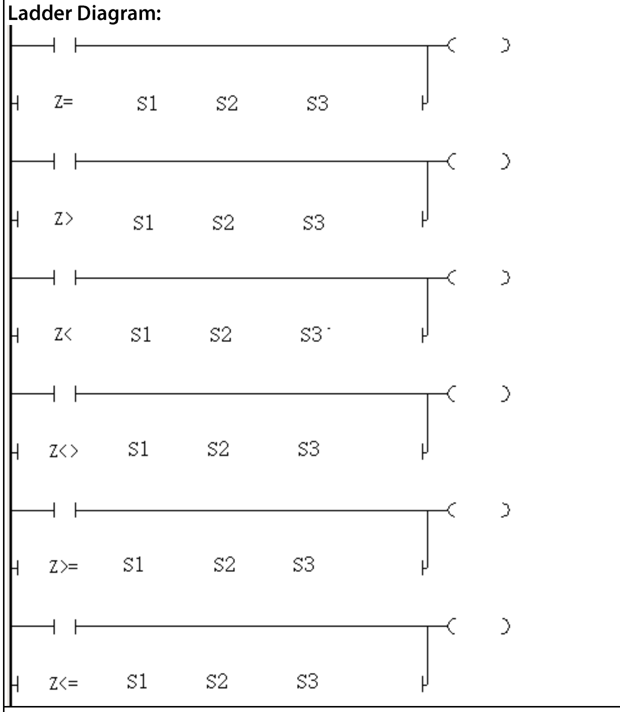


```
LD  X0
ANDZ=   D0 D2 D4
OUT  M1
LD      X1
ANDZ<   D0 D2 D4
OUT     M2
LD      X2
ANDZ>   D0 D2 D4
OUT     M3
LD      X3
ANDZ<>  D0 D2 D4
OUT     M4
LD      X4
ANDZ>=  D0 D2 D4
OUT     M5
LD      X5
ANDZ<=  D0 D2 D4
OUT     M6
```

The absolute value of the result after subtracting the D2 register value from the D0 register value is compared with the absolute value of the D4 register value, and the comparison result is connected in series with other nodes to determine the output state of the latter element.

6.16.12 ORZ (=, <, >, <>, >=, <=): Integer absolute value comparison contact or instruction

Ladder Diagram:



Applicable models	**VC-S VC-P**
Affect the flag	
Step size	**7**

Instruction list:

ORZ= (S1) (S2) (S3)

ORZ< (S1) (S2) (S3)

ORZ> (S1) (S2) (S3)

ORZ<> (S1) (S2) (S3)

ORZ>= (S1) (S2) (S3)

ORZ<= (S1) (S2) (S3)

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

- **Operand Description**

S1: Minuend

S2: Subtraction

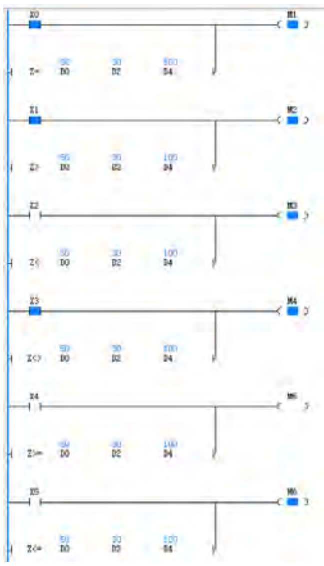
S3: Comparison value

- **Function Description**

The absolute value of the result after the subtraction of S1 and S2 is compared with the absolute value of S3, and the result of the comparison is connected in parallel with other nodes to drive the latter stage operation.

16-Bit Instructions	Turn-On Condition	Non-Conducting Condition
ORZ =	|S1-S2|=|S3|	|S1-S2|<>|S3|
ORZ <	|S1-S2|<|S3|	|S1-S2|>=|S3|
ORZ >	|S1-S2|>|S3|	|S1-S2|<=|S3|

- **Example of use**

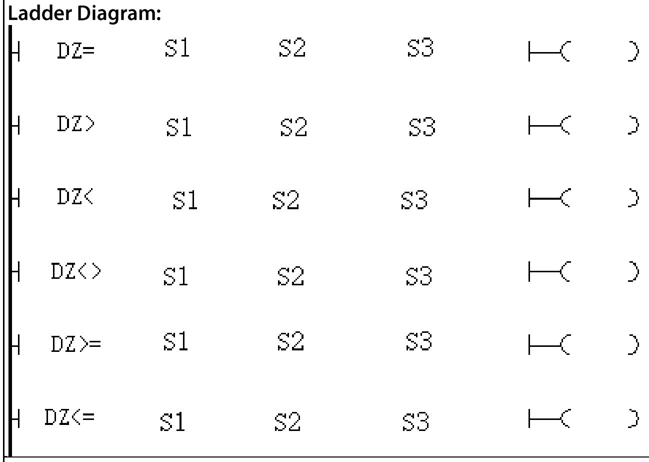


```
LD      X0
ORZ=    D0 D2 D4
OUT  M1
LD      X1
ORZ>    D0 D2 D4
OUT     M2
LD      X3
ORZ<    D0 D2 D4
OUT     M3
LD      X4
ORZ<>   D0 D2 D4
OUT     M4
LD      X4
ORZ>=   D0 D2 D4
OUT     M5
LD      X5
ORZ<=   D0 D2 D4
OUT     M6
```

ORZ <>	|S1-S2|<>|S3|	|S1-S2|=|S3|
ORZ >=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ORZ <=	|S1-S2|<=|S3|	|S1-S2|>|S3|

The absolute value of the result after subtracting the D2 register value from the D0 register value is compared with the absolute value of the D4 register value, and the result of the comparison is connected in parallel with other nodes to determine the output state of the latter element.

6.16.13 LDDZ (=, <, >, <>, >=, <=): Long integer absolute value comparison instruction

Ladder Diagram: 	Applicable models	VC-S VC-P
	Affect the flag	
Instruction list: LDDZ= (S1) (S2) (S3) LDDZ< (S1) (S2) (S3) LDDZ> (S1) (S2) (S3) LDDZ<> (S1) (S2) (S3) LDDZ>= (S1) (S2) (S3) LDDZ<= (S1) (S2) (S3)	Step size	7

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Minuend

S2: Subtraction

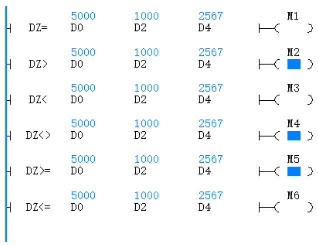
S3: Comparison value

- **Function Description**

Compare the absolute value of the result of the subtraction of S1 and S2 with the absolute value of S3, turn the contact ON or OFF according to the comparison result, and the node directly connected to the left bus.

32-Bit Instructions	Turn-On Condition	Non-Conducting Condition
LDDZ=	|S1-S2|=|S3|	|S1-S2|<>|S3|
LDDZ<	|S1-S2|<|S3|	|S1-S2|>=|S3|
LDDZ>	|S1-S2|>|S3|	|S1-S2|<=|S3|
LDDZ<>	|S1-S2|<>|S3|	|S1-S2|=|S3|
LDDZ>=	|S1-S2|>=|S3|	|S1-S2|<|S3|
LDDZ<=	|S1-S2|<=|S3|	|S1-S2|>|S3|

- **Example of use**

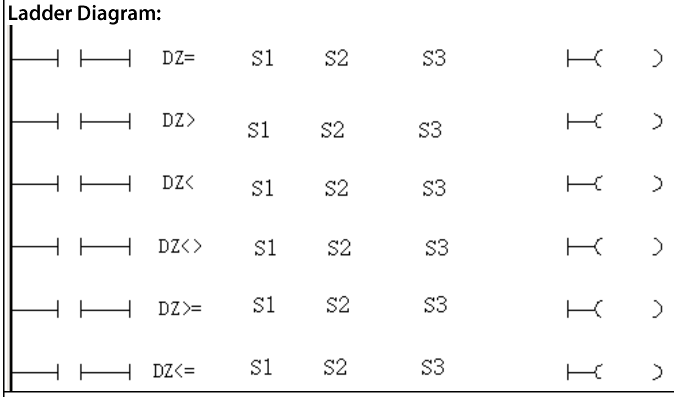


LDDZ= D0 D2 D4
OUT M1
LDDZ> D0 D2 D4
OUT M2
LDDZ< D0 D2 D4
OUT M3
LDDZ<> D0 D2 D4
OUT M4
LDDZ>= D0 D2 D4
OUT M5
LDDZ<= D0 D2 D4
OUT M6

Compare the absolute value of the (D0/D1) register value minus the (D2/D3) register value with the absolute value of the (D4/D5) register value, and turn the contact ON or OFF according to the comparison result.

6.16.14 ANDDZ (=, <, >, <>, >=, <=): Long integer absolute value comparison and instruction

Ladder Diagram:



Applicable models	**VC-S VC-P**
Affect the flag	

Instruction list:

ANDDZ= (S1) (S2) (S3)

ANDDZ< (S1) (S2) (S3)

ANDZ> (S1) (S2) (S3)

ANDDZ<> (S1) (S2) (S3)

ANDDZ>= (S1) (S2) (S3)

ANDDZ<= (S1) (S2) (S3)

Step size	**7**

Operand	Type	Applicable devices													Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- **Operand Description**

S1: Minuend

S2: Subtraction

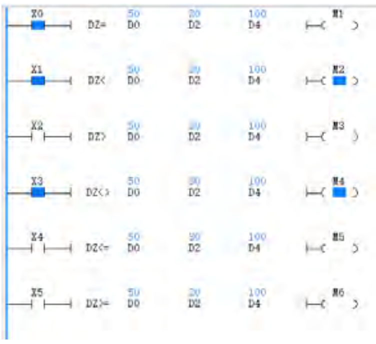
S3: Comparison value

- **Function Description**

The absolute value of the result after the subtraction of S1 and S2 is compared with the absolute value of S3, and the comparison result is connected in series with other nodes for driving the latter stage operation.

32-Bit Instructions	Turn-On Condition	Non-Conducting Condition
ANDDZ =	|S1-S2|=|S3|	|S1-S2|<>|S3|
ANDDZ <	|S1-S2|<|S3|	|S1-S2|>=|S3|
ANDDZ >	|S1-S2|>|S3|	|S1-S2|<=|S3|
ANDDZ <>	|S1-S2|<>|S3|	|S1-S2|=|S3|
ANDDZ >=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ANDDZ <=	|S1-S2|<=|S3|	|S1-S2|>|S3|

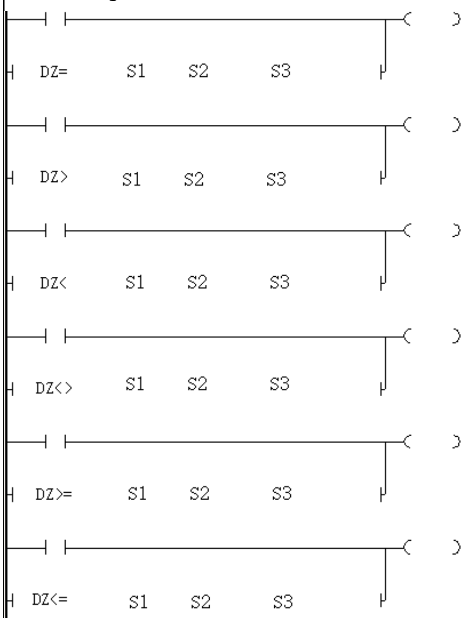
- **Example of use**



LD X0
ANDDZ= D0 D2 D4
OUT M1
LD X1
ANDDZ< D0 D2 D4
OUT M2
LD X2
ANDDZ> D0 D2 D4
OUT M3
LD X3
ANDDZ<> D0 D2 D4
OUT M4
LD X4
ANDDZ<= D0 D2
D4
OUT M5
LD X5
ANDDZ>= D0 D2
D4
OUT M6

The absolute value of the result after subtracting the (D2/D3) register value from the (D0/D1) register value is compared with the absolute value of the (D4/D5) register value.

6.16.15 ORDZ (=, <, >, <>, >=, <=): Long integer absolute value comparison or instruction

Ladder Diagram:	Applicable models	VC-S VC-P
	Affect the flag	
Instruction list: ORDZ= *(S1) (S2) (S3)* **ORDZ < *(S1) (S2) (S3)*** **ORDZ> *(S1) (S2) (S3)*** **ORDZ<> *(S1) (S2) (S3)*** **ORDZ>= *(S1) (S2) (S3)*** **ORDZ<= *(S1) (S2) (S3)***	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

- **Operand Description**

S1: Minuend

S2: Subtraction

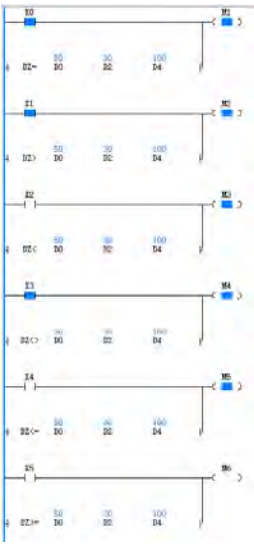
S3: Comparison value

- **Function Description**

The absolute value of the result after the subtraction of S1 and S2 is compared with the absolute value of S3, and the result of the comparison is connected in parallel with other nodes to drive the latter stage operation.

32-Bit Instructions	Turn-On Condition	Non-Conducting Condition
ORDZ =	|S1-S2|=|S3|	|S1-S2|<>|S3|
ORDZ <	|S1-S2|<|S3|	|S1-S2|>=|S3|
ORDZ >	|S1-S2|>|S3|	|S1-S2|<=|S3|

- **Example of use**



```
LD X0
ORDZ=    D0 D2 D4
OUT  M1
LD       X1
ORDZ>    D0 D2 D4
OUT      M2
LD       X3
ORDZ<    D0 D2 D4
OUT      M3
LD       X4
ORDZ<> D0 D2 D4
OUT      M4
LD       X4
ORDZ<= D0 D2 D4
OUT      M5
LD       X5
ORDZ>= D0 D2 D4
OUT      M6
```

ORDZ <>	|S1-S2|<>|S3|	|S1-S2|=|S3|
ORDZ >=	|S1-S2|>=|S3|	|S1-S2|<|S3|
ORDZ <=	|S1-S2|<=|S3|	|S1-S2|>|S3|

The absolute value of the result after subtracting the (D2/D3) register value from the (D0/D1) register value is compared with the absolute value of the (D4/D5) register value, and the result of the comparison is connected in parallel with other nodes to determine the output state of the latter element.

6.16.16 CMP: Integer Compare Set Instruction

Ladder Diagram: CMP *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: CMP ***(S1)*** ***(S2)*** ***(D)***	**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	BOOL			Y	M	S										

● **Operand Description**

S1: The data or device number to be the comparison value.

S2: The data or device number to be the comparison source.

D: The starting element number of the output result.

● **Function Description**

When the power flow is valid, execute the instruction and compare S1 and S2. According to its result (small, equal, large), make one of (D) (D+1) (D+2) ON.

● **Example of use**

LD m0
CMP 1000 2000 M3

6.16.17 LCMP: Long Integer Compare Set Instruction

Ladder Diagram: *LCMP* *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Instruction list: LCMP ***(S1)*** ***(S2)*** ***(D)***	**Step size**	**9**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V	Z	R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V	Z	R	√
D	BOOL			Y	M	S										

● **Operand Description**

S1: Comparison value 1.

S2: Comparison value 2.

D: The starting element number of the output result.

● **Function Description**

When the power flow is valid, execute the instruction and compare S1 and S2. According to its result (small, equal, large), make one of (D) (D+1) (D+2) ON.

● **Example of use**

LD m1
LCMP 200000 300000 M6

6.16.18 RCMP: Floating-Point Compare Set Instruction

Ladder Diagram:	Applicable models	VC-S VC-P
├─┤ ├──[RCMP (S1) (S2) (D)]	Affect the flag	
Instruction list: RCMP (S1) (S2) (D)	Step size	9

Operand	Type	Applicable devices													Index
S1	REAL	Constant							D					R	√
S2	REAL	Constant							D					R	√
D	BOOL			Y	M	S									

● **Operand Description**

S1: Comparison value 1.

S2: Comparison value 2.

D: The starting element number of the output result.

● **Function Description**

When the power flow is valid, execute the instruction, compare S1 and S2, according to the result (small, equal, large), make one of (D) (D+1) (D+2) ON.

● **Example of use**

```
M2
├─┤ ├──[ RCMP  500.3400  200.4000  Y7  ]
```

LD m2
RCMP 500.3400 200.4000 Y7

6.17 Batch Data Processing Instructions

6.17.1 BKADD: Addition of batch data

Ladder Diagram:	Applicable models	VC-S VC-P
├─┤ ├──[BKADD (S1) (S2) (D) (S3)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: BKADD (S1) (S2) (D) (S3)	Step size	9

Operand	Type	Applicable devices													Index
S1	INT								D	SD	C	T	V	R	√
S2	INT	Constant							D	SD	C	T	V	R	√
D	INT								D	SD	C	T	V	R	√
S3	INT	Constant							D				V	R	

● **Operand Description**

S1: The start number of the device that stores the data to perform the addition operation

S2: The Constant to perform the addition operation, or the start number of the device that stores the data to perform the addition operation

D: The start number of the device where the operation result is stored

S3: Number of data

● **Function Description**

1. When the power flow is valid, execute the instruction, add the 16-bit data of point s3 starting from S1 and the 16-bit data (BIN) of point S3 starting from S2, and save the operation result to point S3 starting from D.

2. 16-bit constants can be specified directly in S2,S2When it is a Constant, add the 16-bit data of point s3 starting from S1 and S2 in sequence, and save the operation result to point S3 starting from D.

● **Precautions**

When the operation result overflows, the Carry flag is not turned ON.

● **Example of use**

```
M1             1      30      31
├─┤ ├──[ BKADD  D10    D100    D1000   5   ]
```

LD M1

BKADD D10 D100 D1000 5

When M1=ON, the contents of the 5 units starting from D10 and the contents of the 5 units starting from D100 are added in turn, and the result is stored in the 5 units starting from D1000. D1000=D10+D100, D1001=D11+D101, D1004=D14+D104.

6.17.2 BKSUB: Subtraction of bulk data

Ladder Diagram:	Applicable models	VC-S VC-P
─┤ ├──[BKSUB (S1) (S2) (D) (S3)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: BKSUB *(S1) (S2)* **(D) (S3)**	Step size	9

Operand	Type	Applicable devices													Index
S1	INT								D	SD	C	T	V	R	√
S2	INT	Constant							D	SD	C	T	V	R	√
D	INT								D	SD	C	T	V	R	√
S3	INT	Constant							D				V	R	

● Operand Description

S1: The start number of the device that stores the data to perform the subtraction operation

S2: The Constant to perform the subtraction operation, or the start number of the device that stores the data to perform the subtraction operation

D: The start number of the device where the operation result is stored

S3: Number of data

● Function Description

1. When the power flow is valid, execute the instruction, which will start the S1SAfter subtracting the 16-bit data at 3 points and the 16-bit data (BIN) at point S3 starting from S2, the operation result is stored in point S3 starting from D.

2. You can directly specify a 16 bit Constant in S2. When S2 is a Constant, subtract the 16 bit data of S3 starting from S1 and S2 successively, and then save the operation result to S3 starting from D.

● Precautions

When the operation result overflows, the Carry flag is not turned ON.

● Example of use

LD M1

BKSUB D10 D100 D1000 5

When M1=ON, the contents of the 5 units starting from D10 and the contents of the 5 units starting from D100 are subtracted in turn, and the result is stored in the 5 units starting from D1000. D1000=D10-D100, D1001=D11-D101, ..., D1004=D14-D104.

6.17.3 BKCMP=,>,<,<>,<=,>=: Batch data comparison

Ladder Diagram:	Applicable models	VC-S VC-P
─┤ ├──[BKCMP= (S1) (S2) (D) (S3)] ─┤ ├──[BKCMP< (S1) (S2) (D) (S3)] ─┤ ├──[BKCMP> (S1) (S2) (D) (S3)] ─┤ ├──[BKCMP<= (S1) (S2) (D) (S3)] ─┤ ├──[BKCMP>= (S1) (S2) (D) (S3)] ─┤ ├──[BKCMP<> (S1) (S2) (D) (S3)]	Affect the flag	Zero flag Carry flag Borrow flag
Command list: BKCMP=,>,<,<>,<=,>= *(S1) (S2) (D)* (S3)	Step size	9

Operand	Type	Applicable devices													Index
S1	INT	Constant							D	SD	C	T	V	R	√
S2	INT								D	SD	C	T	V	R	√
D	BOOL		Y	M	S	LM	SM								
S3	INT	Constant							D				V	R	

● **Operand Description**

S1: Device start number of comparison value or stored value data

S2: The start number of the device where the comparison source data is stored

D: The start number of the device where the comparison result is stored

S3: Number of data

● **Function Description**

1. After comparing the 16-bit data (BIN) of the S3 point starting from S1 with the 16-bit data (BIN) of the S3 point starting from S2, the operation result is stored in the S3 point starting from D.

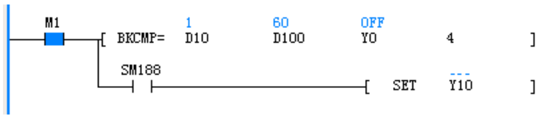
2. A 16-bit Constant can be specified directly in S1. When S1 is a Constant, the 16-bit data of s3 points starting from S1 and S2 are compared in turn, and the operation result is saved to the S3 point starting from D.

3. When the comparison results of the S3 points starting from D are all ON, the data block comparison setting Sign (SM188) is set.

● **Precautions**

When the operation result overflows, the Carry flag is not turned ON.

● **Example of use**



LD M1

BKCMP= D10 D100 Y0 4

LD SM188

SET Y10

When M1=ON, compare the contents of the 4 units starting from D10 with the contents of the 4 units starting from D100, and the result is stored in the 4 units starting from Y0. Also, when the comparison results are all ON, Y10 turns ON.

6.18 Data Sheet Instructions

6.18.1 LIMIT:Upper and lower limit control

Ladder Diagram: LIMIT *(S1)* *(S2)* *(S3)* *(D)*								**Applicable models**		**VC-S VC-P**					
								Affect the flag		**Zero flag Carry flag Borrow flag**					
Instruction list: LIMIT *(S1)* *(S2)* *(S3)* **(D)**								**Step size**		**9**					
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S3	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

● **Operand Description**

S1: Lower limit value

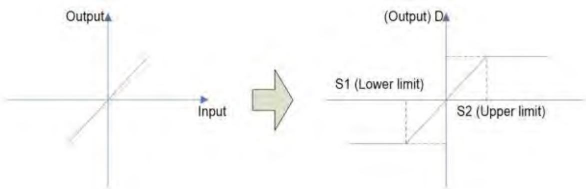
S2: Upper limit value

S3: Input value that needs to be controlled by the upper and lower limits

D: The start number of the device that stores the output value that has passed the upper and lower limit control

● **Function Description**

By judging whether the input value specified in S3 is within the range of the upper and lower limit values specified by S1 and S2, the control is stored in D. When S3<S1, D=S1; When S3>S2, D=S2; When S1<=S3<=S2, D=S2.



Example of use

LIMIT D0 D10 D100 D1000

LD M1

When M1=ON, the limit control of D0~D10 is performed on the content of D100 unit, and the result is stored in D1000.D0(10)<=D100(30)<=D10(100),D1000=30.

6.18.2 DBAND:Dead zone control

Ladder Diagram: ├─┤├──[DBAND (S1) (S2) (S3) (D)]									Applicable models		VC-S VC-P				
									Affect the flag		Zero flag Carry flag Borrow flag				
Instruction list: DBAND (S1) (S2) (S3) (D)									Step size		9				
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S3	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

● **Operand Description**

S1: Dead zone lower limit value

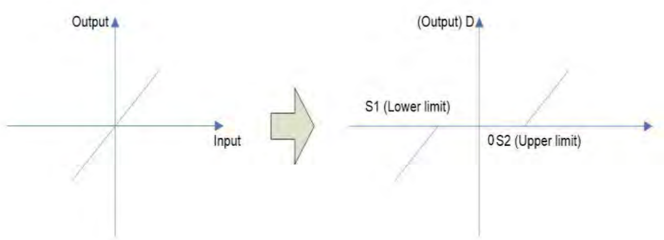
S2: Dead zone upper limit value

S3: Input value to be controlled by dead band

D: The start number of the device that stores the output value that has passed the dead-band control

● **Function Description**

By judging whether the input value specified in S3 is within the dead zone range specified by S1 and S2, the control is saved in D.When S3<S1, D=S3-S1; When S3>S2, D=S3-S2; When S1<=S3<=S2, D=0.



● **Example of use**

LD M1

DBAND D0 D10 D100 D1000

When M1=ON, the dead zone control of D0~D10 is performed on the content of D100 unit, and the result is stored in D1000.D0 (-100)<D100(30)<D10(100),D1000=0;

6.18.3 ZONE: Zone Control

Ladder Diagram: ├─┤├──[ZONE (S1) (S2) (S3) (D)]		Applicable models	VC-S VC-P	
		Affect the flag	Zero flag Carry flag Borrow flag	
Command list: ZONE (S1) (S2) (S3) (D)		Step size	9	
Operand	Type	Applicable devices		Index

S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S3	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

● **Operand Description**

S1: Negative offset value added to the input value

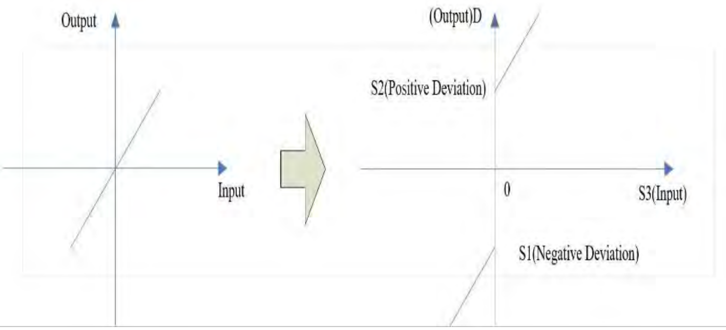
S2: Positive deviation value to add to the input value

S3: Input value to be controlled by zone

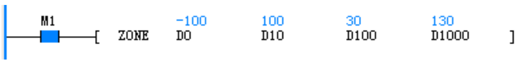
D: The start number of the device that saves the output value that has passed the area control

● **Function Description**

By judging the input value specified in S3 plus the deviation value specified in S1 or S2, the control is saved in D When S3<0, D=S3+S1; When S3>0, D=S3+S2; When S3=0, D=0.



● **Example of use**



LD M1

ZONE D0 D10 D100 D1000

When M1=ON, the area control of D0~D10 is performed on the content of D100 unit, and the result is stored in D1000.D100(30)>0,D1000=D100(30)+D10(100), D1000=130.

6.18.4 SCL:Fixed coordinates

Ladder Diagram: SCL *(S1)* *(S2)* *(D)*]								**Applicable models**	**VC-S VC-P**						
								Affect the flag	**Zero flag Carry flag Borrow flag**						
Instruction list: SCL ***(S1) (S2) (S3)*** **(D)**								**Step size**	**7**						
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S2	INT								D				V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

● **Operand Description**

S1: The input value to execute the fixed coordinate or the device number to save the input value

S2: Start number of conversion table device for fixed coordinates

D: The device number that stores the output value controlled by the fixed coordinate

● **Function Description**

1. According to the specified conversion characteristics, coordinate the input value specified by S1, and then save it to the device number specified by D.

2. The conversion for fixed coordinates is performed according to the data table stored in the device designated by S2. However, when the output data is not an integer value, the first decimal place is rounded and output.

3. The fixed coordinates are set with the conversion table:

Coordinate points		S2
point 1	X coordinate	S2+1

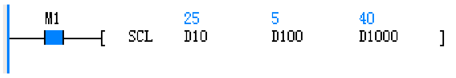
Coordinate points		S2
	Y coordinate	S2+2
point 2	X coordinate	S2+3
	Y coordinate	S2+4
…	…	…
point n (last)	X coordinate	S2+2n-1
	y coordinate	S2+2n

● **Precautions**

1. The data of data table X should be arranged in ascending order. If only part of the data is not in ascending order, and the detection starts from the low order, the operation before this part will still be executed;

2. S1 must be within the range set by the data table;

● **Example of use**

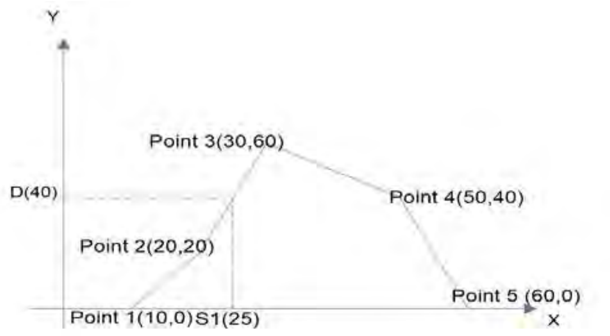


LD M1

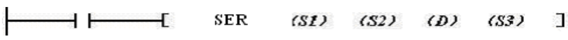
SCL D10 D100 D1000

When M1=ON, the content of unit D10 is fixed and the result is stored in D1000.

Coordinate points		D100	5
point 1	X coordinate	D101	10
	Y coordinate	D102	0
point 2	X coordinate	D103	20
	Y coordinate	D104	20
point 3	X coordinate	D105	30
	y coordinate	D106	60
point 4	X coordinate	D107	50
	y coordinate	D108	40
point 5	X coordinate	D109	60
	y coordinate	D110	0



6.18.5 SER: Data retrieval

Ladder Diagram:	Applicable models	VC-S VC-P
 SER (S1) (S2) (D) (S3)	Affect the flag	Zero flag Carry flag Borrow flag
Command list: SER *(S1)* *(S2)* *(S3)* (D)	Step size	9

Operand	Type	Applicable devices													Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√
S3	INT	Constant							D				V	R	

● **Operand Description**

S1: Search the same data, the maximum value, and the minimum value of the head device number

S2: Retrieves the same data, the reference value of the maximum value, the minimum value, or the save destination device number

D: After retrieving the same data, the maximum value and the minimum value, the start device number of these numbers is stored

S3: Retrieve the same data, the maximum value, the minimum value(1≤S3≤256)

● **Function Description**

1. Retrieve the S3 data starting with S1, retrieve the same data as the data of S2, and save the result in D-D+4.

2.When there is the same data, among the 5 devices starting from D, save the number of the same data, the initial value/final value, the position of the minimum value and the maximum value.

3.When the same data does not exist, the first 3 soft elements store 0, and the other two are the same as above.

● **Example of use**

LD M1

SER D0 D10 D100 D1000 8

When M1=ON, the contents of 8 units starting from D10 are retrieved, and the retrieval results are stored in 5 units starting from D1000.

Retrieved element S1	Numerical value	Compare element value S2	Data location	Search result D	Numerical value
D10	100	100	0	D1000	3
D11	78		1	D1001	0
D12	92		2	D1002	7
D13	100		3	D1003	5
D14	110		4	D1004	6
D15	-20		5		
D16	145		6		
D17	100		7		

6.19 String Command

6.19.1 STRADD: String Combination

Ladder Diagram: [STRADD (S1) (D) (S2)]	**Applicable models**	**VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Instruction list: STRADD *(S1) (S2) (D)*	**Step size**	**7**

Operand	Type	Applicable devices												Index
S1	INT	String	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
S2	INT	String	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

● **Operand Description**

S1: First string unit

S2: Second string unit

D: String storage unit after concatenation

● **Function Description**

1. When the power flow is valid, connect the string units starting with S1 and S2, and save them to the device starting with D;

2. The combination of strings refers to connecting the first character of the S2 unit string to the last character of the S1 unit string ignoring the end marker of the S1 unit string;

3. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H';

4. When the number of characters after connection is odd, add '00H' to the high byte of the last character device, and add '0000H' to the next element of the last character device when it is even;

● **Precautions**

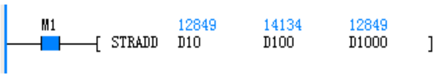
1. When S1 and S2 specify a string, a maximum of 32 characters are allowed, and commas and double quotation marks are delimiters in the host computer software, so the characters cannot be recognized by the host computer software;

2. If both S1 and S2 store '00H', then directly add '0000H' to D;

3. When S1 and D or S2 and D's string unit device addresses overlap, an "instruction operand value is illegal" error will be reported;

4. When there is no '00H' in the corresponding soft element range of the string unit starting from S1 or S2, the error "Instruction operand element number range exceeds" is reported;

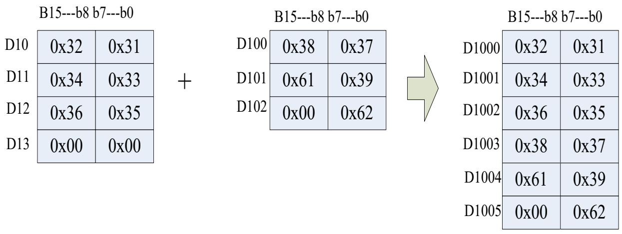
● **Example of use**



LD M1

STRADD D10 D100 D1000

When M1=ON, connect the string unit starting from D10 with the string unit starting from D100, and the result is stored in the unit starting from D1000.



6.19.2 STRLEN: Detect string length

Ladder Diagram: ─┤├── [STRLEN (S) (D)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Instruction list: STRLEN (S) (D)	Step size	5

Operand	Type	Applicable devices													Index
S	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

● **Operand Description**

S: String unit

D: String unit length

● **Function Description**

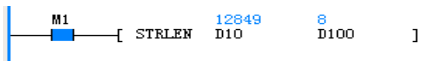
1. When the power flow is valid, the length of the S unit string is detected, and the value is stored in D

2. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H'

● **Precautions**

When there is no '00H' in the corresponding soft element range of the string unit starting from S, the error "Instruction operand element number range exceeds" will be reported;

● **Example of use**



LD M1

STRLEN D10 D100

When M1=ON, the length of the character string unit starting from D10 is detected, and the result is stored in D100.

6.19.3 STRRIGHT: Start reading from the right side of the string

Ladder Diagram: ─┤├── [STRRIGHT (S1) (D) (S2)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Instruction list: STRIGHT (S1) (D) (S2)	Step size	7

Operand	Type	Applicable devices													Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√

D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√
S2	INT	Constant							D				V	R	

● **Operand Description**

S1: String unit

D: Save the extracted string unit

S2: Number of characters to take out

● **Function Description**

1. When the power flow is valid, start from the last valid character of the character string of the S1 unit (not counting '00H'), take out the S2 characters, and save them in the device starting from D;

2. When S2 is equal to zero, "00H" is stored in the D soft element;

3. When the number of characters taken out is odd, add '00H' to the high byte of the soft element that holds the last character, and add '0000H' to the next element of the soft element that holds the last character when it is an even number;

4. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H'

● **Precautions**

1. When there is no '00H' in the corresponding soft element range of the string unit starting from S1, the error "Instruction operand element number range exceeds" will be reported;

2. S2 is greater than or equal to 0;

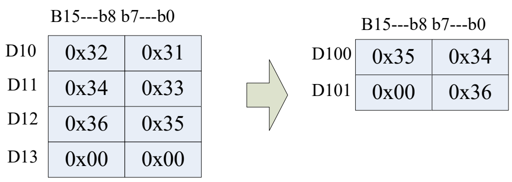
3. S2 must be less than or equal to the number of characters in the string unit of S1;

● **Example of use**

LD M1

STRRIGHT D10 D100 3

When M1=ON, start from the right side of the string unit starting from D10, take out 3 characters and save them in the unit starting from D100.



6.19.4 STRLEFT: Start reading from the left side of the string

Ladder Diagram:

Applicable models	**VC-S VC-P**
Affect the flag	**Zero flag Carry flag Borrow flag**
Step size	**7**

Instruction list: STRLEFT *(S1) (D) (S2)*

Operand	Type	Applicable devices													Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√
S2	INT	Constant							D				V	R	√

● **Operand Description**

S1: String unit

D: Save the extracted string unit

S2: Number of characters to take out

● **Function Description**

1. When the power flow is valid, start from the left side of the S1 string unit, take out S2 characters to the right, and save them in the soft element starting from D;

2. When S2 is equal to zero, "00H" is stored in the D soft element;

3. When the number of characters taken out is odd, add '00H' to the high byte of the soft

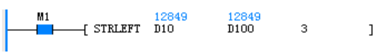
● **Precautions**

1. When there is no '00H' in the corresponding soft element range of the string unit starting from S1, the error "Instruction operand element number range exceeds" will be reported;

2. S2 is greater than or equal to 0

3. S2 must be less than or equal to the number of characters in the S1 unit

● **Example of use**



LD M1

STRLEFT D10 D100 3

When M1=ON, start from the left side of the string unit starting from D10, take out 3 characters and save them in the unit starting from D100.

element that holds the last character, and add '0000H' to the next element of the soft element that holds the last character when it is an even number;

4. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H'

	B15---b8	b7---b0
D10	0x32	0x31
D11	0x34	0x33
D12	0x36	0x35
D13	0x00	0x00

→

	B15---b8	b7---b0
D100	0x32	0x31
D101	0x00	0x33

6.19.5 STRMIDR: Arbitrary read from a string

Ladder Diagram: ├──┤ ├──[STRMIDR (S1) (D) (S2)]						Applicable models		VC-S VC-P						
						Affect the flag		Zero flag Carry flag Borrow flag						
Instruction list: STRMIDR *(S1)* *(D)* *(S2)*						Step size		7						
Operand	Type	Applicable devices												Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√
S2	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√

● **Operand Description**

S1: String unit

D: Extracted string unit

S2: The starting position of the string to be retrieved

S2+1 Number of characters to be fetched n

● **Function Description**

1. When the power flow is valid, for the S1 string unit, take out n character data starting from the S2 character, and save it in the soft element starting from D;

2. When the number of characters taken out is odd, add '00H' to the high byte of the soft element that holds the last character, and add '0000H' to the next element of the soft element that holds the last character when it is an even number;

3. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H';

4. When n is 0, no processing is performed;

5. When n is -1, all the character data of the S1 string unit are taken out and stored in the device starting from D

● **Precautions**

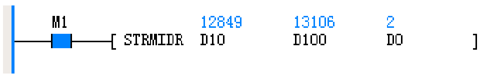
1. S2 must be less than or equal to the number of characters in the string unit of S1;

2. n is greater than -2

3. S2 is greater than or equal to 1

4. When there is no '00H' in the corresponding soft element range of the string unit starting from S1, the error "Instruction operand element number range exceeds" will be reported;

● **Example of use**



LD M1

STRMIDR D10 D100 D0

When M1=ON, D1 (D1=3) data starting from D0 (D0=2) of the string unit starting from D10 is read out and stored in the unit starting from D100.

	B15---b8	b7---b0
D10	0x32	0x31
D11	0x34	0x33
D12	0x36	0x35
D13	0x00	0x00

→

	B15---b8	b7---b0
D100	0x33	0x32
D101	0x00	0x34

6.19.6 STRMIDW: Replace arbitrary from string

Ladder Diagram: ├──┤ ├──[STRMIDW (S1) (D) (S2)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Instruction list: STRMIDW *(S1)* *(D)* *(S2)*	Step size	7

Operand	Type	Applicable devices													Index
S1	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√
S2	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√

● **Operand Description**

S1: The string unit to replace

D: The string unit to be replaced

S2: The starting position of the replacement

S2+1number of characters to replace n

● **Function Description**

1. When the power flow is valid, use the n characters of the S1 string unit to replace the n character data starting from the S2 character in the D string unit;

2. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H';

3. When n is 0, no processing is performed;

4. When n is -1, the contents up to the last character data designated by S1 are stored after the device designated by D

● **Precautions**

1. S2 is less than or equal to the number of characters in the string unit of S1;

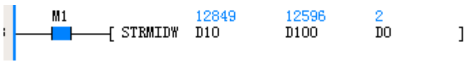
2. n is greater than -2

3. S2 is greater than or equal to 1

4. When the number of replaced characters exceeds the last character of the string unit starting with D, save the data up to the last character

5. When there is no '00H' in the corresponding soft element range of the string unit starting from S1 and D, the error "Instruction operand element number range exceeds" will be reported;

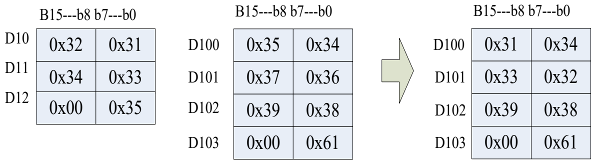
● **Example of use**



LD M1

STRMIDW D10 D100 D0

When M1=ON, replace D1 (D1=3) after the D0 (D0=2) character of the string unit starting from D100 with the first D1 (D1=3) characters of the string unit starting from D10 characters.



6.19.7 STRINSTR: String retrieval

Ladder Diagram: ├──┤ ├──[STRINSTR (S1) (S2) (D) (S3)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Instruction list: STRINSTR *(S1)* *(S2)* *(D)* (S3)	Step size	9

Operand	Type	Applicable devices													Index
S1	INT	string							D	SD	C	T	V	R	√
S2	INT								D	SD	C	T	V	R	√
D	INT								D	SD	C	T	V	R	√
S3	INT	Constant							D				V	R	

● **Operand Description**

S1: String unit to retrieve

S2: Search source

D: Search Results

S3: Search start position

● **Function Description**

1. When the power flow is valid, starting from the S3 character of the S2 character string unit, retrieve the same character string as the S1 character string unit, and save the character string position information of the retrieved result in D;

2. When there is no consistent string, save "0" in D;

3. When the position S3 to start the search is a negative number or "0", no processing is performed;

4. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H';

● **Precautions**

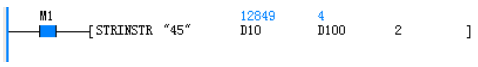
1. When there is no '00H' in the corresponding soft element range of the string unit starting from S1 and S2, the error "Instruction operand element number range exceeds" is reported;

2. S3 is less than or equal to the number of characters in the string unit of S2;

3. When S1 specifies a string, a maximum of 32 characters are allowed, and commas and double quotation marks represent delimiters in the host computer software, so this character cannot be recognized by the host computer software;

4. When S1 is an empty string ('00H'), the detection result is the position of the string unit '00H' of S2 (if S2 is an even number of characters, it is the first '00H' position);

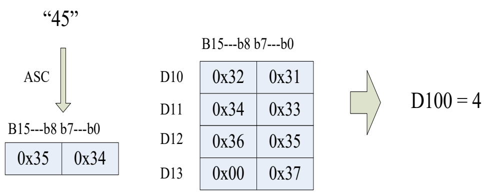
● **Example of use**



LD M1

STRINSTR "45" D10 D100 2

When M1=ON, starting from the second character of the string unit starting from D10, search for the same character as "45", and the result is stored in the unit of D100.



6.19.8 STRMOV: String transmission

Ladder Diagram: ─┤├──[STRMOV (S) (D)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Instruction list: STRMOV (S) (D)	Step size	5

Operand	Type	Applicable devices													Index
S	INT	string	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C	T	V	R	√

- **Operand Description**

S: Source string unit

D: Destination unit

- **Function Description**

1. Transfer all the data of the S string unit, including '00H', to the element unit starting with D;

2. The valid data of the character string unit is the data from the specified device of the character string unit to the detection of the first '00H';

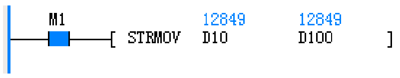
- **Precautions**

1. When there is no '00H' in the corresponding soft element range of the string unit starting from S, it will report "The range of the instruction operand element number exceeds";

2. When the number of characters in the S string unit is an even number, '00H' is stored in the low byte, and the high and low bytes of the corresponding position in D are stored in '00H';

3. When S1 specifies a string, a maximum of 32 characters are allowed, and commas and double quotation marks are delimiters in the host computer software, so the characters cannot be recognized by the host computer software;

- **Example of use**



LD M1

STRMOV D10 D100

When M1=ON, the character string data starting at D10 is transferred to the unit starting at D100.

	B15---b8	b7---b0
D10	0x32	0x31
D11	0x34	0x33
D12	0x36	0x35
D13	0x00	0x00

⇨

	B15---b8	b7---b0
D100	0x32	0x31
D101	0x34	0x33
D102	0x36	0x35
D103	0x00	0x00

6.20 Positioning Commands and Interpolation

6.20.1 ZRN: Origin return command

Instruction type	Command name	Reference chapter
Hhigh speed command	ZRN origin return command	🕮For detailed instructions, please refer to Chapter 11 11.2.1

6.20.2 DSZR: Origin return command with DOG search

Instruction type	Command name	Reference chapter
High speed command	DSZR with DOG search origin return command	🕮For detailed instructions, please refer to Chapter 11 11.2.2

6.20.3 DRVI: Relative Position Control Instruction

Instruction type	Command name	Reference chapter
High speed command	DRVI: Relative Position Control Instruction	🕮For detailed instructions, please refer to Chapter 11 11.2.3

6.20.4 DRVA: Absolute position control command

Instruction type	Command name	Reference chapter
High speed command	DRVA absolute position control instruction	🕮For detailed instructions, please refer to Chapter 11 11.2.4

6.20.5 PLS: Multi-speed pulse output command

Instruction type	Command name	Reference chapter
High speed command	PLS multi-speed pulse output command	🕮For detailed instructions, please refer to Chapter 11 11.2.7

6.20.6 DVIT: interrupt fixed-length instruction

Instruction type	Command name	Reference chapter
High speed command	DVIT interrupt fixed length	🕮For detailed instructions, please refer to Chapter 11 11.2.8

6.20.7 DPTI: maximum fixed-length interrupt positioning instruction

Instruction type	Command name	Reference chapter
High speed command	DPTI maximum fixed-length interrupt positioning instruction	🕮For detailed instructions, please refer to Chapter 11 11.2.9

6.20.8 STOPDV: pulse output stop command

Instruction type	Command name	Reference chapter
High speed command	STOPDV pulse output stop command	🕮For detailed instructions, please refer to Chapter 11 11.2.10

6.20.9 PLSV: Variable speed pulse output command

Instruction type	Command name	Reference chapter
High speed command	PLSV variable speed pulse output command	🕮For detailed instructions, please refer to Chapter 11 11.3.2

6.20.10 LIN: Linear path interpolation command

Instruction type	Command name	Reference chapter
High speed command	LIN linear path interpolation command	🕮For detailed instructions, please refer to Chapter 11 11.4.1

6.20.11 CW: Clockwise arc path interpolation command

Instruction type	Command name	Reference chapter
High speed command	CW Clockwise arc path interpolation command	🕮For detailed instructions, please refer to Chapter 11 11.4.2

6.20.12 CCW: Counterclockwise circular arc path interpolation command

Instruction type	Command name	Reference chapter
High speed command	CCW counterclockwise arc path interpolation command	🕮For detailed instructions, please refer to Chapter 11 11.4.3

6.21 Data Processing Instructions

6.21.1 MEAN: Average command

Ladder Diagram: MEAN (S1) (D) (S2)	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: MEAN *(S1) (D) (S2)*	Step size	7

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T		R	√
S2	INT	Constant							D					R	√
D	INT			KnY	KnM	KnS	KnLM		D	SD	C			R	√

● **Operand Description**

S1: The starting word device number where the desired average value data is stored

S2: Average number of data (1~64)

D: Word device number to store the acquired average value data

● **Function Description**

1. The average value of S2 16-bit data starting from S1 is stored in D, and the remainder is rounded off.

● **Example of use**

LD M1

MEAN D0 D10 4

When M1=ON, find the average value of 4 unit data starting from D0, and save it in D10. When D0=32, D1=10, D2=15, D3=-14, D10=10.

6.21.2 WTOB: Data separation instruction in byte units

Ladder Diagram: WTOB (S1) (D) (S2)	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: WTOB *(S1) (D) (S2)*	Step size	7

Operand	Type	Applicable devices													Index
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

● **Operand Description**

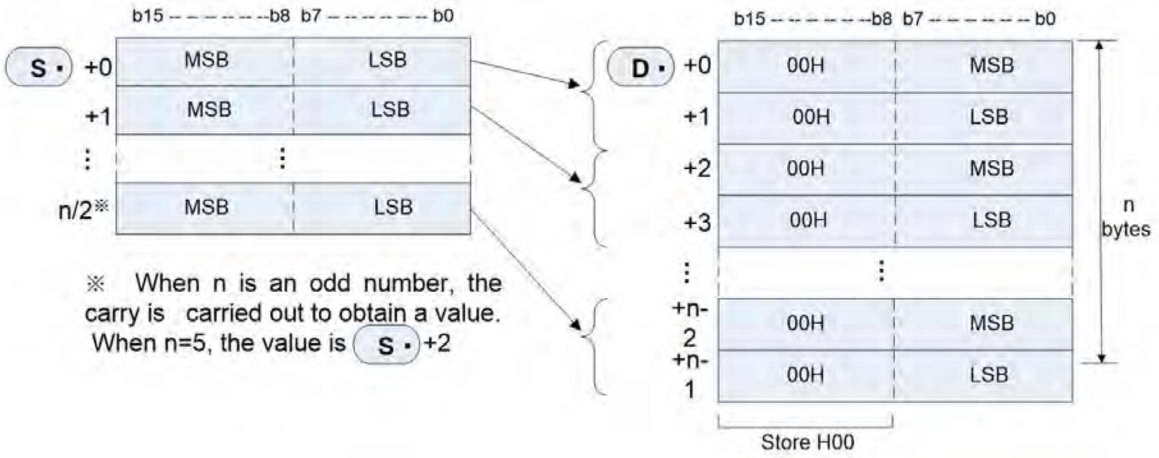
S1: The start number of the device that stores the data to be separated in byte units

S2: The number of byte data to be separated (S2≥0)

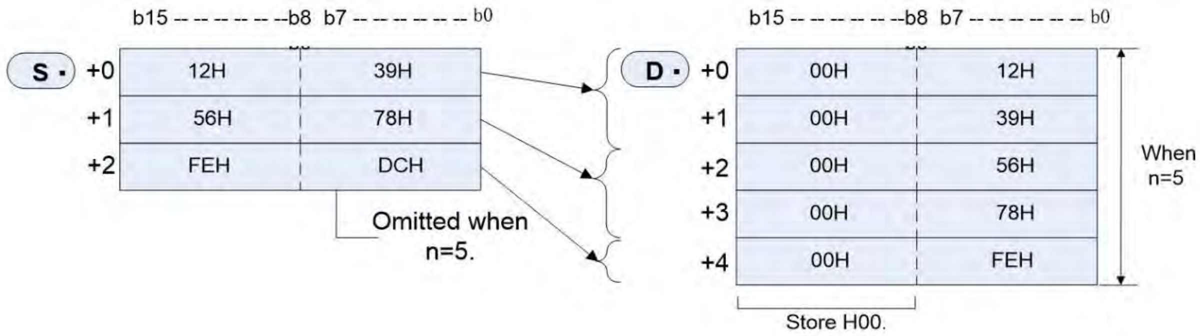
D: The start number of the device that saves the result that has been separated in byte units

● **Function Description**

1.The 16-bit data stored in the S2/2 soft elements starting with S1 is separated into S2 bytes, and stored in the low byte of the S2 soft elements starting with D, and the high byte is cleared.



2. When S2 is an odd number, in the last data of the separation source, only the high byte (8 bits) is the object data. For example, when n=5, the data of the low byte of S~S+2 is stored in D~D+4.



3.When S2=0, the instruction is not executed.

4. Source and destination operands cannot overlap.

● **Example of use**

LD M1

WTOB D0 D10 6

When M1=ON, divide the data of 3 units starting from D0 into 6 units according to the high and low bytes, and save them in the 6 units starting from D10. When D0=0x102, D1=0x304, D2=0x506, D10=0x01, D11=0x02, D12=0x03, D13=0x04, D14=0x05, D15=0x06.

6.21.3 BTOW:Data combination instruction in byte unit

Ladder Diagram: ─[BTOW (S1) (D) (S2)]	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: BTOW *(S1)* *(D)* *(S2)*	Step size	7

Operand	Type	Applicable devices													Index
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

● **Operand Description**

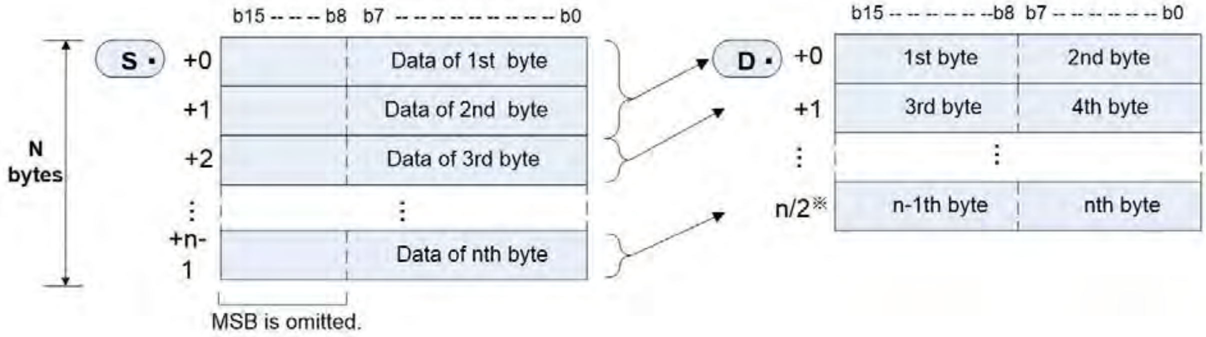
S1: The start number of the device that stores the data to be combined in byte units

S2: The number of byte data to be combined (S2≥0)

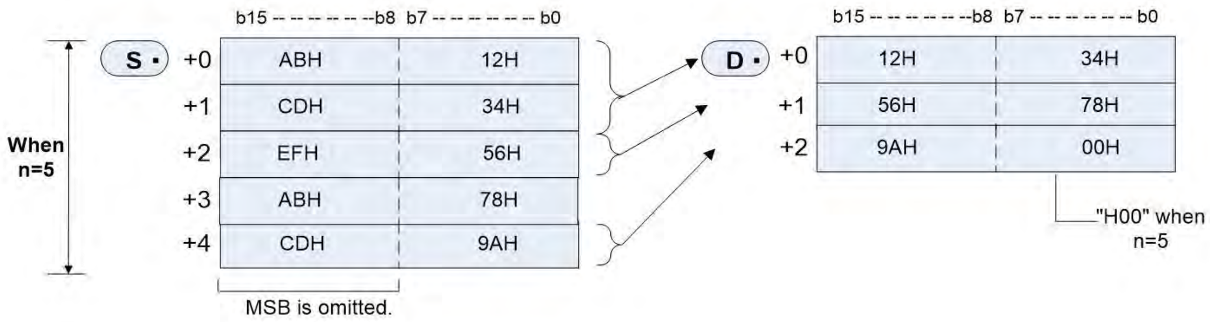
D: The start number of the device that saves the result of combining in byte units

● **Function Description**

1.The 16-bit data after combining the low bytes (8 bits) of the S2 16-bit data starting from S1 is stored in the S2/2 soft elements starting with D. The high byte (after S1) of the combined 16-bit data of the source (8 bits) is ignored.



2.When S2 is odd, the last combined low byte is cleared.



3.When S2=0, the instruction is not executed.

4. Source and destination operands cannot overlap.

● **Example of use**

LD M1

BTOW D0 D10 6

When M1=ON, combine the 6 unit data starting from D0 to generate 3 unit data, which are stored in the 3 units starting from D10. When D0=0x01, D1=0x02, D2=0x03, D3=0x04, D4=0x05, D5=0x06, D10=0x102, D11=0x304, D12=0x506.

6.21.4 UNI: 4-bit combination instruction for 16-bit data

Ladder Diagram: UNI (S1) (D) (S2)	**Applicable models**	**VC-S VC-P**
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: UNI *(S1) (D) (S2)*	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

● **Operand Description**

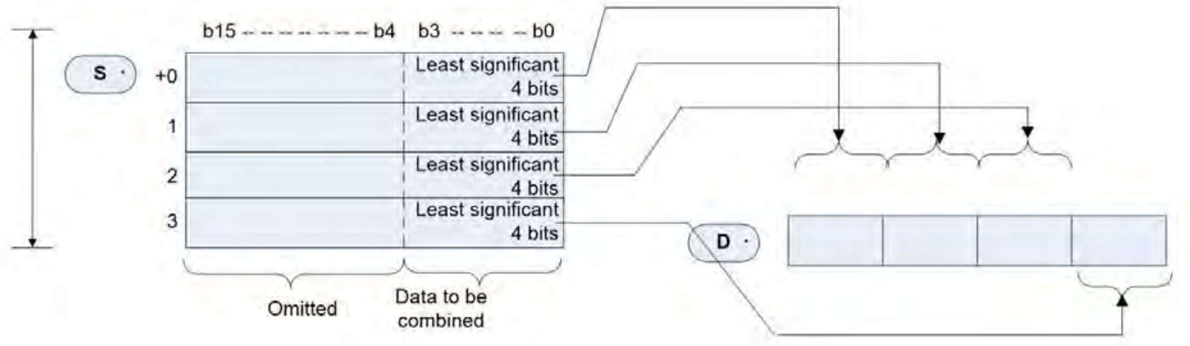
S1: The start number of the device that stores the data to be combined

S2: Number of combinations (0-4. No processing when S2=0)

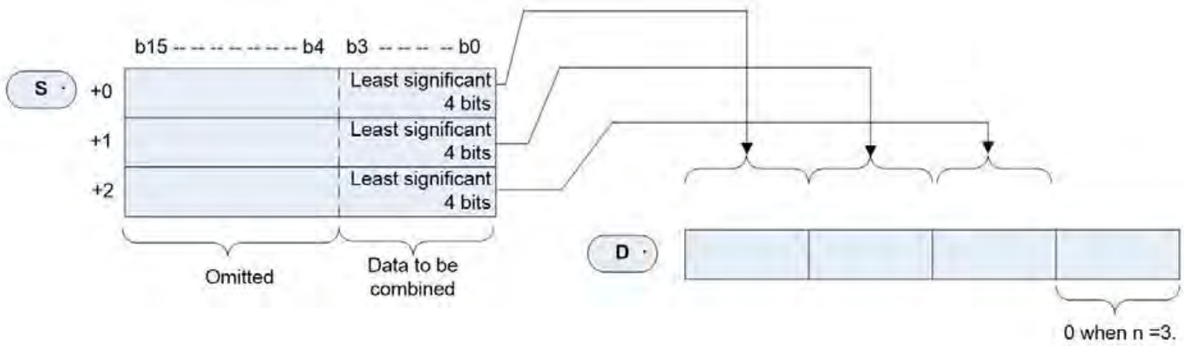
D: The device number where the combined data is stored

● **Function Description**

1.Save the 16-bit data of the S2 point starting with S1 to the S2 point device starting with D.



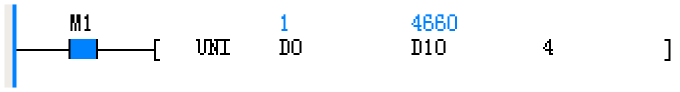
2.When S2 is from 1 to 3, the one digit of the lower digit {4×(4-S2)} of D is zero.



3.Specify 1-4 in S2, and when S2=0, the instruction is not executed.

4. Source and destination operands cannot overlap.

● **Example of use**



LD M1

UNI D0 D10 4

When M1=ON, combine the lower 4 bits of the 4 unit data starting from D0 and save it in D10. When D0=0x01, D1=0x02, D2=0x03 D3=0x04, D10=0x1234.

6.21.5 DIS: 4 bit separate instruction of 16-bit data

Ladder Diagram: DIS (S1) (D) (S2)	Applicable models	VC-S VC-P
	Affect the flag	**Zero flag Carry flag Borrow flag**
Command list: DIS *(S1) (D) (S2)*	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

● **Operand Description**

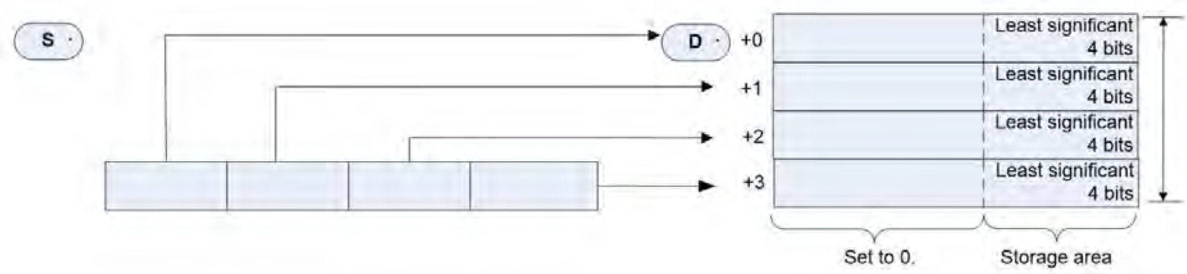
S1: The start number of the device that stores the data to be separated

S2: Number of separations (0-4. No processing when S2=0)

D: The device number where the separated data is stored

● **Function Description**

1.Save the S2 16-bit data starting with S1 into the S2 soft elements starting with D.



2.The valid range of S2 is 1-4, and the rest of the data does not execute the instruction.

3.The upper 12 bits of the S2 soft elements starting from D are cleared.

4. Source and destination operands cannot overlap.

● **Example of use**

LD M1

DIS D0 D10 4

When M1=ON, the D0 unit data is separated every 4 bits and stored in the 4 units starting from D10. When D0=0x1234, D10=0x01, D11=0x02, D12=0x03, D13=0x04.

6.21.6 ANS:Signal alarm set instruction

Ladder Diagram: ANS (S1) (S2) (D)	Applicable models	VC-S VC-P
	Affect the flag	Zero flag Carry flag Borrow flag
Command list: ANS (S1) (S2)(D)	**Step size**	7

Operand	Type	Applicable devices													Index
S1	INT											T			√
S2	INT	Constant							D					R	√
D	BOOL			S											√

● **Operand Description**

S1:Timing timer number for judging time, only applicable to 100ms timer, T0-T209

S2: Judging time data (1—32767)

D: Set annunciator device, S900-S999

● **Function Description**

1.When the power flow duration is greater than S2, D is set; when the command power flow duration is less than S2, the timer S1 is reset and D is not set; the power flow is invalid, and S1 is reset.

Address number	Name	Function
SM400	Signal alarm is valid	After SM400 is turned ON, the following SM401 and SD401 work
SM401	Signal alarm action	Any action in the state S900-S999, SM401 is ON
SD401	On state minimum number	Save the minimum number of actions in S900-S999

● **Example of use**

LD M0

ANS T0 100 S901

When the power flow is valid, if the power flow is not interrupted within 10 seconds, S901 is set.

6.21.7 ANR:Signal alarm reset instruction

Ladder Diagram: ⊣ ⊢[ANR]		Applicable models	VC-S VC-P
		Affect the flag	Zero flag Carry flag Borrow flag
Command list: ANR		Step size	1
Operand	Type	Applicable devices	Index

● **Operand Description**

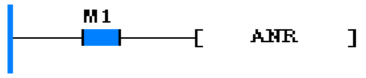
No operands.

● **Function Description**

1.When the power flow is valid, reset the running status of the signal alarms S900-S999; if there are multiple status actions, reset the one with the smallest number. When the power flow is valid again, the next one with the smallest number is reset.

Address number	Name	Function
SM400	Signal alarm is valid	After SM400 is turned ON, the following SM401 and SD401 work
SM401	Signal alarm action	Any action in the state S900-S999, SM401 is ON
SD401	On state minimum number	Save the minimum number of actions in S900-S999

● **Example of use**



LD M1

ANR

When the power flow is valid, if there are more than one S set by ANS, the one with the smallest number is reset.

6.22 Other Instructions

6.22.1 RND: Generate random number instruction

Ladder Diagram: ⊣ ⊢[RND (D)]									Applicable models			VC-S VC-P				
									Affect the flag			Zero flag				
Instruction List: RND (D)									Step size			3				
Operand	Type	Applicable devices														Index
D	INT		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T		Z	R	√

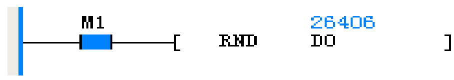
● **Operand Description**

D: The start number of the device where random numbers are stored.

● **Function Description**

1. Generate a pseudo-random number from 0 to 32767, and store its value in the D unit as a random number; if the generated random number is 0, set the Zero flag (SM80).

● **Example of use**



LD M1

RND D0

When M1=ON, a random number is generated and stored in D0, D0=26406.

6.22.2 DUTY: Generate timing pulse command

Ladder Diagram: [DUTY (S1) (S2) (D)]	**Applicable models**	VC-S VC-P	
	Affect the flag		
Instruction list: DUTY *(S1) (S2) (D)*	**Step size**	**7**	

operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	BOOL	SM														

● **Operand Description**

S1: Number of scans for ON

S2: Number of scans for OFF

D: The destination address that is always output at the timing

● **Function Description**

1. The timing pulse output unit D changes in the manner of S1 scan ON and S2 scan OFF;

2. SM unit, SM430-SM434

Target address of timing output	Device for counting the number of scans
SM430	SD330
SM431	SD331
SM432	SD332
SM433	SD333
SM434	SD334

3. This instruction can be used 5 times, but multiple DUTY instructions cannot use the same timing clock to output the target address.

● **Example of use**

When M1=ON, 10 scans of SM330 are ON, 10 scans are OFF, and the count value of the number of scans is stored in SD330.

● Precautions

The operation starts at the rising edge of the command, and the power flow does not stop even if it is cut off, and stops at STOP or power-off.

Chapter 7 Sequential Function Chart

7.1 Introduction to Sequential Function Chart

7.1.1 What is sequential function chart

Sequential Function Chart (Sequential Function Chart) is a programming language that has gradually developed and become popular in recent years. It is used to divide PLC programming projects into structured processes. It uses the programming elements and language structures specified in the IEC61131-3 standard to divide the complex system process into sequential multi-stage process steps and the conversion process between steps, thereby realizing the sequence control function.

Because SFC programming is intuitive and process-oriented, each step after decomposition and each transition condition is a relatively simple program process, which is very suitable for the application field of sequential control, so it has gradually been widely used.

7.1.2 What is the sequence function diagram of VEDA VC series PLC

The sequential function diagram of VEDA VC series PLC is a programming language used by VEDA VC series PLC products. In addition to the standard SFC functions, one or more ladder blocks can be built in.

Programs written with VEDA VC series PLC sequence function diagrams can be converted into corresponding ladder diagrams and statement list programs.

The sequential function chart program of VEDA VC series PLC also supports multiple independent processes, and the number can reach up to 20. These independent processes can run independently, and the step states within each process are scanned and transferred separately by process. Jumps can be made between individual processes.

7.1.3 Basic concepts of sequential function chart

SFC has the following two basic concepts: stepping states and transitions. Other concepts, such as jumps, branches, multiple independent processes, etc., are derived on this basis.

● Step state

1. Definition of step state

A step state is actually an independent program, representing a working state or a process in the sequence control process. A complete sequence function diagram program can be formed by organically combining multiple step states.

2. Step state execution

In the sequential function chart program, the step state is represented by a fixed S element.

A stepping state that is being executed is called an effective stepping state, and its corresponding S element state is ON. At this time, the PLC scans and executes all the instruction sequences in the stepping state. The step state that has not been executed is called an invalid step state, and the corresponding S element state is OFF. At this time, the PLC does not scan and execute the corresponding internal command sequence.

● Transfer

The sequence control process is a series of step state switching process. A PLC that is executing a certain stepping state will leave the current stepping state and enter and execute a new stepping state when certain logic conditions are met. This switching process is called a step state transition.

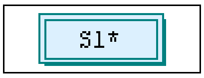
The occurrence of transition must satisfy certain logical conditions, which are called step transition conditions.

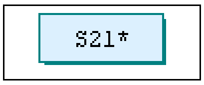
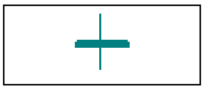
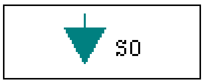
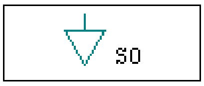
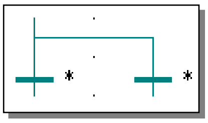
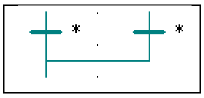
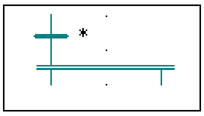
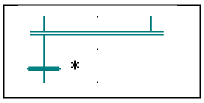
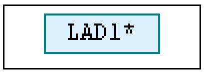
7.1.4 Programming primitives and their connection rules

● Programming primitives

VEDA VC series PLC sequence function diagram consists of the following basic programming primitives.

programming primitives

Programming primitives	Graphic expression	Specific instructions
Initial stepper		Represents an initial step state, the number of a step state is the specified S element number, and the number cannot be repeated. The execution of a SFC network must start with an initial stepper. The address range of the S soft element corresponding to the initial step is S0～S19

Programming primitives	Graphic expression	Specific instructions
Normal stepper		Represents an ordinary step state, the number of a step state is the specified S element number, and the number cannot be repeated. The S software address range corresponding to the common stepper is S20--S991 soft element
Transfer character		It represents a transition, and a transition condition (built-in ladder diagram) that makes the next step valid can be built in. The user can define the code in it, and when the transition condition is reached, the state of the next S soft element connected with the transition character is set to enter the next state of progress. Transition characters must be connected between step characters
Jump character		The jump symbol, connected after the jump symbol, can turn on the specified S element when the transition condition is reached. Loops or jumps for stepping states
Reset character		The reset symbol, connected after the transition symbol, can turn off the specified S element when the transition condition is reached. For the end of a SFC block
Select branch		After the step symbol is connected, it represents a plurality of mutually independent transition conditions. When any one of the transition conditions is reached, the previous step state is ended, and the corresponding step branch under the transition condition is entered. It is used to select one of multiple step branches. After selecting one branch, other branches will not be selected again.
Choose a confluence		Connected at the junction of the selection branch, it represents the junction of the selection step branch. When the transition condition of one of the branches is reached, it will transfer to the next progress state
Parallel branch		After the step is connected, the following multiple branches wait for the same transition condition. When the transition condition is established, the following multiple stepping branches are enabled and executed at the same time
Parallel confluence		Connected at the junction of parallel branches, the transition condition represents the sum of the end conditions of each branch. Multiple parallel stepping branches have been executed, and only after the transition conditions are met, the next stepping state can take effect.
Ladder block		The ladder diagram block is used to represent the ladder diagram instructions other than the sequence control chart flow, and can be used for the start of the initial step and the general operation.

● Programming Primitive Connection Rules

1. The initial step symbol cannot be preceded by other primitives, and the subsequent primitive must be a transition symbol, or it may not participate in the connection.

2. Ladder blocks are not connected to any other entities.

3. The primitive directly connected to the common step symbol must be a transfer symbol, and the common step symbol cannot exist in isolation in the graph.

4. The reset character and jump character should be preceded by a shift character and cannot be followed by other elements.

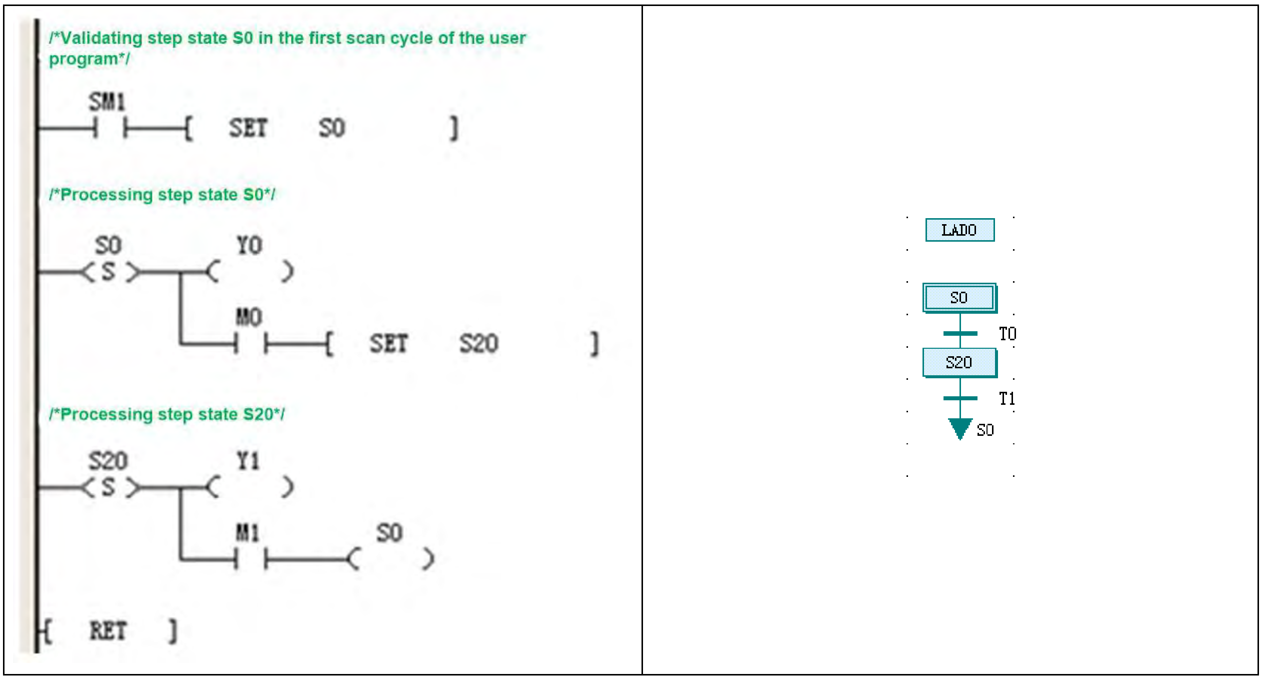
5. Transfer and jump characters cannot exist in isolation.

7.1.5 Sequential function chart structure

SFC process structure is divided into three categories: simple sequence structure, selection structure and parallel structure. In addition, jump is also a kind of selection structure.

● Simple sequential structure

The following figure is an example of a simple sequence structure and its ladder diagram representation.



In the simple sequence structure, when the step transition condition is satisfied, the sequence transitions from the previous step state to the next state without any branch structure. When the transition condition is satisfied in the last step state, exit the SFC block, or transfer to the initial step state.

1. Ladder block

The ladder block is used to start the sequence function chart segment, that is, the S element of the initial step symbol is set to ON, and the power-on start method is adopted in the above routine.

Ladder blocks are also used in other general blocks of non-sequential function charts.

2. Initial step state

In the example, the initial step state is initiated by the ladder block. The range of S element is 0～19.

3. Normal stepper

Used for programming in sequential processes. The range of S element is 20～1023.

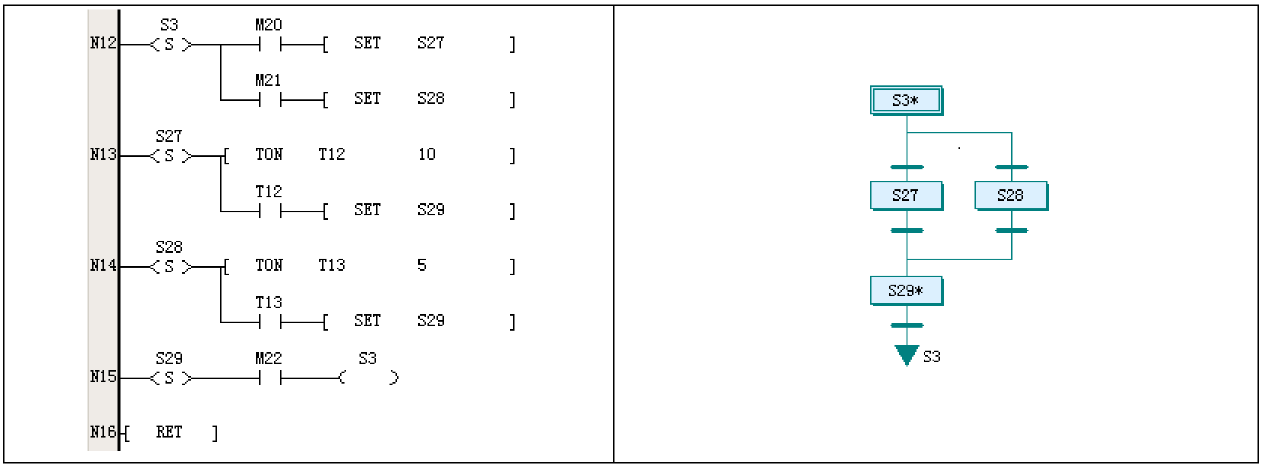
4. Transfer or reset

The last transition character of the sample program is connected to the jump character to jump to the initial step state. This is a process that operates in a continuous loop.

The last transition character can also be connected to a reset character, which resets the last step state. After the reset, the process operation of this simple sequence structure is completed, and then waits for the start of the next process operation.

● Choose branch structure

The example of the selection branch structure is shown in the following figure, the left figure is the ladder diagram, and the right figure is the corresponding sequence function diagram.



1. Choose branch

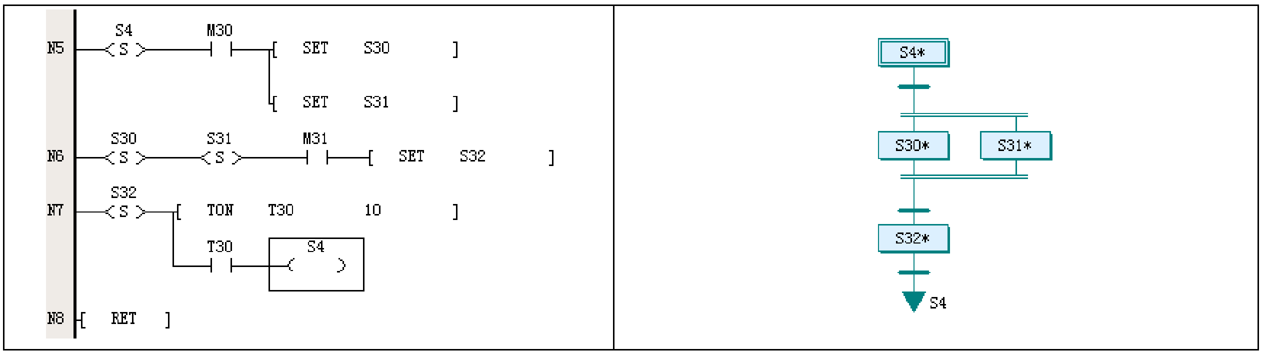
According to the transition conditions of each branch, the stepping state on the corresponding branch is selected to be activated. The user must ensure that the transition conditions in the branches are mutually exclusive. So the selection structure can only select one branch at a time while the process is running. As shown in the program in the above figure, in the N12 line of the program, the two stepping states of S27 and S28 are respectively M20 and M21 as transition conditions. When it is ensured that M20 and M21 will not be set at the same time, S27 and S28 can only be the two Choose one.

2. Select confluence

At the junction of alternative branches, all branches are connected to the same stepping state, and the transition conditions are independent of each other. As shown in the program in the figure above, the transition condition of the S27 stepping state in the N13th line is that the time T12 is up; and the transition condition of the S28 stepping state in the N14 line is that the T13 time is up. The transition result is to enter the next step state S29.

● Parallel branch structure

An example of a parallel structure is shown in the following figure, the left figure is a ladder diagram, and the right figure is the corresponding sequential function diagram.



1. Parallel branch

When the transition conditions of the parallel branch structure are satisfied, each step state connected to the parallel branch structure is activated simultaneously. This is also a common sequential control structure, that is, under certain conditions, multiple processes will be started and processed in parallel. As shown in the program of line N5 in the figure, M30 is the transition condition. When M30 is set, the stepping states of S30 and S31 are valid at the same time.

2. Parallel confluence

When the transition condition of the parallel merge structure is satisfied, each step state connected to the parallel branch structure will be invalid at the same time, and will be transferred to the subsequent step state. As shown in the program in the line of Figure N6, when in the stepping state of S30 and S31, when M31 is set, it transfers to the stepping state of S32, and ends the stepping state of S30 and S31.

The transfer conditions for parallel merges are to ensure that all the independent steps processed before the merge can be completed before the transfer can take place.

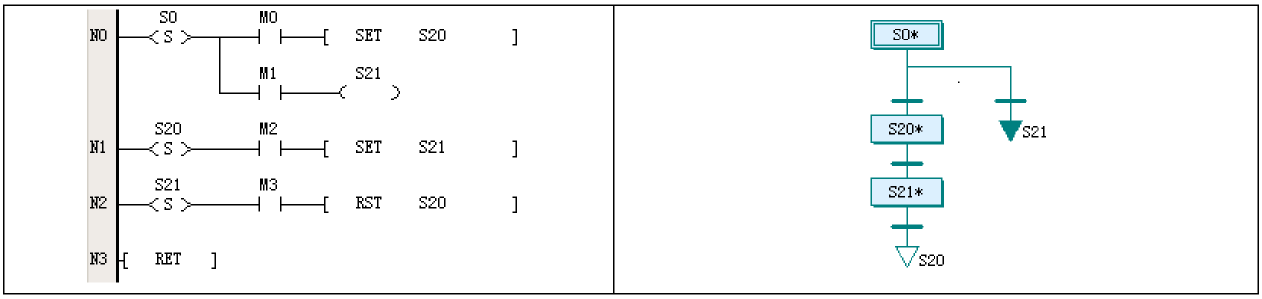
● Jump

Jump structures are often used for the following purposes: spanning part of a stepping state; looping: returning to the initial stepping state or a normal stepping state; transferring to other processes.

1. Step state across sections

In a process, according to certain transition conditions, when sequential execution is not required, the jump symbol can be used to transition to the required stepping state and cross part of the stepping state.

The following is an example diagram. The left side is the ladder diagram, and the right side is the corresponding sequence function diagram.



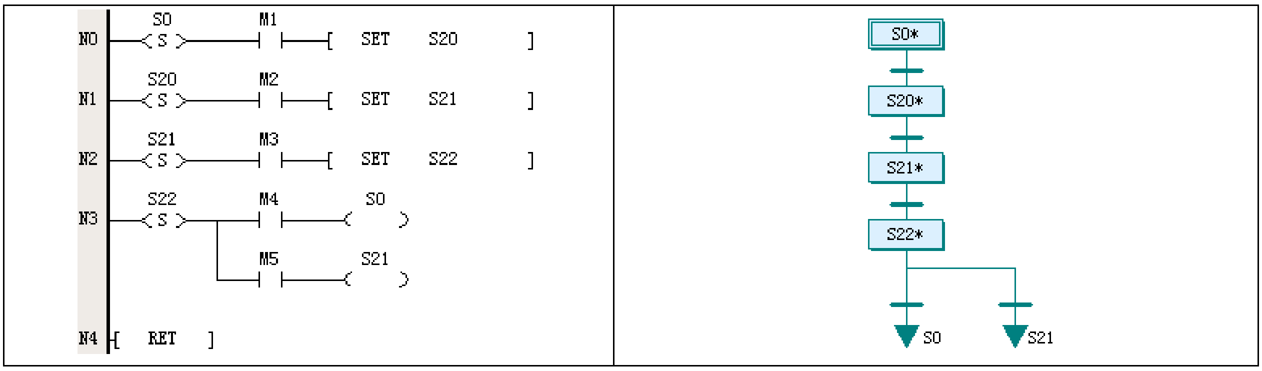
In the sequential function diagram, the S21 jump symbol is used to indicate the jump, and the S20 step state is crossed. The branch structure is actually selected before the jump.

In the ladder diagram, the second branch of the N0th line is the jump instruction. The jump instruction takes the form of the OUT coil instead of the SET instruction form of the sequential transfer. When running in S0 stepping state, when M1 is ON, it will jump to S21 state.

2. Cycle

In a process, according to certain transition conditions, when it is necessary to cycle between some or all of the stepping states, the jump symbol is used to realize the function of the cycle. At the last transition of this process, jump to the previous common step symbol to realize part of the step state cycle function; if it is to jump to the initial step symbol, then realize the full step state cycle function.

The following is an example program that realizes the above two loop structures at the same time. The left picture is the ladder diagram, and the right picture is the corresponding sequential function diagram.



In the sequence function diagram, in the step state of S22, when one of the transition conditions is satisfied, jump to S21, and re-run the step state of S21. In another transition condition, it will jump to the S0 initial step state and re-run all step states.

In the ladder diagram, the jumps of these two loops are implemented in the N3th line, and you can see the OUT coil of the jump instruction.

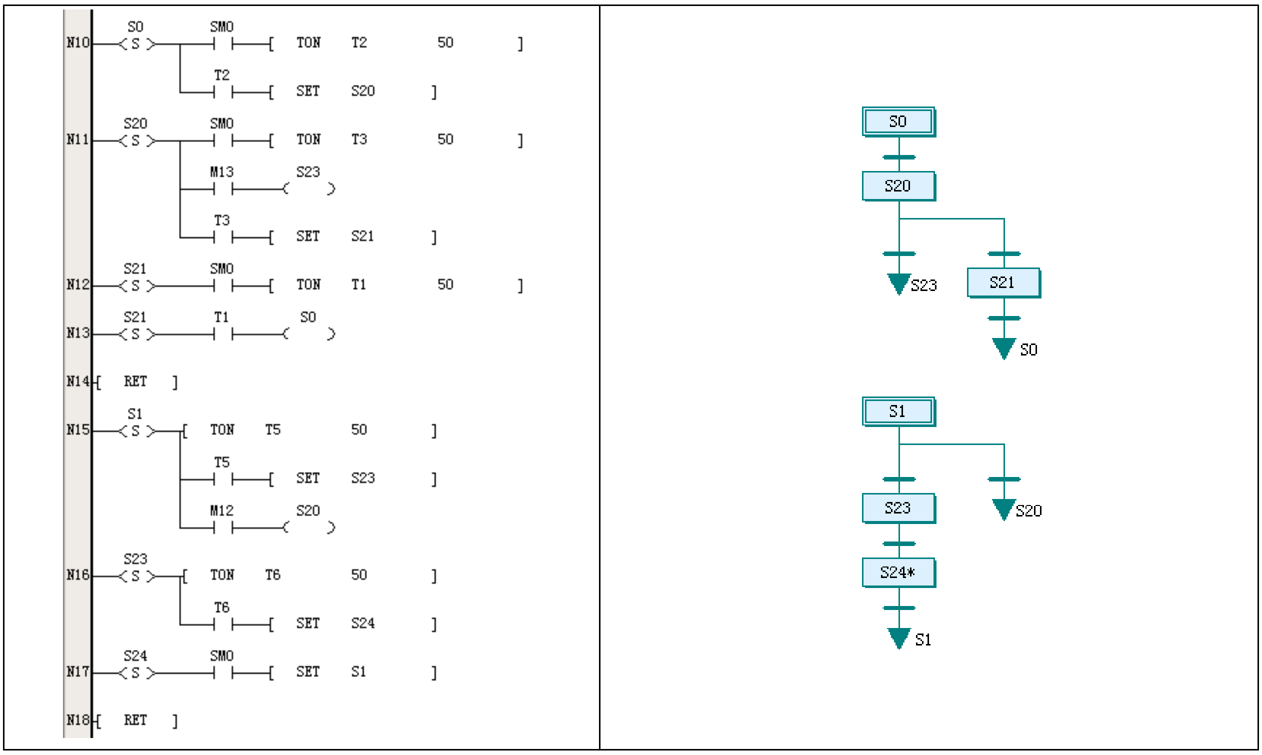
3. Jump between different independent processes

There can be multiple independent processes in the VEDA VC series PLC sequence function diagram program at the same time, and the jump between these processes is supported. A transfer condition can be set in an independent process, and when the condition is satisfied, it will directly transfer to another independent process. You can jump to the initial step state of another process, or you can go to the normal step state.

📖 Notice

Jumping between multiple processes adds complexity to the PLC program and must be treated with caution.

The following figure shows an example program that implements jumping from one independent process to another. The figure on the left is the ladder diagram, and the figure on the right is the corresponding sequential function diagram.



In the sequence function diagram, in the step state of the above process S20, you can jump to the step state S23 according to the transition condition; and in the step state S1 of the following process, you can also jump to the step S20 step according to the transition condition. state.

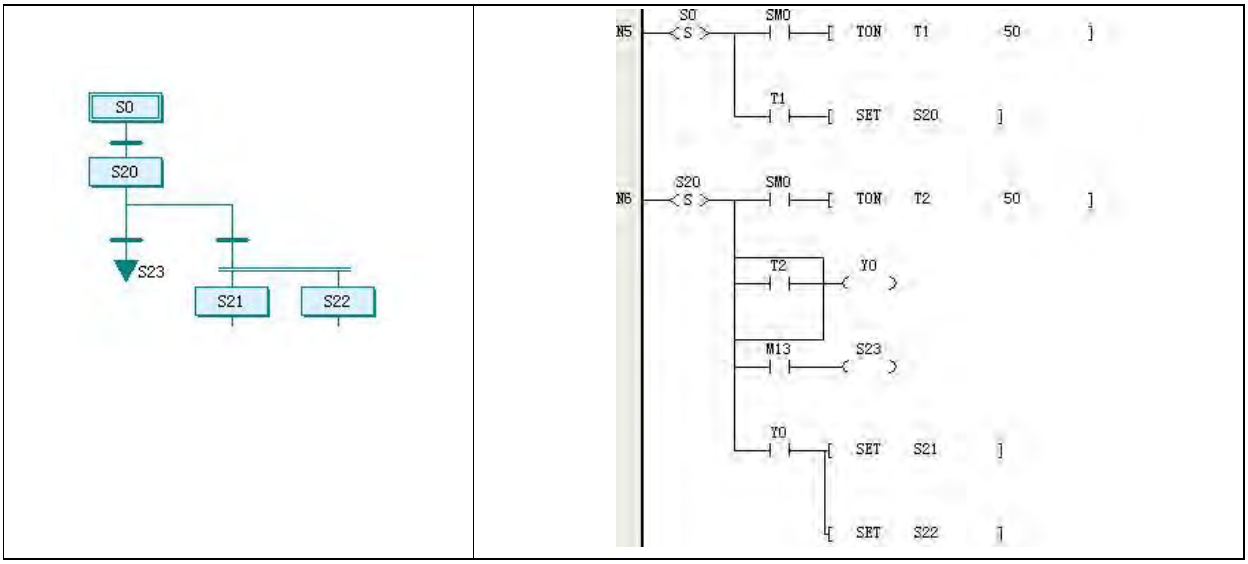
In the example diagram, it can be seen that this kind of jump is based on the selection branch structure, so when a jump between different processes occurs, the stepping state in the process where the jump occurs will all be invalid. For example, in the step state of the above process S20 in the example diagram, if it transfers to the step state of the following process S23, then S20 will be set to OFF, and all the step states S0, S20, and S21 of this independent process are OFF, that is, is in an invalid state.

7.1.6 Sequential function chart program execution

The execution process of the sequential function chart program is the same as that of the ordinary ladder diagram: both of them are continuously scanned from top to bottom, left and right.

The difference between the execution process of the sequential function chart program and the ordinary ladder diagram is that each step state of the sequential function chart will switch between the valid and invalid states according to the sequential conditions, and the internal instruction sequence of the corresponding valid step state will be scanned and executed, and the invalid state will be executed. The internal instructions in the stepping state will not be scanned and executed; but all the program lines of the ordinary ladder diagram main program will be passed and executed in each scan cycle.

As shown in the figure below, the ladder program on the right is converted from the sequential function chart program on the left. When the S20 stepping state is valid, the T2 timer will be scanned and timed,It will not enter the S21 and S22 states until T2 is completed; when M13 is OFF, it will not enter S23,None of their internal instructions are scanned for execution.



The ON/OFF switching between each S element is performed according to the step transition condition, and as a result, the previous step state is transferred to the next step state. When an S element transitions from ON to OFF, the output device of its internal command will be reset or cleared. See5.3.1 STL: SFC state load instruction.

📖 Notice

1. VEDA VC series PLC sequence function chart program generally contains both sequence function chart and ladder diagram blocks. The ladder diagram block is used to handle transactions other than the flow, including the operation of starting the sequential function chart, and is not controlled by any S element. In each scan cycle, the program lines of these ladder blocks scanned by the PLC will be executed.

2. Since the state change of the S element will affect the built-in instruction of the stepping state, and there is also a process of switching the up and down stepping state, there are some matters needing attention in the operation of the software element and the use of the instruction when programming the SFC.

7.2 Correspondence Between Sequential Function Diagram and Ladder Diagram

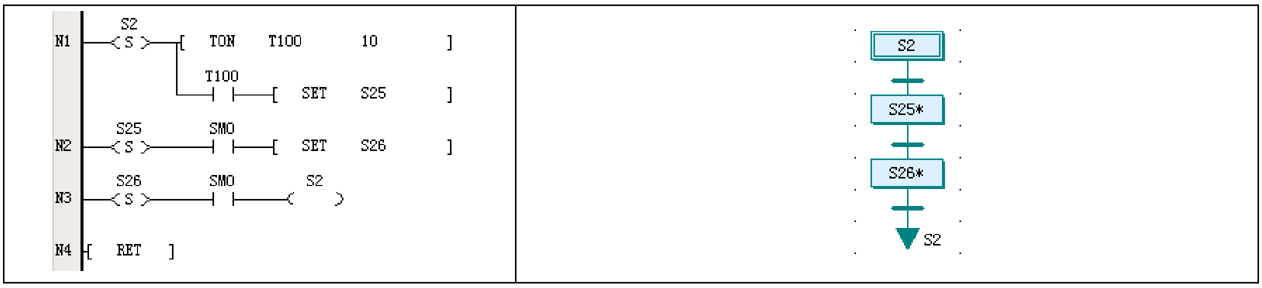
The SFC program can be represented by a ladder diagram. Use the ladder diagram to understand the actual meaning of the SFC program structure.

In the ladder diagram, various primitive symbols of SFC programming have corresponding SFC instructions, and the corresponding process also has a specific structure.

7.2.1 STL instruction and step status

In the ladder diagram, a step state is started by the STL instruction. Each step state is marked by an S element.

The following left figure shows the ladder diagram program of a simple sequence structure example program. The picture on the right shows the SFC program of this process.



In the ladder diagram, the S2 stepping state starts from the STL state loading instruction. The subsequent TON timer statement is the internal instruction sequence of the S2 stepping state. A stepping state internal instruction sequence can have multiple statements, which is basically the same as the ordinary ladder diagram program, and is actually a relatively complete program segment.

The difference between the initial stepping state and the ordinary stepping state is only the range of the S element selected.

For details of STL instructions, please refer to5.3.1 STL: SFC state load instruction. It should be noted that when the step state transitions from ON to OFF, the built-in OUT, TON, TOF, PWM, HCNT, PLSY, PLSR, DHSCS, SPD, DHSCI, DHSCR, DHSZ, DHST, DHSP, BOUT correspond to The Destination operand will be cleared.

📖 Notice

Since the PLC scans continuously and periodically, when a step state is transferred to the next step state, those built-in statements in the original step state will not be affected by the ON to OFF transition until the next scan. see7.4.1 Common programming mistakes reusing step status characters.

7.2.2 SFC state transition instruction

As shown in the figure above, the transition symbol in the right figure is implemented by the SFC state transition instruction in the ladder diagram in the left figure.

The transition condition is composed of the normally open contact elements in front of the SET statement. Normally open contact elements are controlled by built-in statements or external operations.

When the power flow of the SFC state transition instruction is valid, the specified step state is set to be valid, and the current valid step state is set to invalid at the same time, and the action of the step state transfer is completed.

7.2.3 RET instruction and SFC block

As shown in the figure above, the SFC program on the right starts with the initial step symbol of S2, and returns to the step symbol of S2 after 2 ordinary step symbols. In the ladder diagram, the end of the SFC program segment must be marked with the RET instruction.

The RET instruction can only be used in the main program.

7.2.4 SFC state jump instruction, reset instruction

In the above figure, the jump symbol S2 is shown in the N3 line in the ladder diagram. Using the OUT instruction, the jump is realized. Jumps can be in the same process or between different independent processes.

If the reset symbol S26 is adopted, the N3 row in the ladder diagram is the RST instruction, which realizes the reset to the previous step state S26.

7.2.5 SFC Alternative Branches, Parallel Branches, and Convergence

For an example of a ladder diagram of an alternative branch, see7.1.5 Sequential function chart structuremiddleChoose branch structure.

For a ladder diagram example of a parallel branch, see7.1.5 Sequential function chart structuremiddleParallel branch structure.

7.3 SFC Programming Steps

1) Analyze the process and determine the program process structure

The program flow structure can be divided into simple sequence structure, selection structure, parallel structure, and jump is also a kind of selection structure. When programming with SFC, the first step is to determine which process structure it is. For example, a single object continuously completes operations through sequential steps before and after, which is generally a simple sequential structure; if there are multiple product processing options, each option has different parameters and cannot be processed at the same time, it should be determined as the selection structure; Relatively independent, it may be a parallel structure.

2) Identify the main steps and main transition conditions, resulting in a process sketch

After the process structure is determined, the next step is to determine the main steps and main transition conditions in general. The process structure is divided into detailed operation processes, each operation process is a step, and the important sign of the end of the operation process is the transition condition. This will give you a sketch of the process.

3) Make SFC sequence function diagram according to the process sketch

Open the SFC programming interface of the VEDA PCT programming software, and turn the process sketch into an SFC sequence function diagram. At this time, the executable PLC program can be obtained, but the program needs to be improved.

4) Make a table of input and output points, and determine the operation objects and actual transition conditions of each step

Input points are mostly transition conditions, and output points are mostly operation objects. According to the point table, the sequential function diagram can be further revised.

5) Input of steps and transition conditions

In the SFC programming interface, use the right mouse button to click on the SFC primitive, the corresponding right-click menu can be popped up, and the built-in ladder diagram option can be selected to open the built-in ladder diagram editing work area of the element, and input the ladder diagram program and conditions.

6) Writing Ladder Blocks

Don't forget to write some general-purpose processing function ladder blocks in the program, such as the start of the sequence flow, as well as the general operations such as stop, alarm and so on. These all need to be placed in a ladder block.

Notice

Start-stop operations are related to personal and equipment safety. Considering the particularity of SFC programming, close all outputs that should stop running as much as possible when stopping.

7.4 SFC Programming Considerations

Since the STL statement has some characteristics, and the PLC is periodically scanned according to the statement sequence, there are several important considerations for SFC programming.

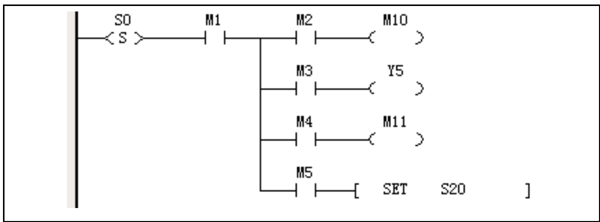
7.4.1 Common programming mistakes reusing step status characters

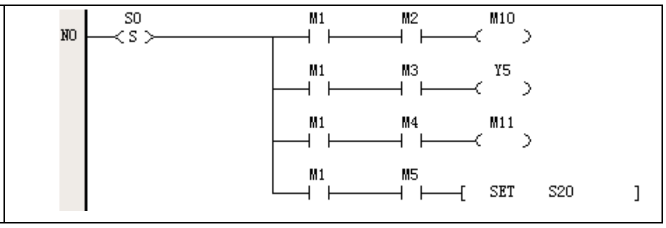
In the same PLC program, each step state symbol used for sequence control programming is corresponding to a unique S element and cannot be reused.

This requirement must be paid attention to when adopting the ladder diagram input.

1) Branch after transition condition

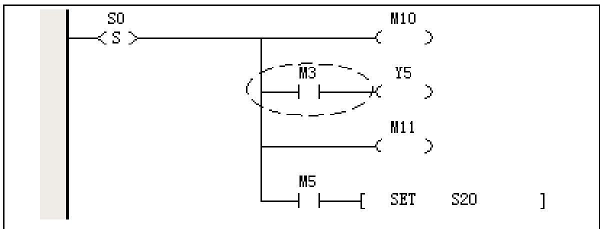
After the transition condition, the branch with condition cannot be separated. For example, the following program on the left will not pass compilation, because M1 has become a transition condition, and no branch can be made thereafter. It should be modified to the program shown on the right, which can be compiled correctly.

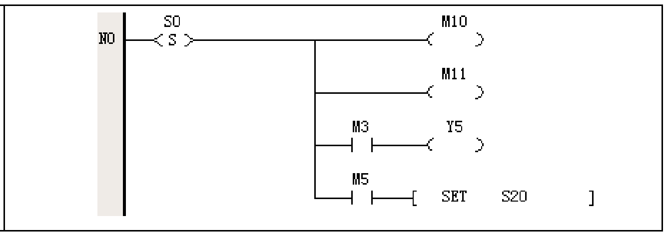




2) Incorrect use of normally open and normally closed contacts and output coils

When a normally open or normally closed contact instruction is used in a branch, the output coils in the subsequent branches cannot be directly connected to the internal bus, otherwise it cannot be compiled, as shown in the left figure below. Modify the branch sequence as shown in the figure on the right, and it can be compiled.



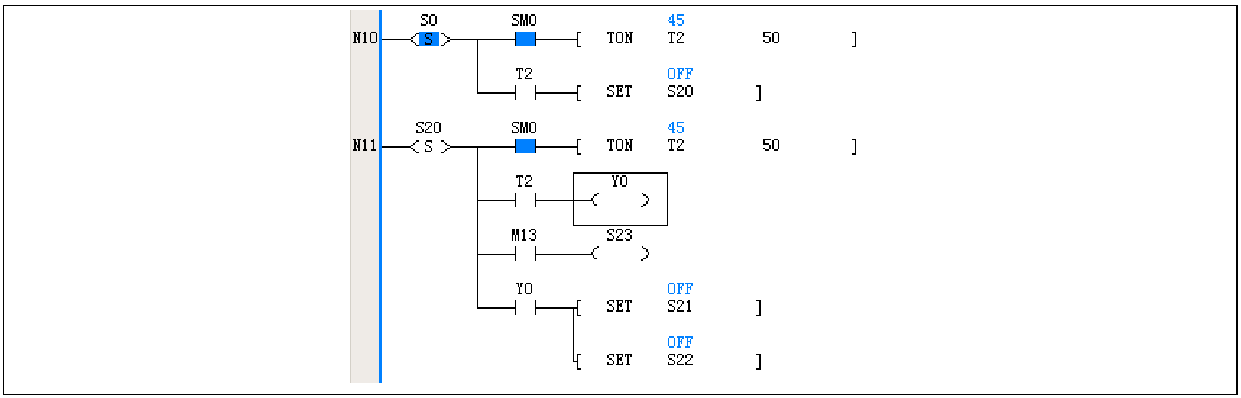


3) Repeated use of devices in adjacent step states

When the PLC executes the program, it cyclically scans according to the instruction sequence. When transitioning from the previous step state to the next step state, the instruction sequence in the previous step state has just finished scanning, and the next step state instruction sequence has also been opened for scanning to form a control output.

According to the above analysis, when the STL instruction changes from ON to OFF, although some internal components will be reset (see5.3.1 STL: SFC state load instruction), but this reset operation can only be performed on the next scan cycle. At the moment of transition of the stepping state, the internal components in the previous stepping state still maintain the original data and state until the next scanning passes through the stepping state.

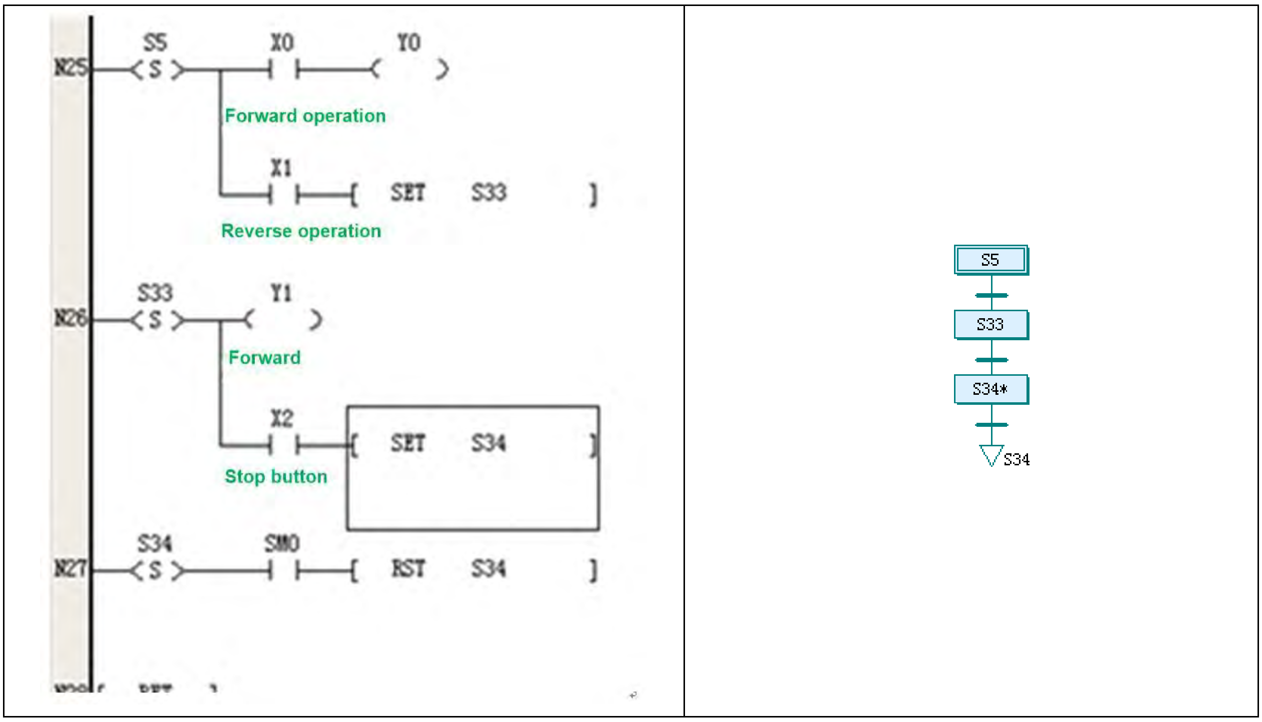
As shown in the figure below, the T2 timer is used simultaneously in the upper and lower linked step states. When the step state changes from S0 to S20, the T2 element will keep the count value and the on state. Therefore, the S20 stepping state cannot perform the timing operation according to the user's original design, but directly enters the following S21 and S22 stepping states. Therefore, in different stepping states, although programming soft components can be used repeatedly, it is best not to use them in adjacent stepping states, otherwise it may cause unexpected results.



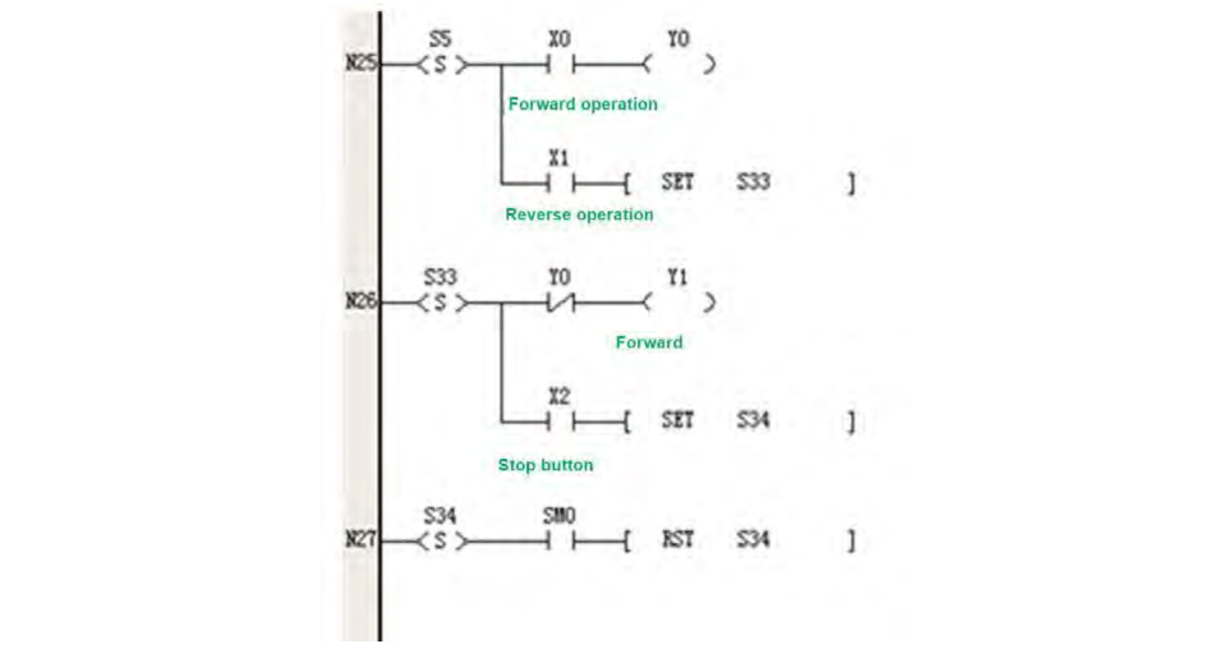
4) Device failed to interlock

In SFC programming, there may be conflicts between some soft elements due to the special circumstances of stepping state transition. It needs to be interlocked at this point.

For example, the following figure shows an example of a forward and reverse sequence operation program. Y0 and Y1 are the forward and reverse control outputs of the device operation, respectively. X0 is the forward operation, X1 is the reverse operation, and X2 is the stop button. It is required that Y0 and Y1 are interlocked, that is, they cannot be ON at the same time. However, in this routine, when the device is running in the forward direction, when X1 is turned on to make the S5 stepping state transfer to the S33 stepping state, Y0 and Y1 are ON at the same time and the time is one program scan cycle.



Therefore, interlocking statements should be added to the program. The following is an example. In the program in the above figure, the normally closed contact of Y0 is added before the output coil of Y1 as an interlock.

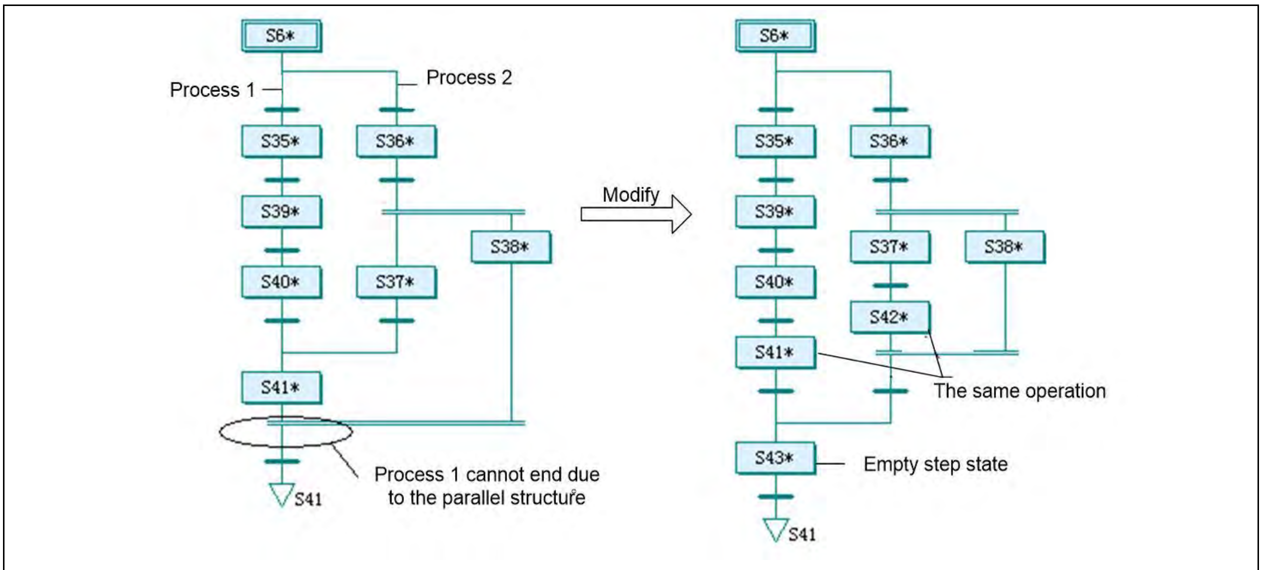


5) Jump and transition mixed

Jumps are mostly applications that switch between different processes and non-adjacent stepping states. A transition is an operation that switches between adjacent step states. If you change the place where the jump should be used from the OUT coil to the SET statement, or change the SET statement to the OUT coil where the jump should be used, it will not pass the compilation.

6) The selection branch transition is a parallel branch merging structure, which makes it impossible to end the process.

Alternative branching is a multi-select one process, and if parallel branches are mixed in it, a process error may occur that causes the alternative branch to run unfinished. As shown below. In the program on the left, when process 1 is executed to step state S41, because the transition condition is a parallel branch, and the system will not run process 2 at this moment, the transition condition at this place will never be realized, and a process error occurs.



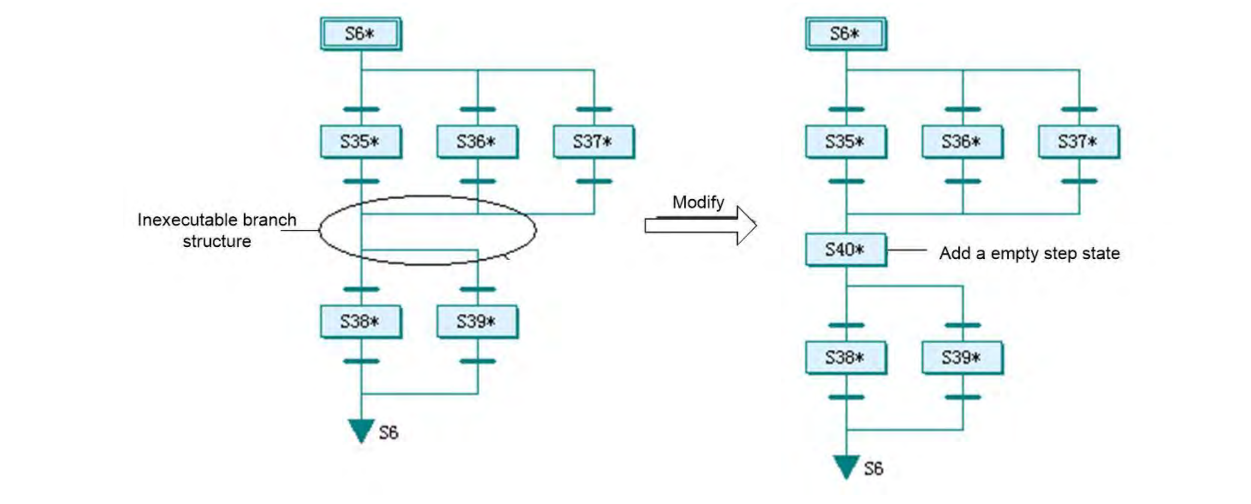
The modification method is shown in the figure on the right, adding step state S42, the function is exactly the same as S41; adding S43 empty step state, only as a programming structural element, without substantial operation. The transition conditions of S38, S41, and S43 need to be designed by the programmer. For example, the transition conditions of the original S41 can be used.

7.4.2 Programming skills

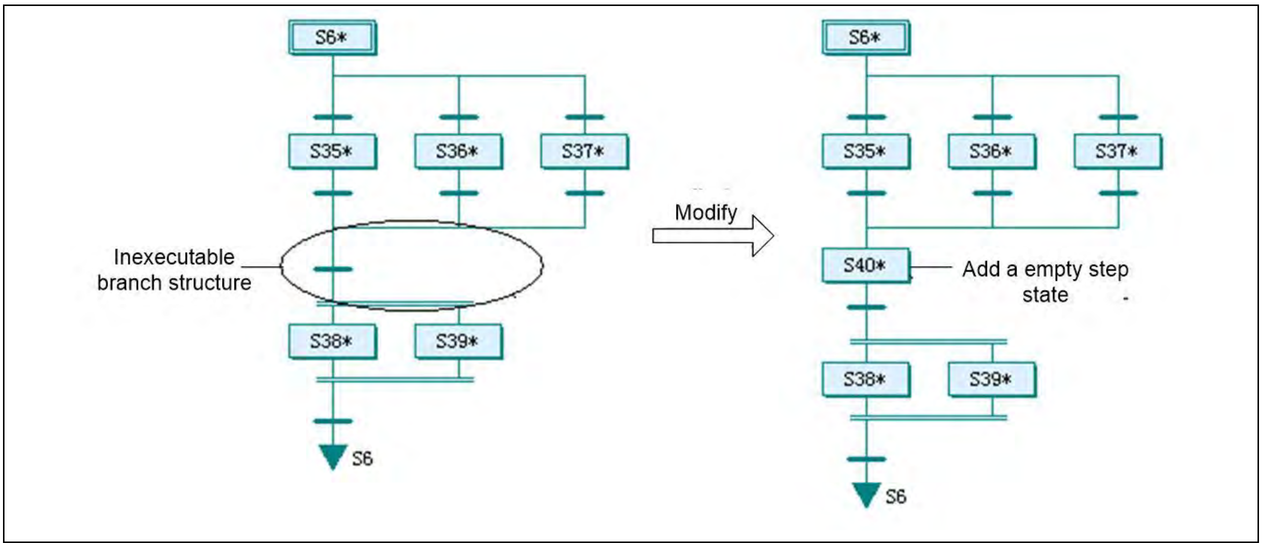
1) Smart use of empty step state

Some branch designs with syntax problems require the use of empty stepping states to solve branching problems. The so-called empty stepping state means that there is no operation with substantial content arranged in the stepping state, and the transition is directly waited for. Below are some examples.

In the left figure below, the alternative branch is connected to another alternative branch immediately after the confluence, which cannot be compiled. It can be modified according to the figure on the right to add an empty step state.



In the left figure below, it is not possible to connect another parallel immediately after selecting a branch. It can be modified according to the figure on the right, and an empty step state can be added.

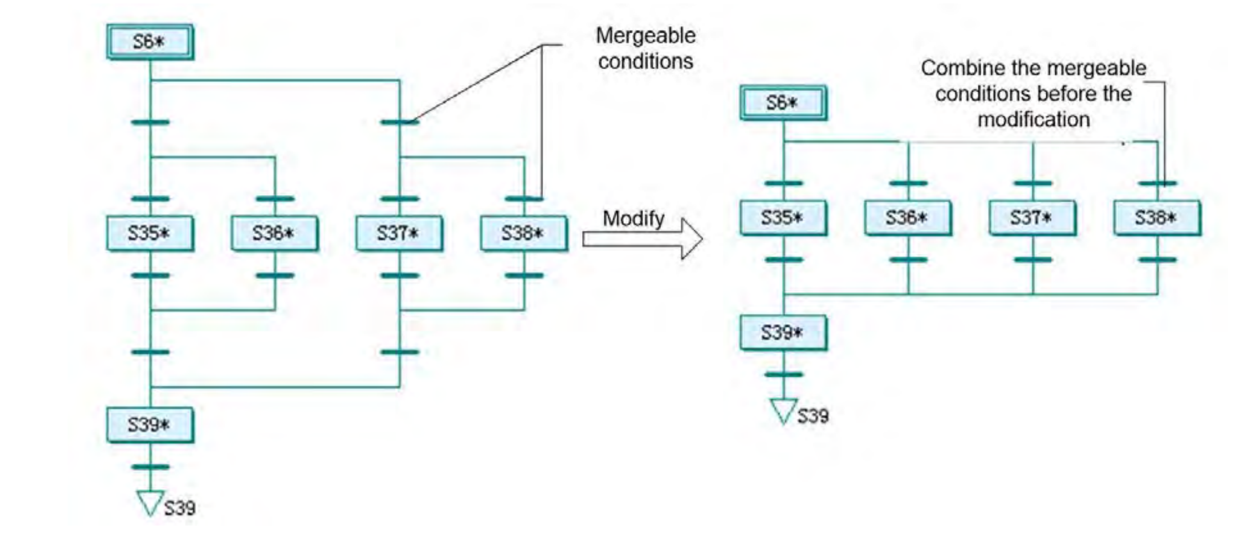


For other problem branches, such as parallel merging followed by parallel branch, parallel branch followed by selection branch, the problem can also be solved by adding an empty step state.

2) Merge branches and transition conditions

Some seemingly complicated branches are actually caused by improper analysis at design time, and can be merged or simplified appropriately.

As shown in the figure below, the designer first made the first selection branch, and then made two selection branches respectively. In fact, it is only necessary to use a selection branch with four branches, and the upper and lower transition symbols of the original design are merged into a first-level transition symbol with the transition condition AND.



3) Take advantage of the power failure hold function

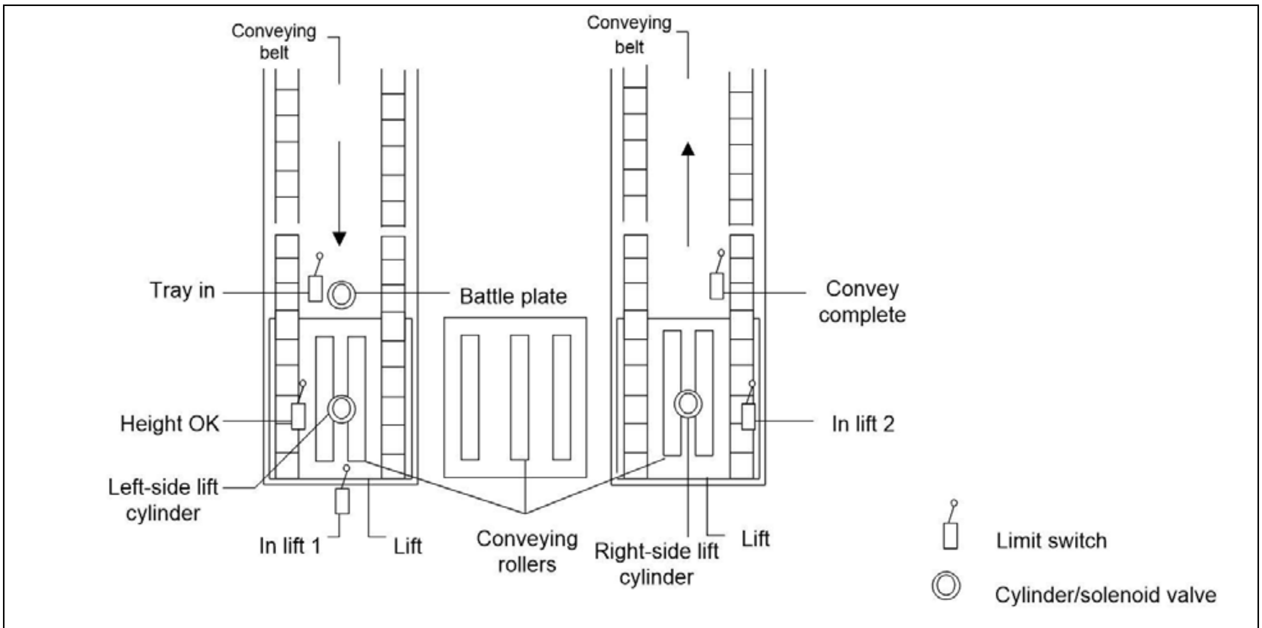
The value of the S element can be maintained by the power failure retention setting, and the operation can be restarted from the stepping state at the time of power failure after the power is restored.

7.5 Sequential Function Chart Programming Example

The examples in this section can only be used as a concise SFC programming demonstration case, the operations and conditions are simplified. The equipment configuration design expresses a rough concept and cannot be regarded as a design for actual equipment. It is only for learning reference.

7.5.1 Simple structure process

The following example is a workpiece pallet lift conveyor. The conveyor uses cylinder lifts and transfer rollers to transfer workpiece pallets from one conveyor belt to another. The figure below shows the top view of the conveyor belt and workpiece pallet lifting conveyor.

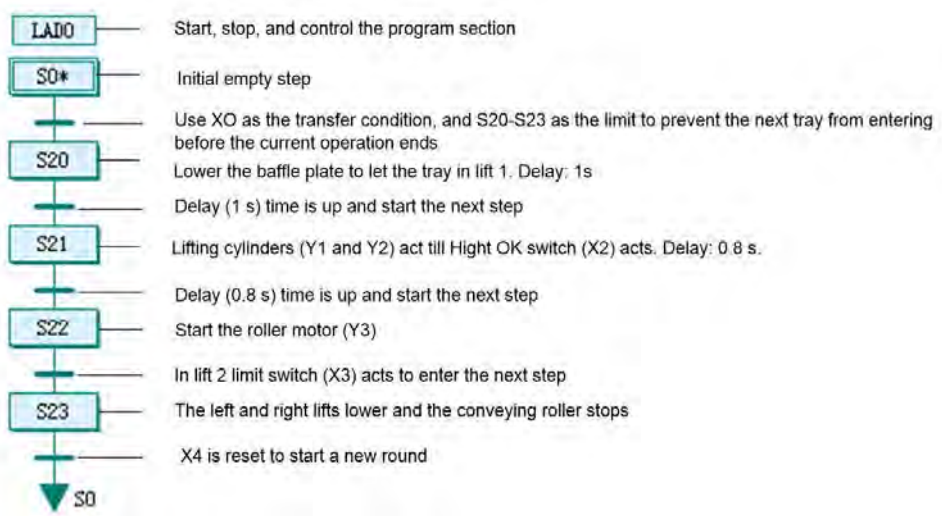


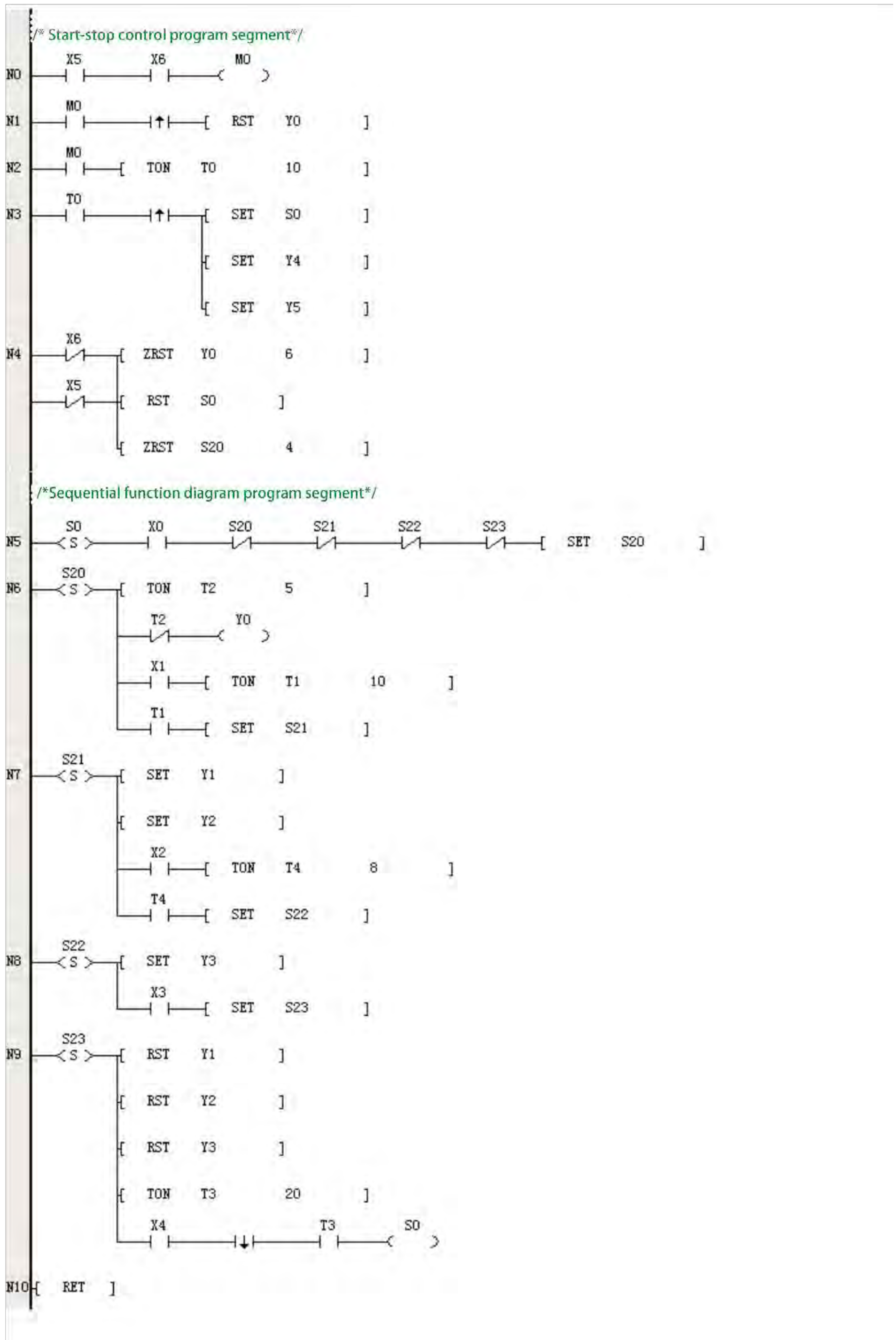
After the equipment is started, the workpiece pallet is conveyed along the left conveyor belt to the entrance of the lifting conveyor, and the "pallet entry travel switch" is triggered. When no workpiece pallet is conveyed on the entire conveyor, the inlet baffle is lowered to convey the workpiece pallet into the lift conveyor. Wait until the workpiece pallet completely enters the elevator on the left, and touch the "in-position travel switch", the lifting cylinder will act, the elevator will rise, and the "up-in-position travel switch" will be triggered when it is in place. The transfer roller motor starts after being lifted to the right position, and transfers the workpiece pallet to the elevator on the right. After reaching the position, the "transfer position switch" will be triggered, and then the cylinder of the elevator will act and the elevator will descend. The workpiece pallet falls onto the right conveyor and is taken away from the elevator. When the transmission is completed and the travel switch is reset, a complete lifting and conveying process ends, and then the next lifting and conveying process is entered. The table below is the table of input points and output points.

Serial number	Point address	Monitoring object		Serial number	Point address	Monitoring object
1	X0	Tray entry travel switch		8	Y0	Inlet flapper cylinder solenoid valve
2	X1	In-position travel switch		9	Y1	Left elevator cylinder solenoid valve
3	X2	Lift-to-position limit switch		10	Y2	Right elevator cylinder solenoid valve
4	X3	Transfer position limit switch		11	Y3	Transfer Roller Motor Contactor
5	X4	Transmission complete travel switch		12	Y4	Left conveyor motor contactor
6	X5	start switch		13	Y5	Right conveyor motor contactor
7	X6	Emergency switch auxiliary signal				

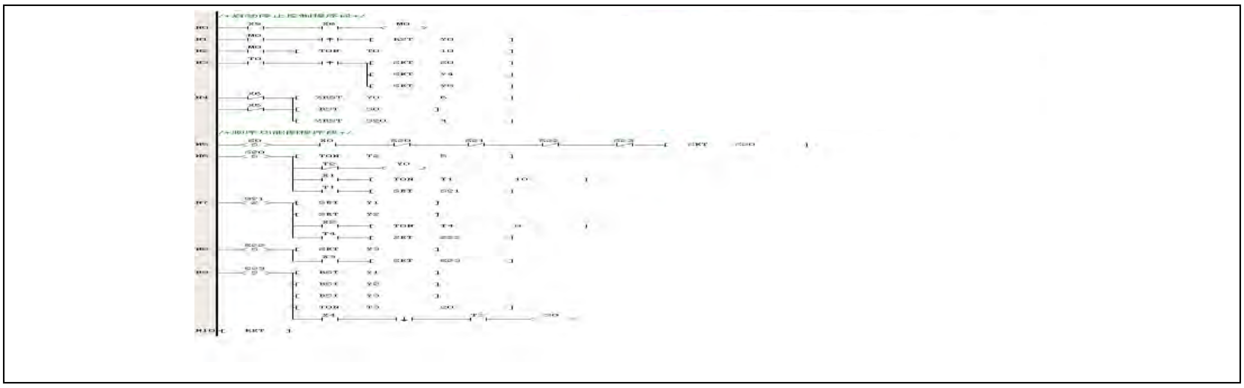
It can be seen that this is a simple sequential process. Each pallet is conveyed in several consecutive steps, with no other options or parallel steps, and no parallel processes. Designing programs with sequential function diagrams is simpler, faster, and more organized than conventional logic design.

The following is the sequential function chart program and the corresponding ladder program.



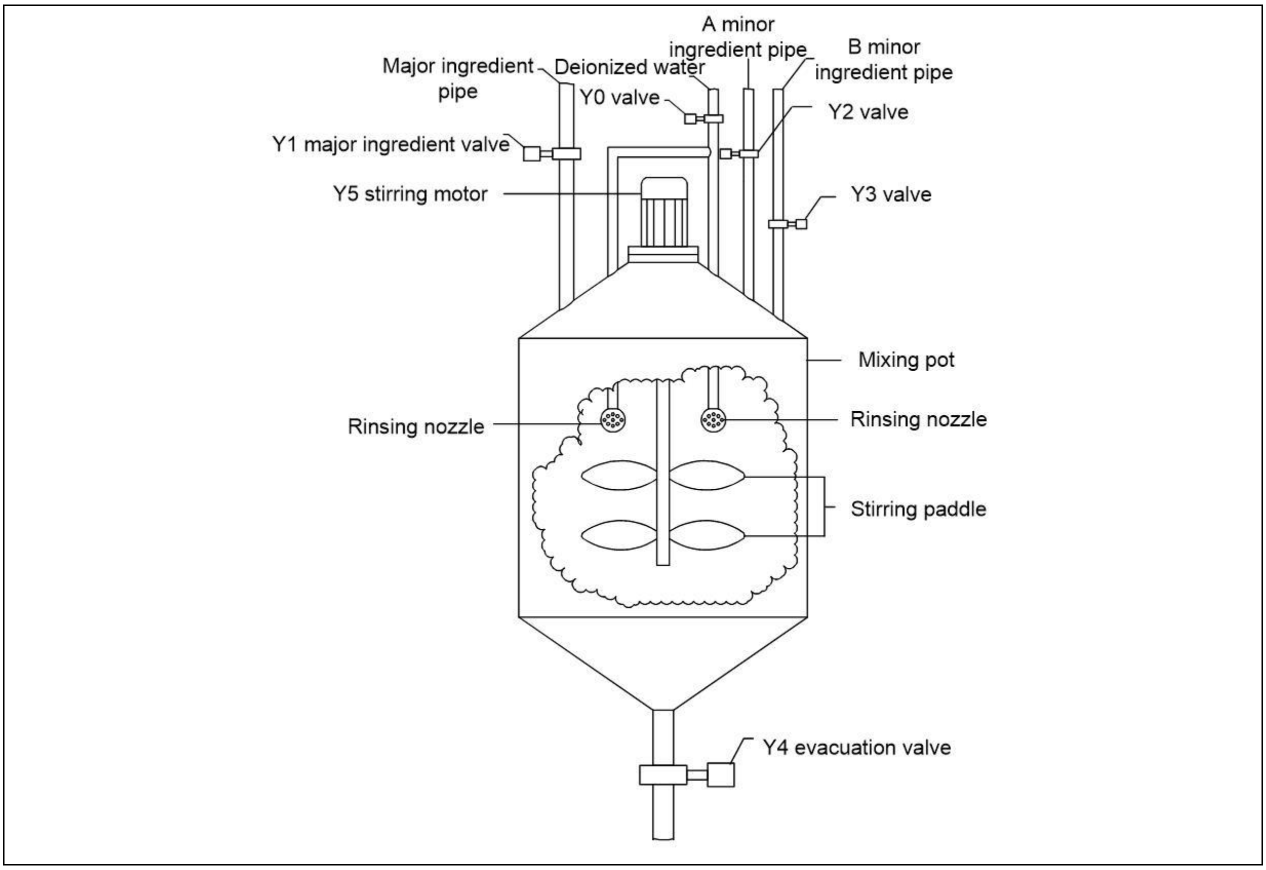


/* Start-stop control program segment*/
N0 X5 X6 M0
N1 M0 RST Y0
N2 M0 TON T0 10
N3 T0 SET S0
SET Y4
SET Y5
N4 X6 ZRST Y0 6
X5 RST S0
ZRST S20 4
/*Sequential function diagram program segment*/
N5 S0 X0 S20 S21 S22 S23 SET S20
N6 S20 TON T2 5
T2 Y0
X1 TON T1 10
T1 SET S21
N7 S21 SET Y1
SET Y2
X2 TON T4 8
T4 SET S22
N8 S22 SET Y3
X3 SET S23
N9 S23 RST Y1
RST Y2
RST Y3
TON T3 20
X4 T3 S0
N10 RET



7.5.2 Choose structure

The following example is a material mixing operation flow. Through this process, two kinds of products, A and B, can be produced. The following figure is a schematic diagram of the production equipment.



When running, the first step is to select the next batch of variety A or B through the touch screen, and then start production. The second step is to add main raw materials. When the weight reaches 2000kg, stop feeding; the third step is to add auxiliary raw materials. When producing Type A products, add 500kg of auxiliary materials A, and when producing type B products, add 500kg of auxiliary materials B; the fourth step is to stir for 20 minutes ; The fifth step is discharging. When the remaining material is less than 20kg and the delay time is up, the discharging is completed. After these are completed, re-enter the next batch of production process.

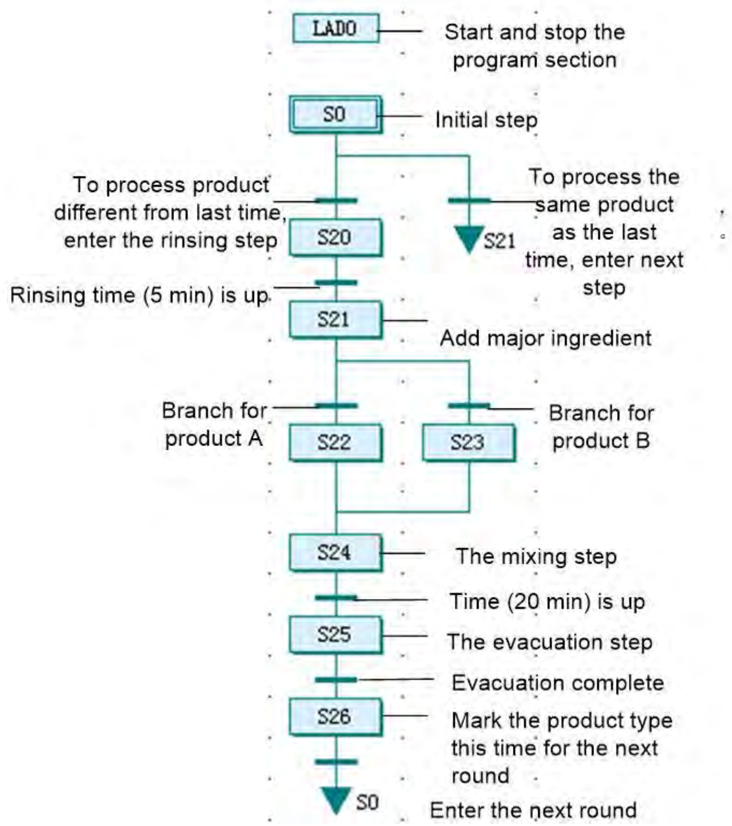
If it is the first time to start production, or the product variety of the previous batch is different from the next batch, open the deionized water and discharge valve before adding the main raw material, and clean for 5 minutes.

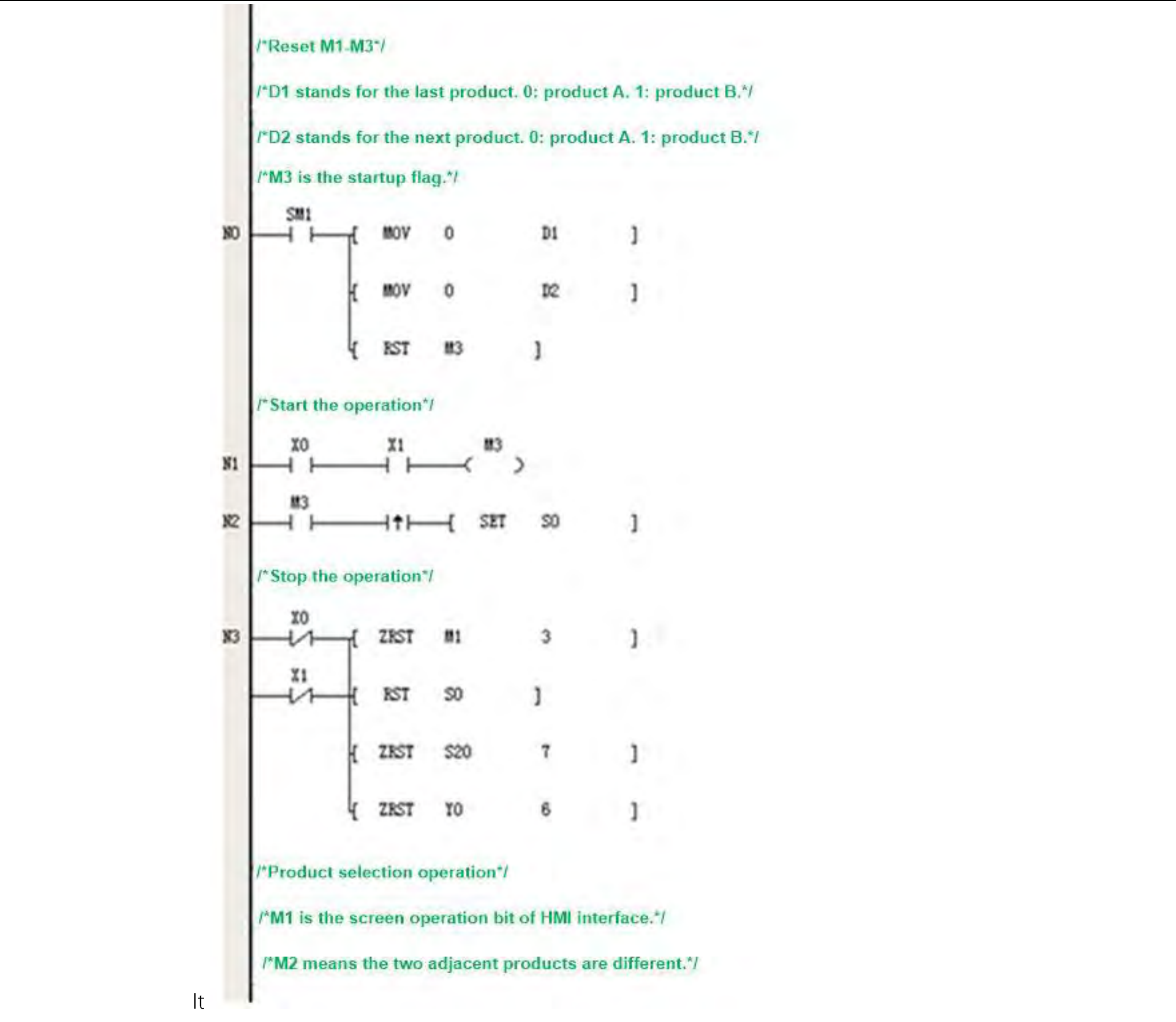
The following is the table of input points and output points.

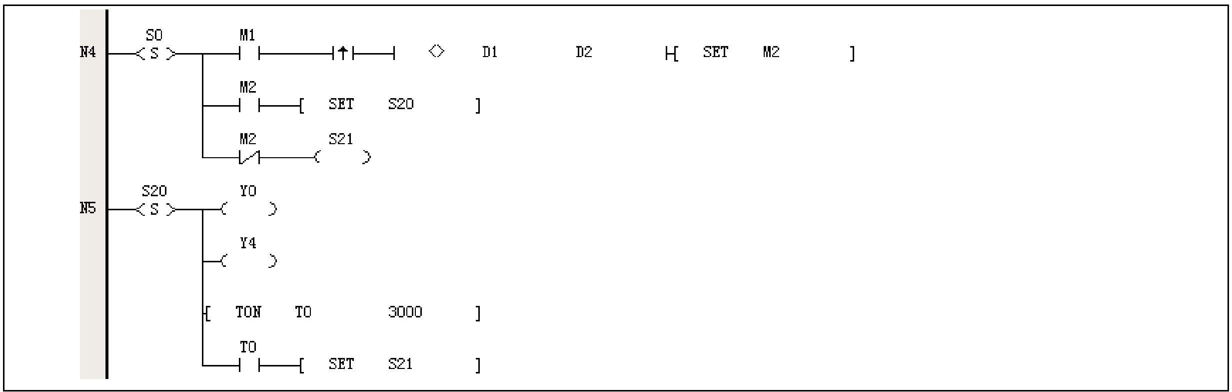
Serial number	Point address	Monitoring object		Serial number	Point address	Monitoring object
1	X0	Deionized water valve open		10	X11	The discharge valve is open
2	X1	Deionized water valve closed		11	X12	Discharge valve closed state
3	X2	Main raw material valve open state		12	Y0	Deionized water solenoid valve
4	X3	Main raw material valve closed state		13	Y1	Main raw material solenoid valve
5	X4	A auxiliary material valve open state		14	Y2	A auxiliary material solenoid valve
6	X5	A auxiliary material valve closed state		15	Y3	B auxiliary material solenoid valve
7	X6	B auxiliary material valve open state		16	Y4	Discharge solenoid valve
8	X7	B auxiliary material valve closed state		17	Y5	Stirring Motor Contactor
9	X10	Stirring motor running status				

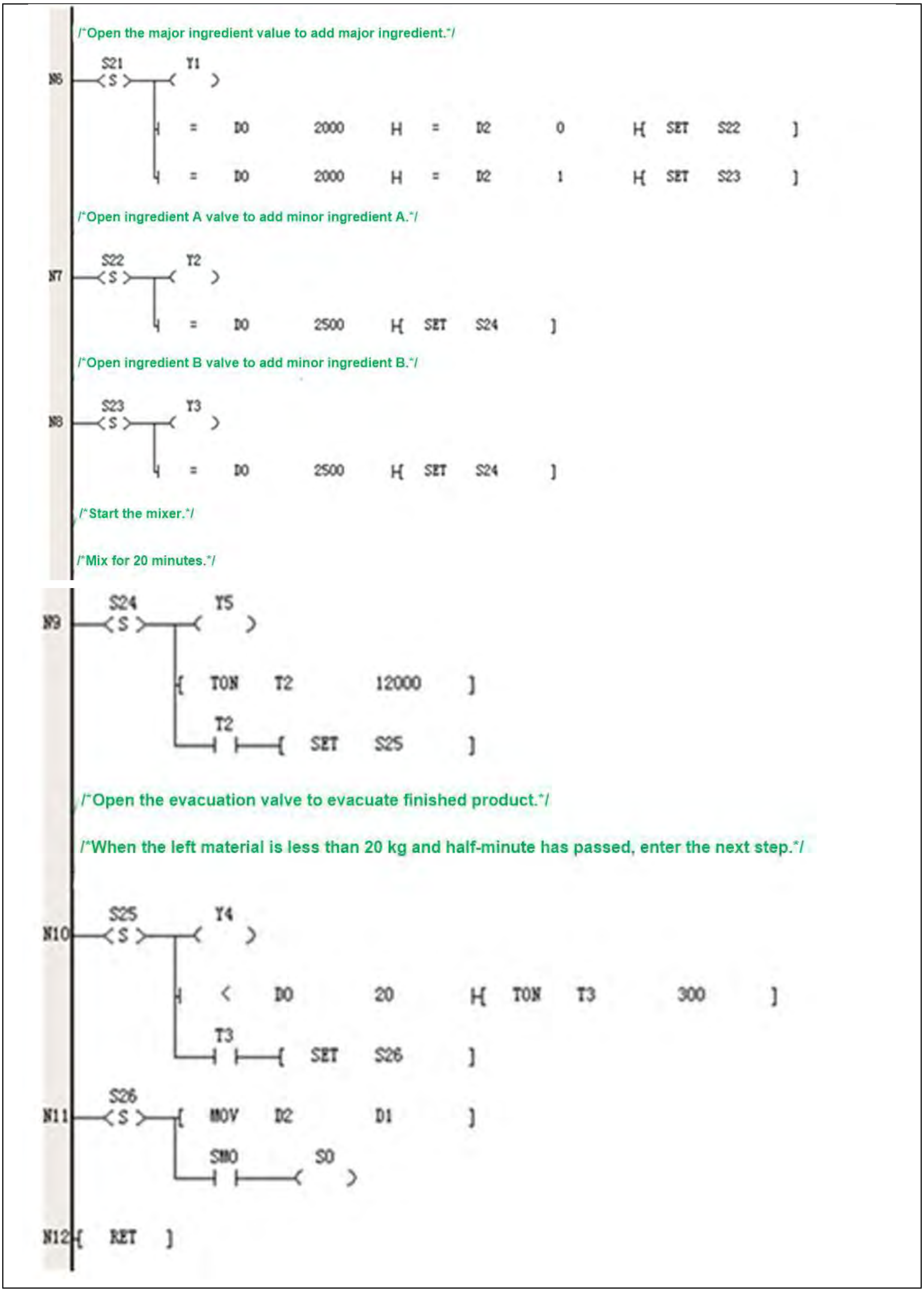
It can be seen that this is a process of selecting a structure. When producing a product, only one of A or B can be selected. It is only possible to switch varieties after production is completed. At the same time, there is also a selection and jump structure in the process, that is, the cleaning step.

The following is the sequential function chart program and the corresponding ladder program.



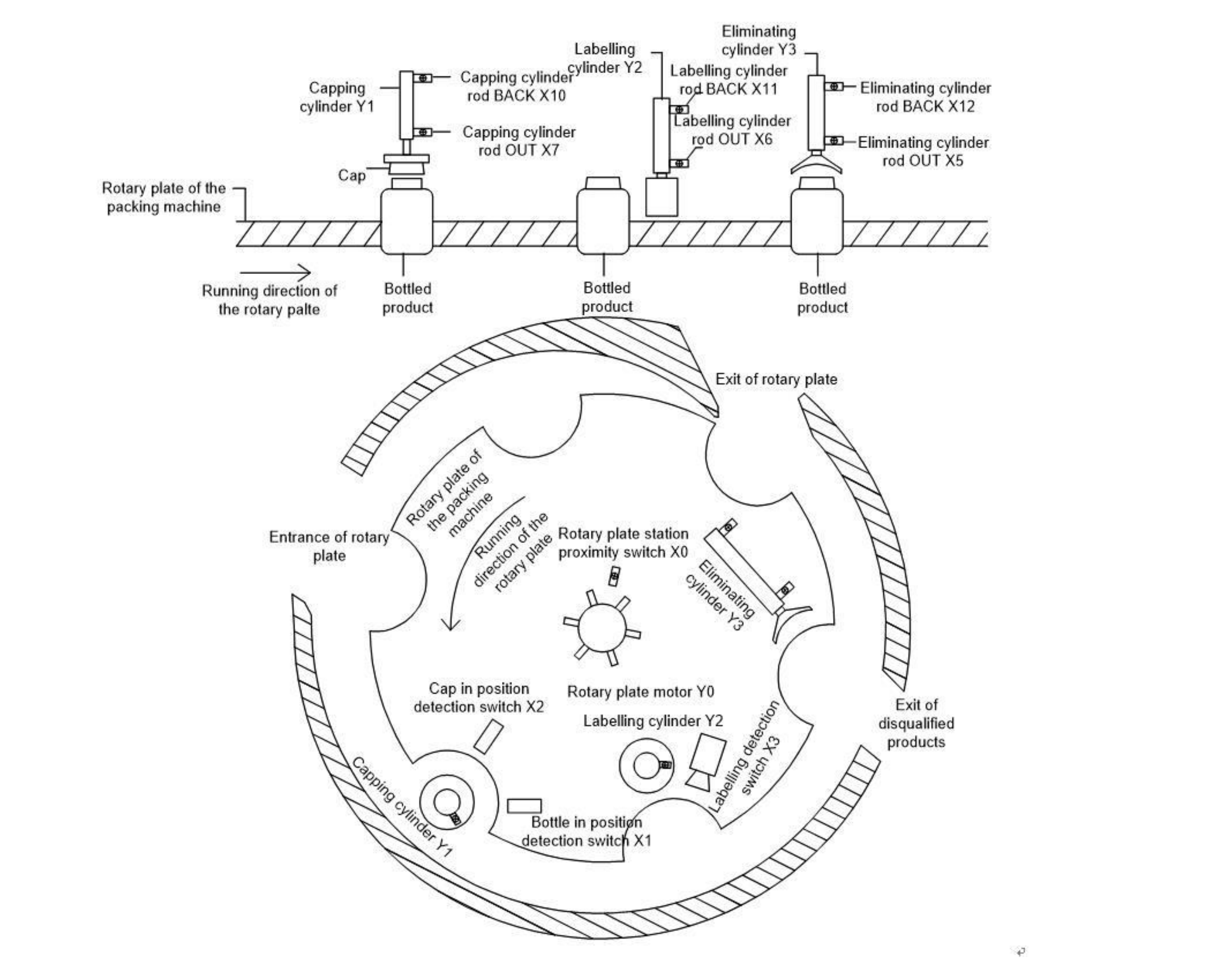

/*Reset M1-M3*/
/*D1 stands for the last product. 0: product A. 1: product B.*/
/*D2 stands for the next product. 0: product A. 1: product B.*/
/*M3 is the startup flag.*/
N0
SM1
MOV 0 D1
MOV 0 D2
RST M3
/*Start the operation*/
N1
X0
X1
M3
N2
M3
SET S0
/*Stop the operation*/
N3
X0
ZRST M1 3
X1
RST S0
ZRST S20 7
ZRST Y0 6
/*Product selection operation*/
/*M1 is the screen operation bit of HMI interface.*/
/*M2 means the two adjacent products are different.*/
It


N4
S0
S
M1
<> D1 D2
SET M2
M2
SET S20
M2
S21
N5
S20
S
Y0
Y4
TON T0 3000
T0
SET S21



7.5.3 Parallel structure

The following example is a packaging machine for bottled products. The packaging machine is to cap the bottled product and then apply the product label. In this process, the bottle caps and labels are inspected, and the defective products are removed by the subsequent rejecting device, and the genuine products can be directly sent to the next process. If there is no bottle sent from the previous process, the related capping and labeling processes will not work. The three processes are carried out at the same time, and the turntable goes one station at a time. The following figure is a schematic diagram of the production equipment.



When running, the turntable moves one station at a time, which is detected by the X0 proximity switch. At each station the carousel stops until all operations are complete. The capping, labeling, and rejecting mechanisms are all driven by the cylinder, and respectively detect the cylinder stroke in place and the cylinder return complete signal.

The table of input points and output points is shown below:

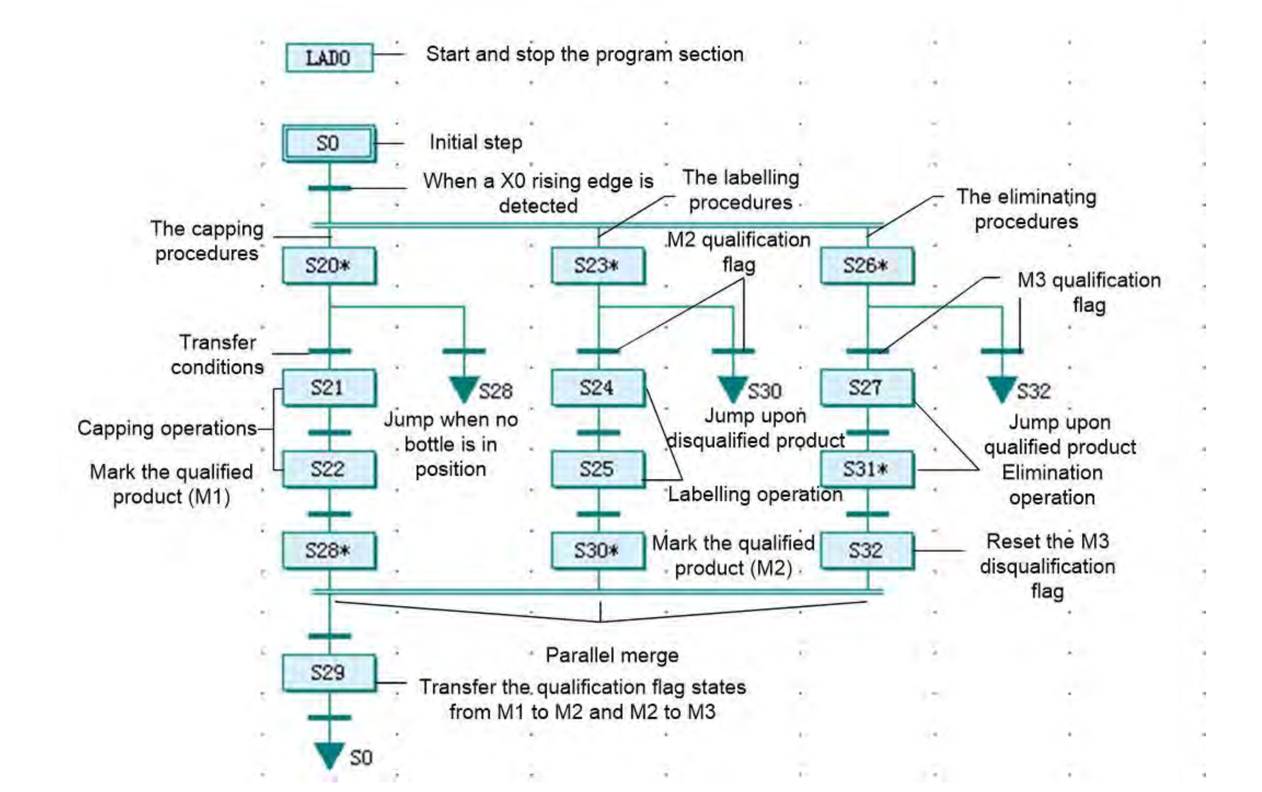
Serial number	Point address	Monitoring object		Serial number	Point address	Monitoring object
1	X0	Turntable station detection proximity switch		8	X10	Covered return trip completed
2	X1	There is a bottle detection photoelectric switch in the station		9	X11	Labeling is complete
3	X2	Cover is detecting photoelectric switch		10	X12	Eliminate the return trip
4	X3	Labeling detection device		11	Y0	Turntable Motor
5	X5	Eliminate the stroke in place		12	Y1	capped cylinder
6	X6	Labeling stroke in place		13	Y2	Labeling cylinder
7	X7	Cover travel in place		14	Y3	reject cylinder

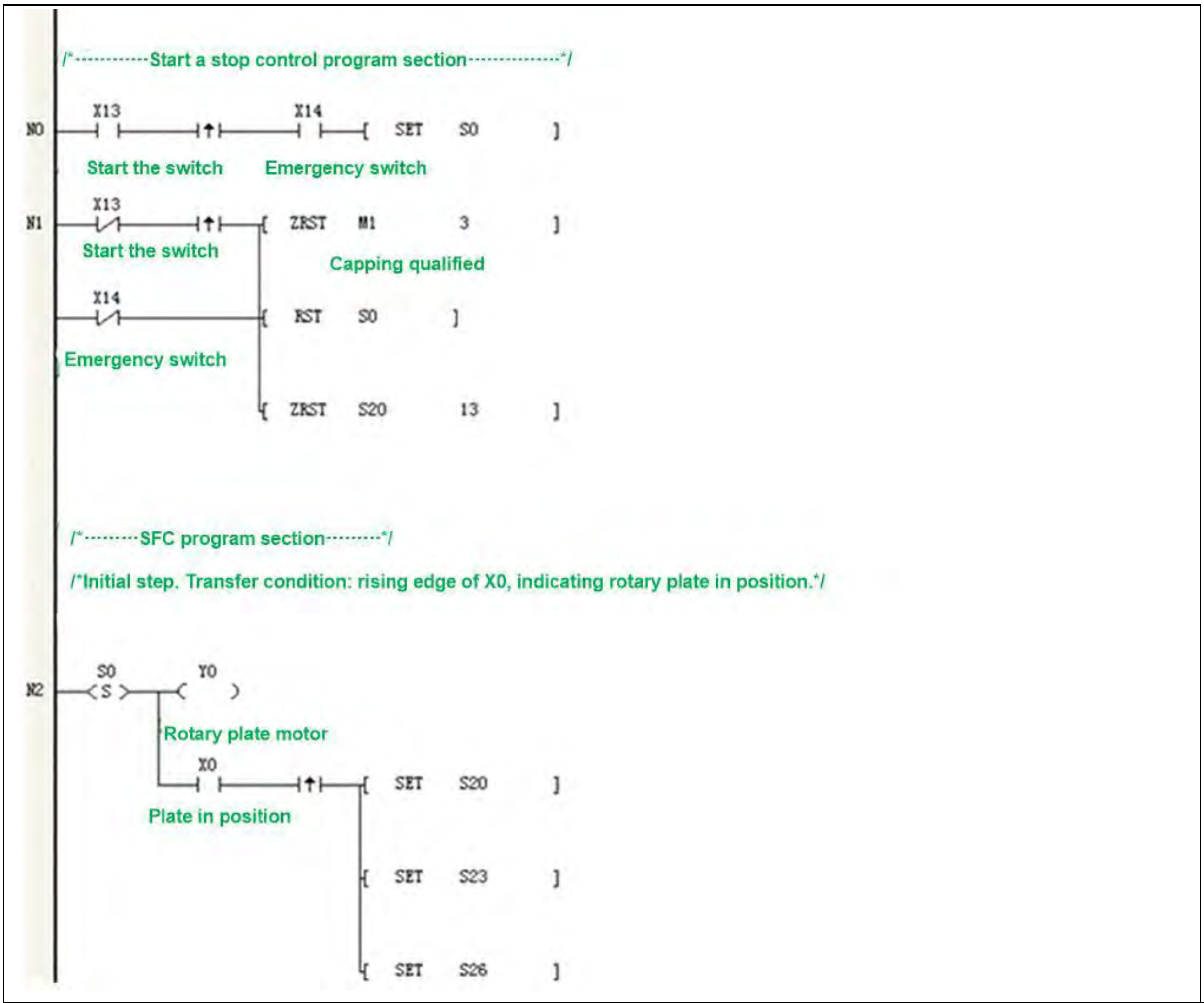
By analyzing the production process, it can be seen that this is a process with a parallel branch structure. After the turntable rotation step is completed, the operations of the three stations are performed in parallel, and the equipment will not proceed to the next step until all operations are completed. The following is the sequential function chart program and the corresponding ladder program.

In the program, M1-M3 are the genuine marks for the three processes of capping, labeling, and rejection.

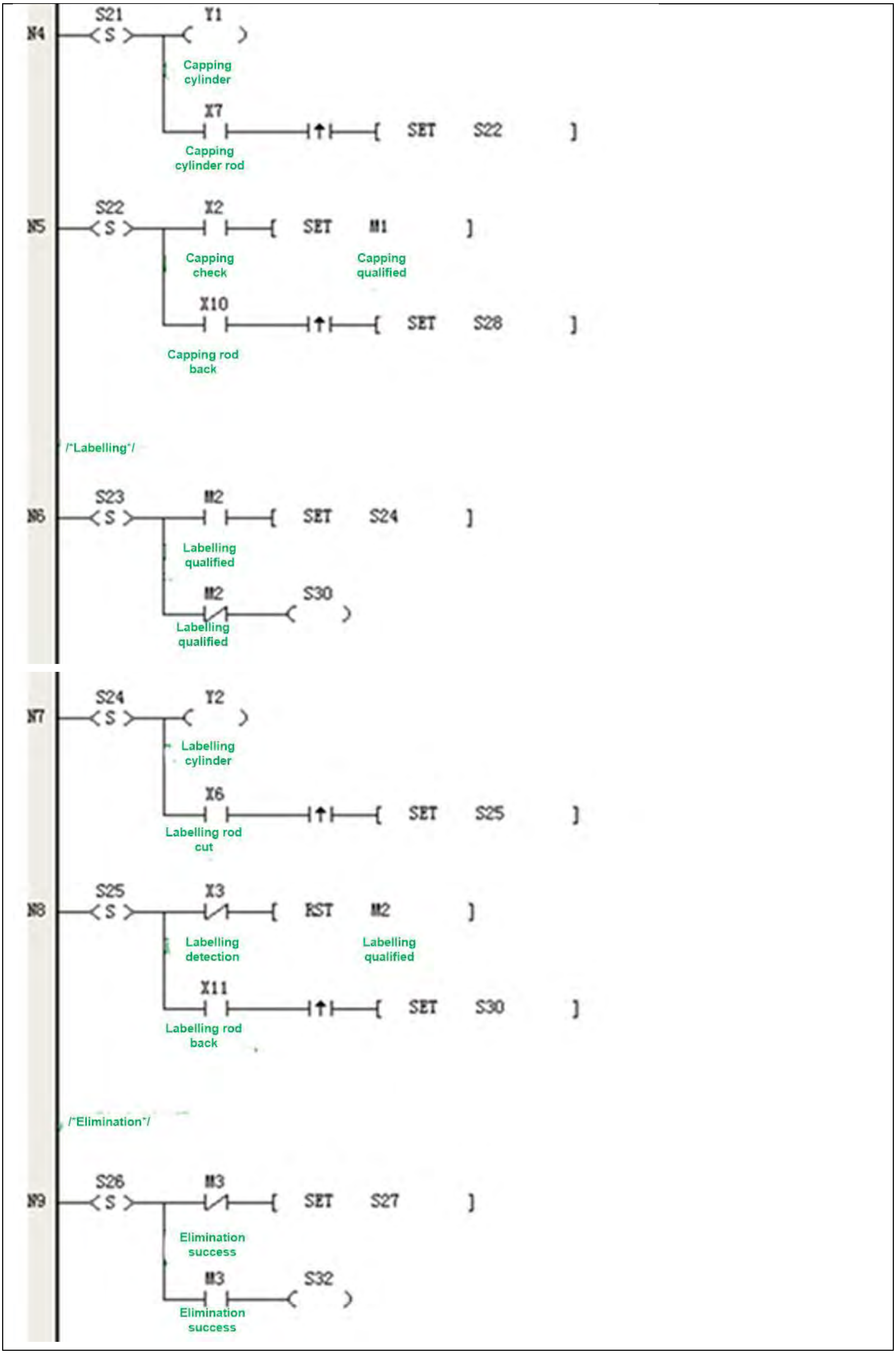
When the capping process goes to S22, X2 is used to detect whether the capping is successful (the cap is positive), and the cap is set to M1, indicating that the capping process produces genuine products; when the labeling process goes to S25, X3 is used to detect whether the label is attached If it is not correctly attached, reset M2, indicating that the labeling process produces defective products; when all the processes are over, in step S29, the state of M2 is passed to M3, and then the state of M1 is passed to M2.

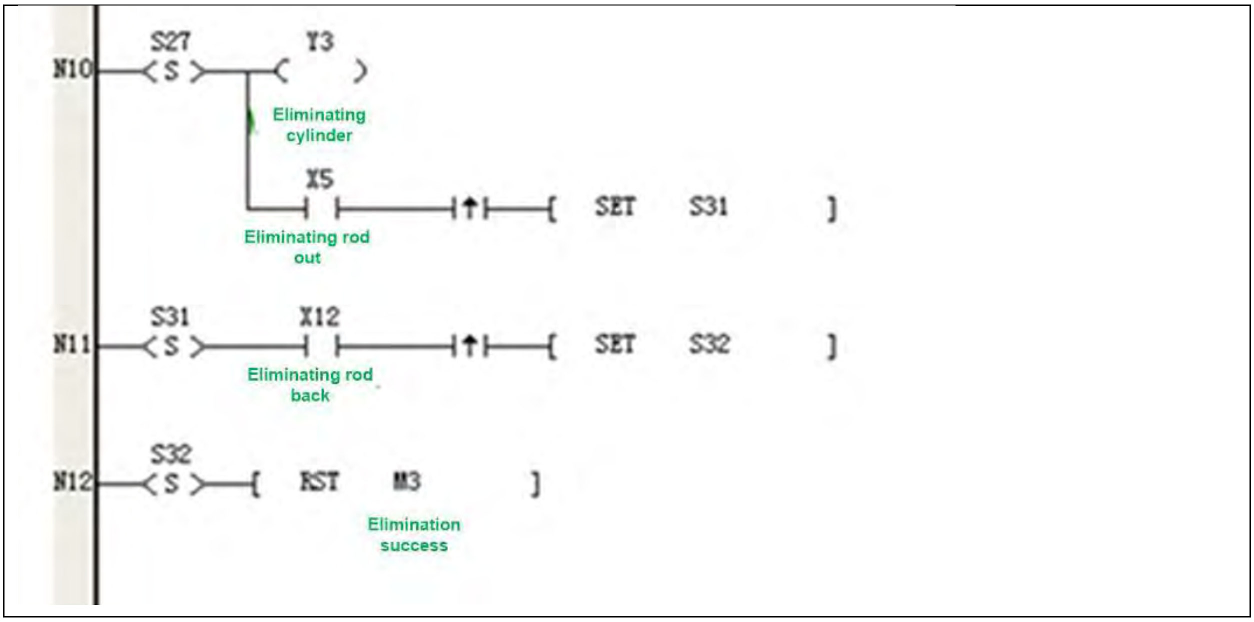
The capping process detects whether there is a bottle in place according to X1. If there is no bottle in place, the capping operation is not performed; when the labeling process starts to run, when M2 is ON, it indicates that the genuine product of the capping process has arrived, and the labeling operation is performed, and M2 is OFF to indicate that The defective product is in place, and the labeling operation is not performed; the rejection process is selected and executed according to the M3 Sign. When it is ON, it indicates the genuine product, and the rejection operation is not performed. When it is OFF, the defective product is rejected. Wait for the next step and process.

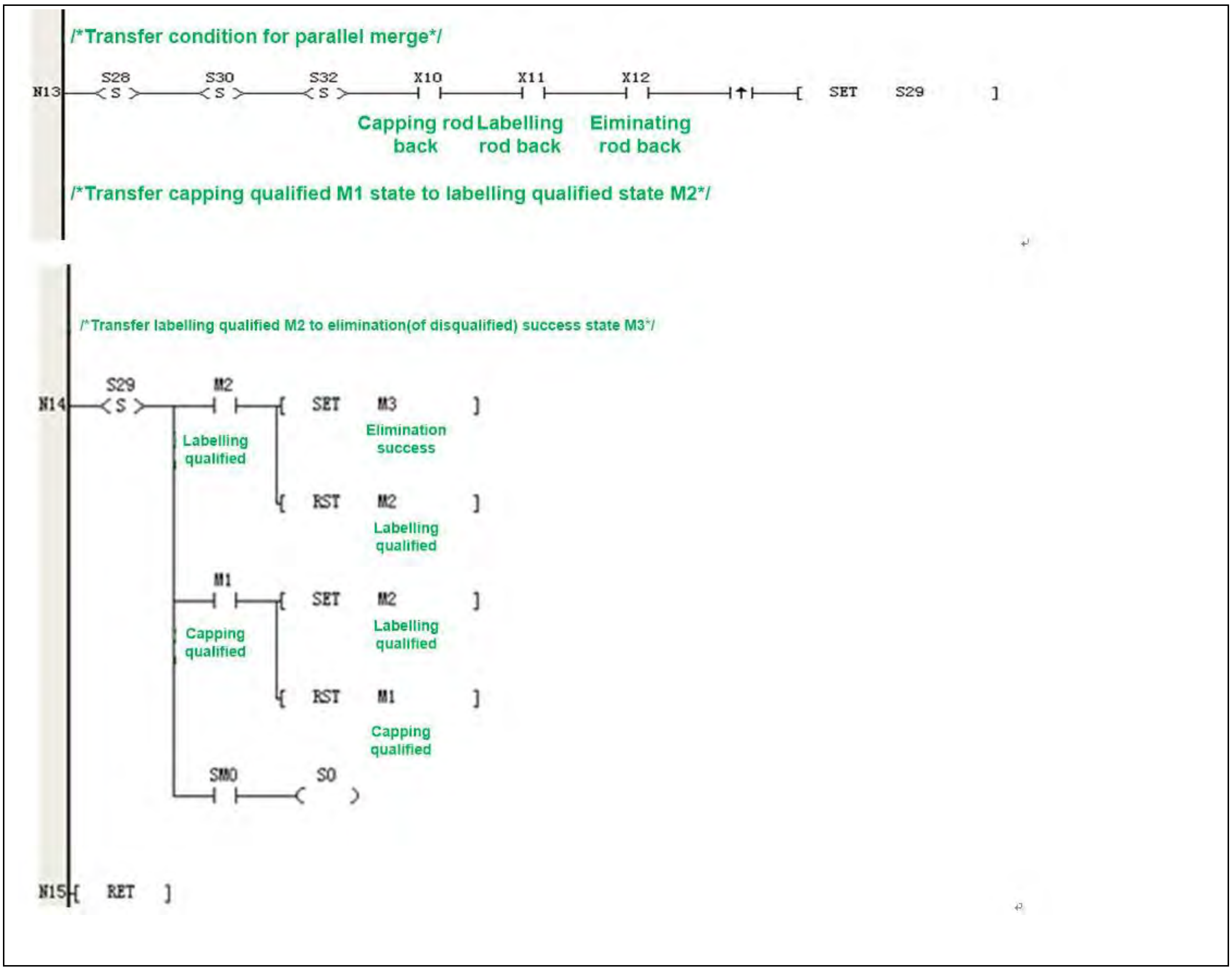



/*------------Start a stop control program section---------------*/
N0
X13
X14
SET S0
Start the switch
Emergency switch
N1
X13
ZRST M1 3
Start the switch
Capping qualified
X14
RST S0
Emergency switch
ZRST S20 13
/*---------SFC program section---------*/
/*Initial step. Transfer condition: rising edge of X0, indicating rotary plate in position.*/
N2
S0
S
Y0
Rotary plate motor
X0
SET S20
Plate in position
SET S23
SET S26


/*Capping*/
N3
S20
S
X1
SET S21
Bottle in position
X1
S28
Bottle in position


N4
S21
Y1
Capping cylinder
X7
SET S22
Capping cylinder rod
N5
S22
X2
SET M1
Capping check
Capping qualified
X10
SET S28
Capping rod back
/*Labelling*/
N6
S23
M2
SET S24
Labelling qualified
M2
S30
Labelling qualified
N7
S24
Y2
Labelling cylinder
X6
SET S25
Labelling rod cut
N8
S25
X3
RST M2
Labelling detection
Labelling qualified
X11
SET S30
Labelling rod back
/*Elimination*/
N9
S26
M3
SET S27
Elimination success
M3
S32
Elimination success


N10
S27
Y3
Eliminating cylinder
X5
SET S31
Eliminating rod out
N11
S31
X12
SET S32
Eliminating rod back
N12
S32
RST M3
Elimination success


/*Transfer condition for parallel merge*/
N13
S28
S30
S32
X10
X11
X12
SET S29
Capping rod back
Labelling rod back
Eiminating rod back
/*Transfer capping qualified M1 state to labelling qualified state M2*/
/*Transfer labelling qualified M2 to elimination(of disqualified) success state M3*/
N14
S29
M2
Labelling qualified
SET M3
Elimination success
RST M2
Labelling qualified
M1
Capping qualified
SET M2
Labelling qualified
RST M1
Capping qualified
SM0
S0
N15
RET

Chapter 8 High Speed Input

8.1 High-Speed Counter

8.1.1 High-speed counter configuration

（1） VC-B general model has 8 high-speed input ports X0~X7, of which 2 are 50KHz and 6 are 10KHz, which can realize single-phase single counting, single-phase double counting or AB-phase counting and high-speed interrupt function.

（2） VC-S series general-purpose models have 8 high-speed input ports X0~X7, support the highest pulse input frequency of 200KHz, and can realize single-phase single counting, single-phase double counting or AB-phase counting and high-speed interrupt function.

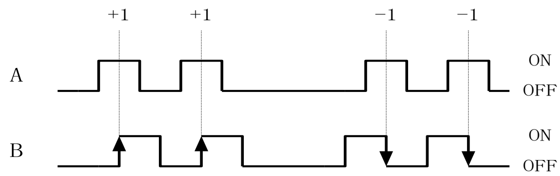
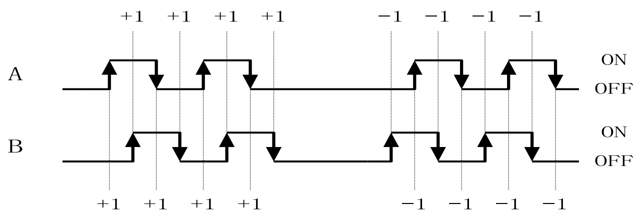
（3） The built-in high-speed counters of VEDA VC series small PLC are shown in the following table: according to the number of the counters, they are allocated to the input X0~X7.

- High-speed counter configuration table

Input point / Counter		X0	X1	X2	X3	X4	X5	X6	X7	Highest frequency KHZ		
										VC-B	VC-S	VC-P
Single-phase single-ended counting input method	C236	Increase/ decrease										
	C237		Increase/ decrease							50	200	
	C238			Increase/ decrease								
	C239				Increase/ decrease							
	C240					Increase/decrease				10		
	C241						Increase/ decrease					
	C242							Increase/ decrease				
	C243								Increase/ decrease		200	
	C244	Increase/ decrease		Reset						50	200	
	C245					Increase/decrease		Reset		10	200	
	C246	Increase/ decrease		Reset	Start up					50	200	
	C247					Increase/decrease		Reset	Start up	10	200	
Single-phase up/down count input mode	C248	Increase	Reduce							30	200	
	C249			Increase	Reduce						200	
	C250					Increase	Reduce			5	200	
	C251							Increase	Reduce		200	
	C252	Increase	Reduce	Reset						30	200 200	
	C253					Increase	Reduce	Reset		5	200	
	C254	Increase	Reduce	Reset	Start up					30	200 200	
	C255					Increase	Reduce	Reset	Start up	5	200	
Two-phase counting input method	C256	Phase A	Phase B							30	200	
	C257			Phase A	Phase B						200	
	C258					Phase A	Phase B			5		
	C259							Phase A	Phase B			
	C260	Phase A	Phase B	Reset						30	200	
	C261					Phase A	Phase B	Reset		5	200 200	
	C262	Phase A	Phase B	Reset	Start up					30	200	
	C263					Phase A	Phase B	Reset	Start up	5	200	

（4） The high-speed counter performs actions according to specific inputs in the manner shown in the above table, and processes high-speed actions according to interrupts. The counting action has nothing to do with the scan cycle of the PLC.

(5) This type of counter is a 32-bit up-counting/down-counting type counter. According to different up-counting/down-counting switching methods, it can be divided into the following four types:

Counting method	Counting action
Single-phase single-ended counting input	According to the ON/OFF of SM236～SM247, the corresponding C236～C247 is counted down/up counted respectively
Single-phase up/down count input	Corresponding to the action of up-counting input or down-counting input, the counters C248～C255 automatically up/down count. Through SM248～SM255, the current counting direction of the corresponding counter can be known. When the SM element is OFF, it counts up, and when it is ON, it counts down.
Two-phase counting input	When SM100～SM103 are set to OFF, the counters C256～C263 do automatic normal up/down counting according to the two-phase input. Through SM256～SM263, the current counting direction of the corresponding counter can be known. When the SM element is OFF, it is up counting, and when it is ON, it is down counting. The count direction is defined as follows: 
Two-phase quadruple count input	When SM100~SM103 are set to ON, the counters C256~C263 will automatically count up and down with quadruple frequency according to the two-phase input. Through SM256~SM263, the current counting direction of the corresponding counter can be known. When the SM element is OFF, it counts up, and when it is ON, it counts down. count. The count direction is defined as follows: 

8.1.2 The relationship between high-speed counter and SM element

(1) Number of special auxiliary relays for up-count/down-count switching

Type	Counter number	Increase/decrease settings
Single-phase single-ended counting input	C236	SM236
	C237	SM237
	C238	SM238
	C239	SM239
	C240	SM240
	C241	SM241
	C242	SM242
	C243	SM243
	C244	SM244
	C245	SM245
	C246	SM246
	C247	SM247
SM element is ON for down counting, OFF for up counting (default is FFO)		

(2) Special auxiliary relay number for quadruple frequency switching

Type	Counter number	Quadruple setting
Two-phase counting input	C256	SM100
	C257	SM101
	C258	SM102
	C259	SM103

(3) Number of special auxiliary relay for counting direction monitoring

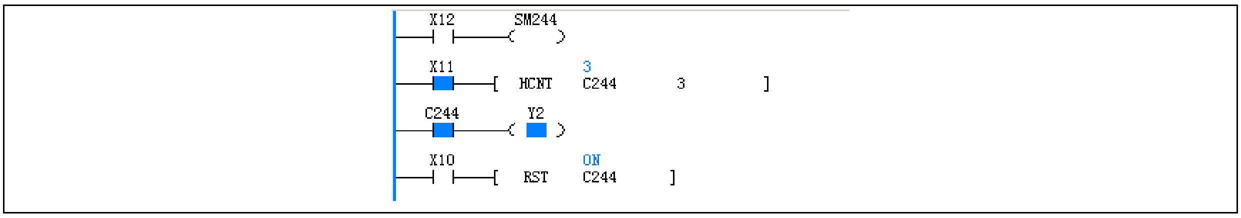
Type	Counter number	Increase/decrease monitor
Single-phase up/down count input	C248	SM248
	C249	SM249
	C250	SM250
	C251	SM251
	C252	SM252
	C253	SM253
	C254	SM254
	C255	SM255
Two-phase counting input	C256	SM256
	C257	SM257
	C258	SM258
	C259	SM259
	C260	SM260
	C261	SM261
	C262	SM262
	C263	SM263

	C260	SM100
	C261	SM102
	C262	SM100
	C263	SM102
SM element is ON for quadruple frequency mode, OFF is no frequency multiplication (default is OFF)		

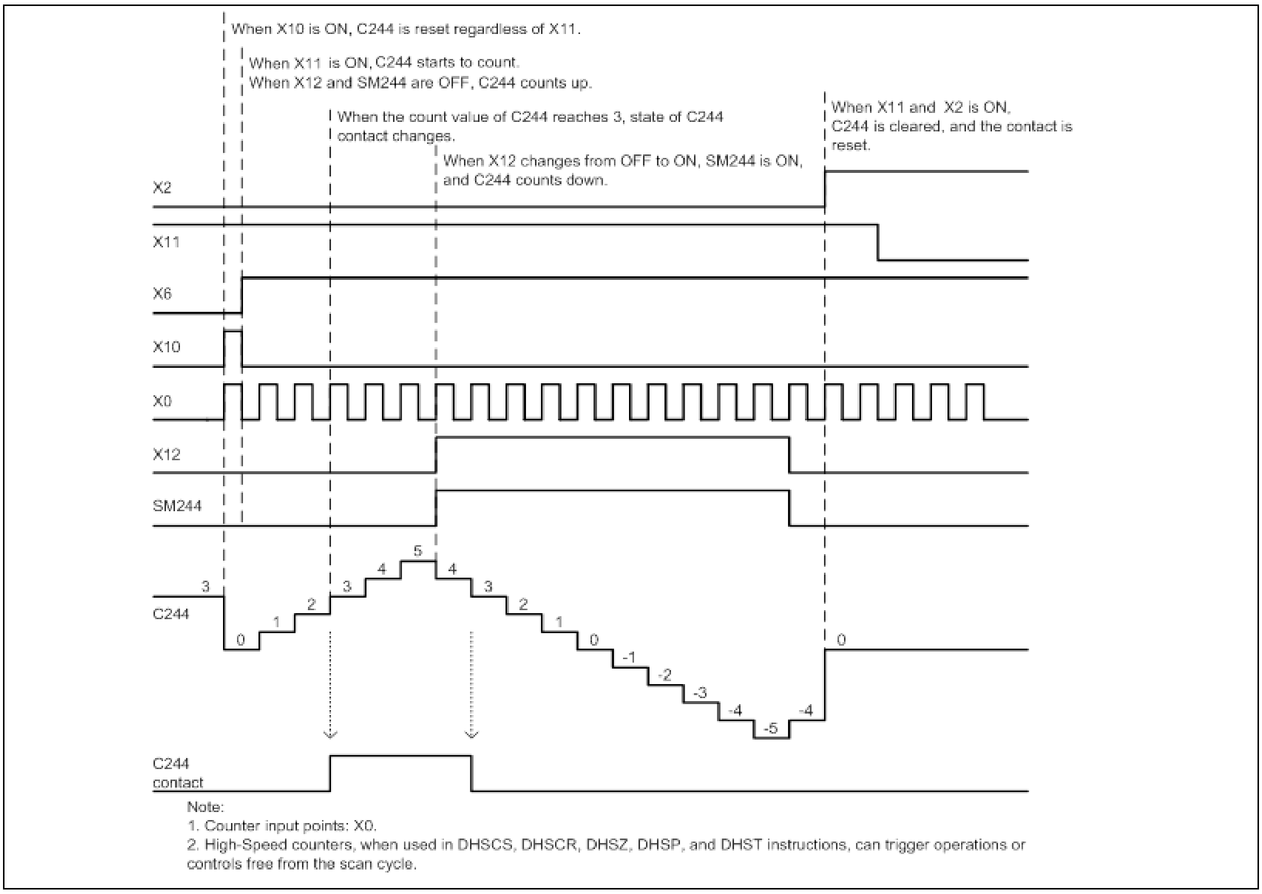
8.1.3 How to use the high-speed counter

(1) How to use the single-phase single-ended counting input high-speed counter

Features of single-phase single-ended counting input high-speed counter: pulse input only when OFF→It counts when it is ON, and the increment or decrement of the counter is determined by the corresponding special auxiliary relay SM. A sample program is shown below:



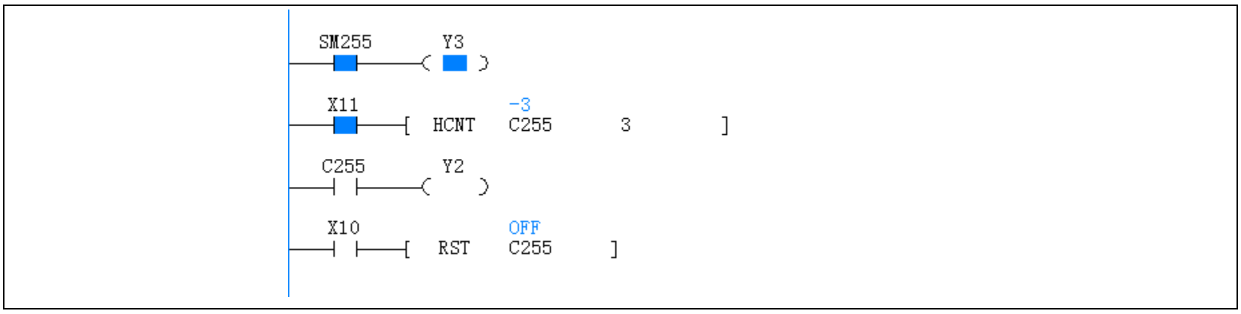
The timing operation diagram of the contacts in the program:



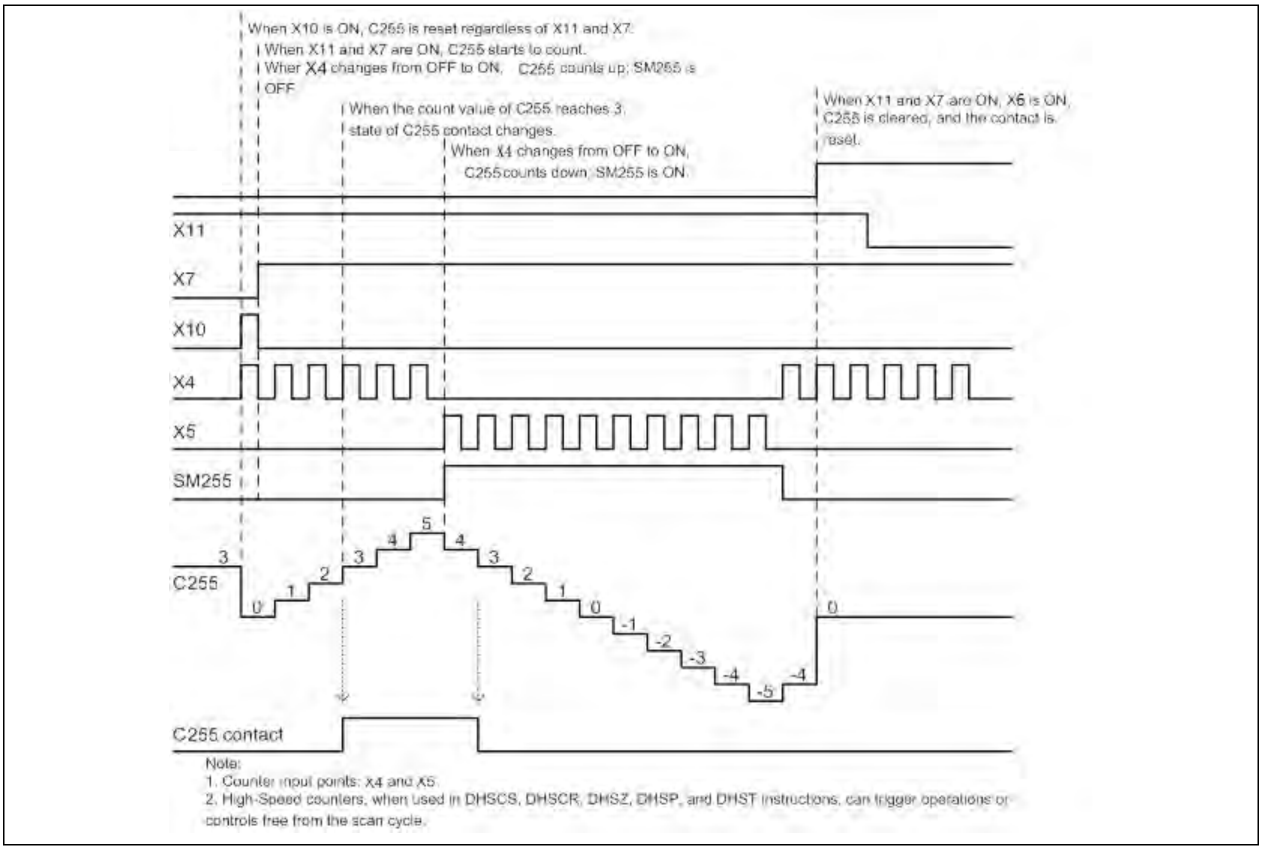
(2) How to use the single-phase up-down count input high-speed counter

Features of single-phase up-down count input high-speed counter: pulse input only when OFF→It counts when ON, and the increment or decrement of the counter is determined by two input points respectively. The corresponding special auxiliary relay SM is the current increase and decrease state of the high-speed counter.

A sample program is shown below:



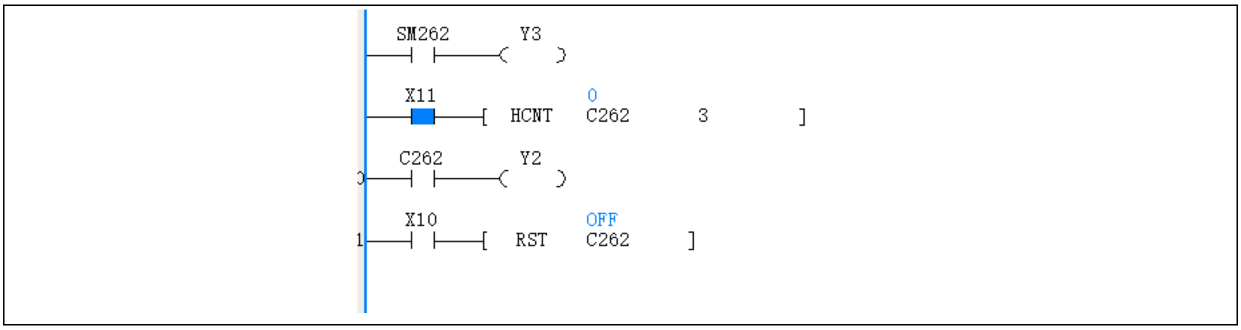
The timing operation diagram of the contacts in the program:



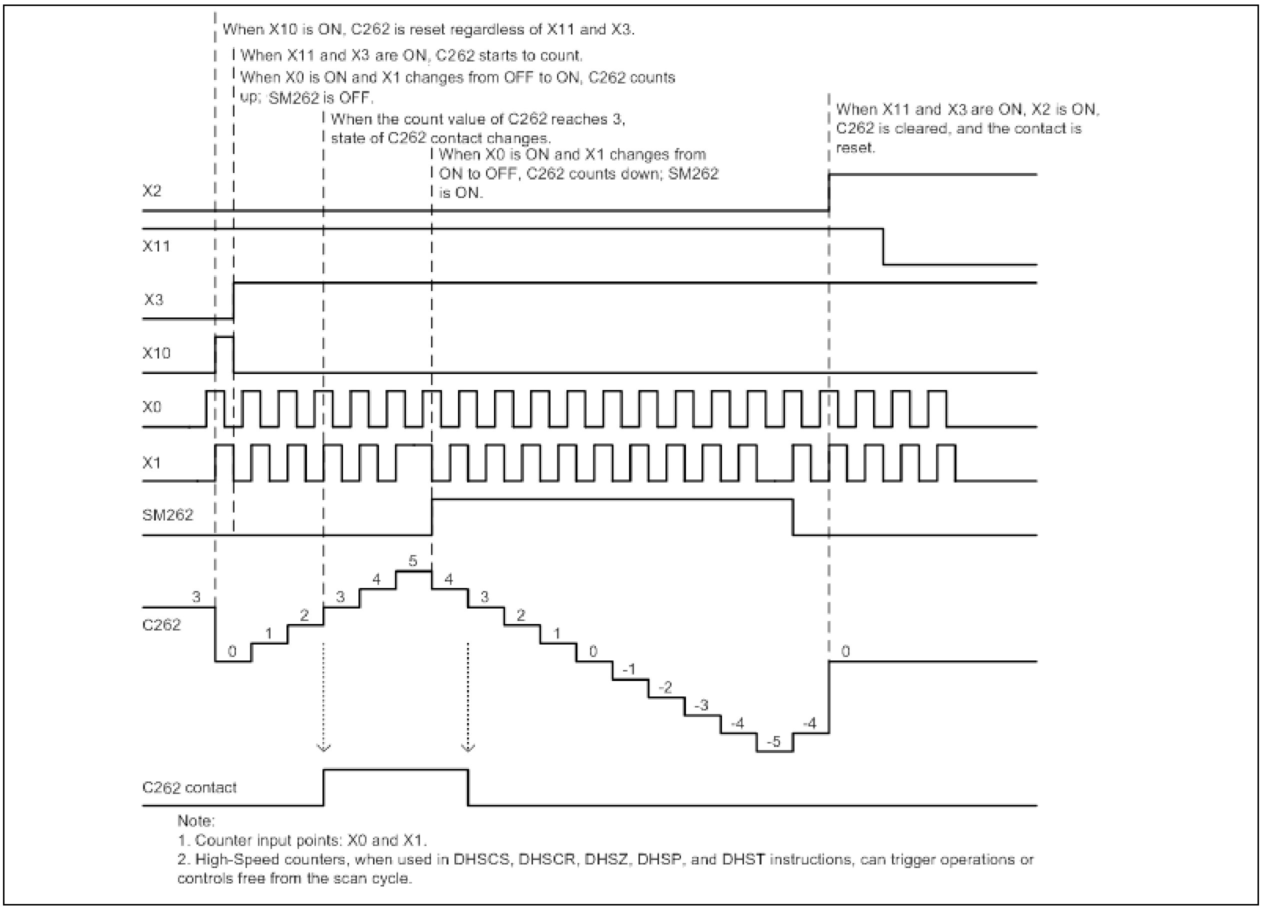
(3) How to use the two-phase counting input high-speed counter

Features of two-phase count input high-speed counter: pulse input only when OFF→When it is ON, it counts, and the increment or decrement of the counter is determined by the phase difference between the two input points. The special auxiliary relay (SM element) corresponding to the high-speed counter is the current increase or decrease state of the high-speed counter.

A sample program is shown below:

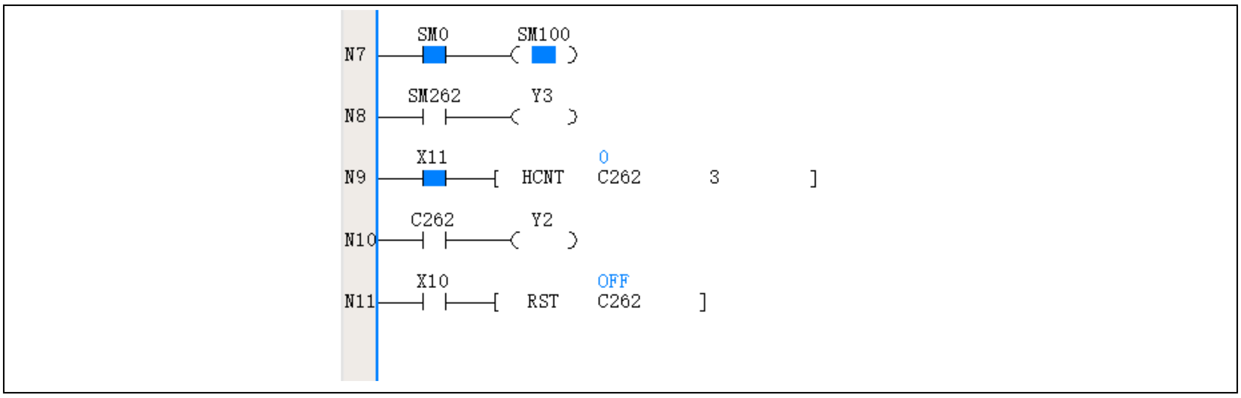


The timing operation diagram of the contacts in the program:

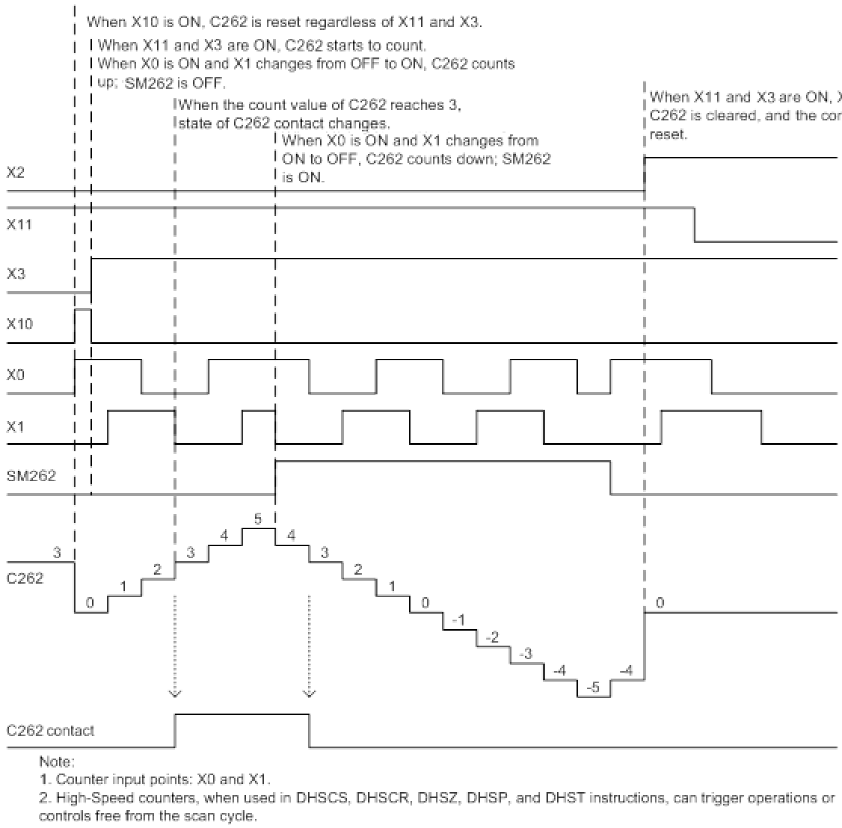


(4) How to use the double-phase quadruple frequency count input high-speed counter

Features of the dual-phase quadruple counting input high-speed counter: The pulse dual input counts at both OFF→ON and ON→OFF, and the counter is incremented or decremented by the phase difference between the two input points, respectively. The special auxiliary relay (SM element) corresponding to the high-speed counter is the current high-speed counter increment/decrement state. The program demonstration is shown below:



The timing operation diagram of the contacts in the program:



8.1.4 Precautions for high-speed counters

(1) Classification of high-speed counters;

C0~C235for software counters; C236~C263It is a hardware counter (not affected by the scan cycle), and it should be selected reasonably according to different usage modes.

(2) Frequency sum limit;

When multiple high-speed counters (hardware counter method) are used at the same time, or when high-speed counters (hardware counter method) are used together with the SPD instruction,

The total input frequency of VC-B series cannot exceed 60kHz.

(3) For VC-B series, when multiple software high-speed counters or high-speed counters and SPDs are used at the same time, the total number of input frequencies is shown in the following table:

Conditions of Use	Total number of input frequencies
DHSCS, DHSCR, DHSCI, DHSZ, DHSP, DHST are not used	≤60kHz
DHSCS, DHSCR, DHSCI, DHSP, DHST are used	≤30kHz
DHSZ has use	≤20kHz

(4) VC-S series supports the simultaneous use of 8 channels of 200KHz high-speed counters, and the total input frequency sum is 1600KHz.

🕮 Notice

Input points X0~X7 are used as input signals in functions such as high-speed counter, SPD frequency measurement command, pulse capture, and external interrupt. Since several different functions may use the same input point or points, these functions cannot be used at the same time. When programming the PLC, only one of the multiple functions corresponding to each input point can be used. If the input points of X0~X7 are used repeatedly in the user program, the user program cannot be compiled.

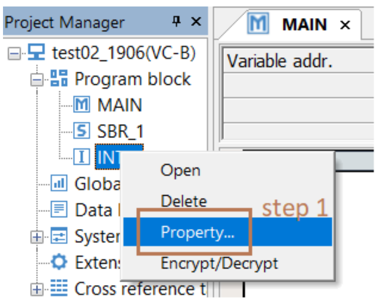
8.2 Input Interrupt

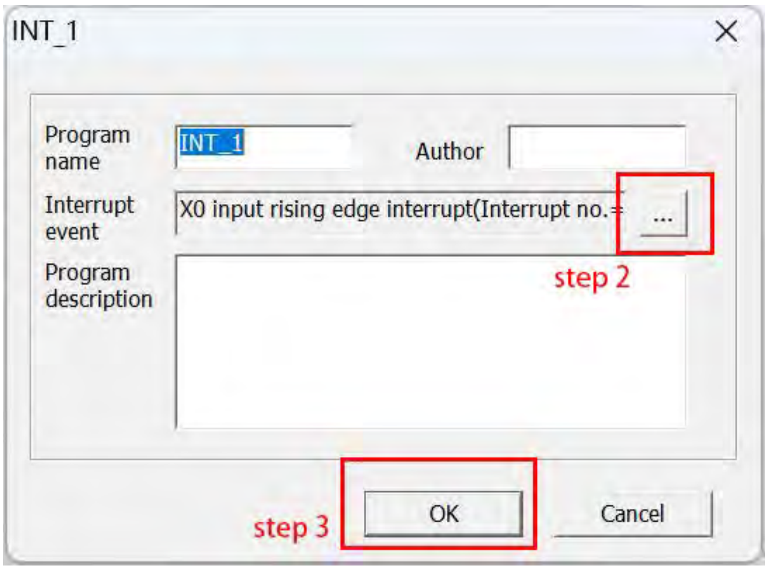
1) Input interrupts can be divided into rising edge interrupts, falling edge interrupts and counter interrupts. The interrupt numbers corresponding to X0~X7 are as follows:

Rising edge interrupt				Falling edge interrupt			
Port	Interrupt number	Corresponding interrupt source	Interrupt Enable SM	Port	Interrupt number	Corresponding interrupt source	Interrupt Enable SM
X0	0	X0 rising edge interrupt	SM25	X0	8	X0 falling edge interrupt	SM33
X1	1	X1 rising edge interrupt	SM26	X1	9	X1 falling edge interrupt	SM34
X2	2	X2 rising edge interrupt	SM27	X2	10	X2 falling edge interrupt	SM35
X3	3	X3 rising edge interrupt	SM28	X3	11	X3 falling edge interrupt	SM36
X4	4	X4 rising edge interrupt	SM29	X4	12	X4 falling edge interrupt	SM37
X5	5	X5 rising edge interrupt	SM30	X5	13	X5 falling edge interrupt	SM38
X6	6	X6 rising edge interrupt	SM31	X6	14	X6 falling edge interrupt	SM39
X7	7	X7 rising edge interrupt	SM32	X7	15	X7 falling edge interrupt	SM40

2) Interrupt use

Interrupt needs to be used in conjunction with the interrupt subroutine. Select the interrupt event in the properties of the interrupt subroutine, that is, set the interrupt event number. In the case of "interrupt enable" and the corresponding interrupt enable control SM element is ON, when the set When an interrupt event occurs, the PLC system suspends the normal execution of the main program (remember the current pause point), starts executing the interrupt subroutine from the address entry specified by ②, and returns to the pause point of the main program after the execution is completed, and continues to execute the main program. Because the PLC system takes high priority response processing to the interrupt signal, it is not affected by the scan time.





📖 Notice

After the "interrupt enable" Sign corresponding to each interrupt is turned on, the "global interrupt enable" needs to be turned on, that is, the interrupt function can be enabled only after executing the EI instruction (instruction programming: LD SM0; EI); if the global interrupt is disabled EI command, all interrupt responses are disabled. When the interrupt enable setting Sign of the input number is enabled and the input signal satisfies the interrupt setting, the corresponding interrupt subroutine will be executed. For details, please refer to Chapter 9 Interruption

8.3 External Pulse Capture Function

The hardware input points that support the external pulse capture function are X0~X7. The corresponding SM device table is as follows:

Input hardware port	Device SM
X0	SM90
X1	SM91
X2	SM92
X3	SM93
X4	SM94
X5	SM95
X6	SM96
X7	SM97

📖 Notice

1. When the external input point changes from OFF to ON, the SM device of the corresponding port is turned ON.
2. SM90 to SM97 are cleared at the start of the user program.
3. When using pulse capture, it is still necessary to abide by the limitation of the sum of the input pulse frequency of each PLC series, otherwise an abnormality may occur.
4. Using the high-speed counter or SPD command corresponding to HCNT on the same input point, regardless of whether the command is valid or not, the pulse capture is invalid after the first scan cycle.

Chapter 9 Interrupt

9.1 Interrupt Overview

Interrupt overview: Not affected by the scan cycle of the main program, the interrupt function is used as a trigger signal to execute the interrupt program (interrupt subroutine) function immediately.

In general high-speed signal application processing, the delay caused by the scanning cycle and the time deviation have an impact on the mechanical action. This situation can be obtained.

to improve.

9.2 Interrupt Event Handling Mechanism

A. Interrupt handling mechanism

1. When an interrupt event occurs and the interrupt event has been enabled, the interrupt event number will be added to the interrupt request queue record. The interrupt request queue is a first-in, first-out queue with a depth of 8.

B. The system handles interrupt requests:

1. When the system detects that the interrupt request queue is not empty, the system interrupts the normal execution flow of the user program.
2. The system queries the queue head record of the interrupt request queue. The queue head records the number of the first interrupt event, and the user interrupt program corresponding to the interrupt event number will be called and executed.
3. When the corresponding interrupt program is executed (the interrupt return instruction is executed), the interrupt request is processed, the queue head record of the request queue will be deleted from the queue, the next record in the request queue becomes the queue head record, and then The record will also move forward one position.
4. The system detects again whether the interrupt request queue is empty. If it is not empty, the above steps are executed in a loop until the interrupt request queue is empty.
5. When the interrupt request queue is empty, the system returns to the execution flow of the interrupted main program to continue execution.

C. The system only processes one interrupt request at a time. If the system is processing an interrupt request, the newly occurred interrupt event will not be responded immediately, but will be recorded at the end of the interrupt request queue, waiting for the system to process the previous interrupt. After the request, the interrupt request is processed.

D. When the number of records in the interrupt request queue reaches 8, the system will automatically shield new interrupt events, and new interrupt events cannot be added to the interrupt queue. Until all requests in the interrupt request queue have been processed and the interrupted main program has also been executed, the system will release the shield.

📖 Notice

1. The interrupt program time should not be too long, otherwise other interrupt events will be masked (interrupt request is lost), the system scan time is too long, and the execution efficiency of the main program is low.

2. It is forbidden to call other user subroutines in the interrupt program.

3. To refresh the I/O immediately in the interrupt, please use the immediate refresh instruction (REF). Note that the execution time of REF is related to the number of I/O points to be refreshed.

4. To make a certain type of interrupt event generate an interrupt request, it should be ensured that the interrupt event Sign to which the interrupt request belongs should be enabled (each type of interrupt event enable/disable control, there are related SM components. To enable a certain type of interrupt event, the corresponding SM element should be turned ON) and the global interrupt enable Sign is turned on.

5. If a corresponding interrupt request is generated, but there is no corresponding interrupt program in the user program, the system will also respond to the interrupt request, but only do empty operations.

9.3 Timed Interrupt

A. Overview of timed interrupts

1. Timing interrupt is not affected by the scan cycle, and executes an interrupt program according to the set timing value.

B. Applicable occasions

1. The timing interrupt program is mainly used in the occasions that require timing processing and the system is required to deal with it in time, such as timing sampling of analog input, and timing refresh of analog output according to a certain waveform.

C. VEDA VC series PLC provides users with 3 timing interrupt resources, as shown below:

Timed interrupt	Interrupt event number	Timing set value (SD)	Enable Control (SM)
0	22	SD47 (1~32767ms)	SM47
1	23	SD48(1~32767ms))	SM48
2	24	SD49 (1~32767ms))	SM49

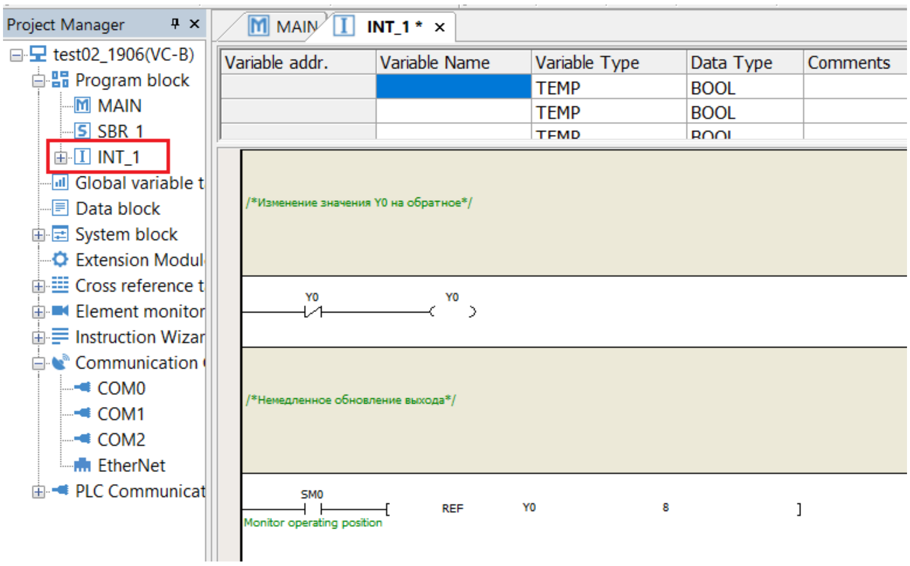
📖 Notice

1. When the timed interrupt is disabled, the timed interrupt that has been added to the interrupt queue is still executed.
2. When the timer interrupt is disabled and then enabled, the timing of the timer will start from zero.
3. If you want to change the setting of the interrupt timing value while the program is running, it is recommended to follow the steps below: first disable the timing interrupt, change the setting of the timing value, and then enable the interrupt.

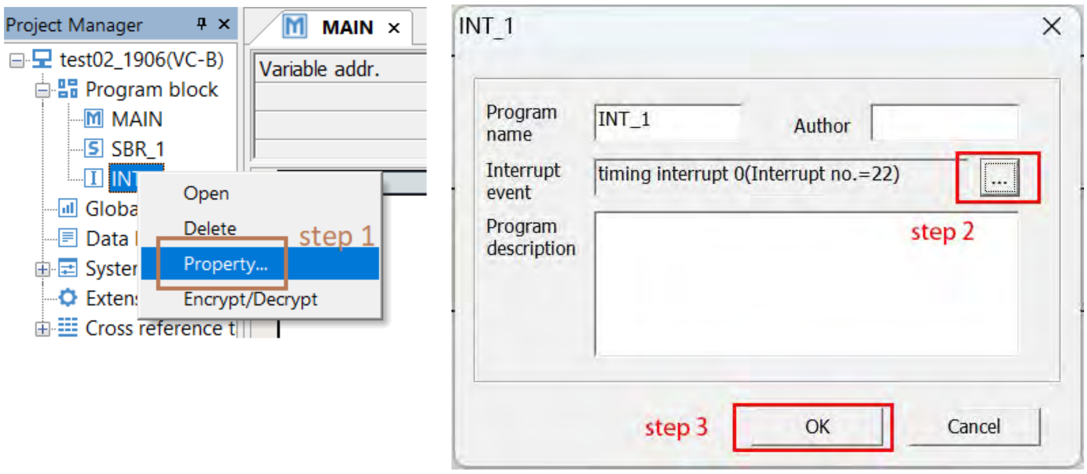
D. Demonstration of timed interruption cases

This example uses the timer interrupt 0 function to flip the output of Y0 once a second, so that Y0 has the effect of timed flickering.

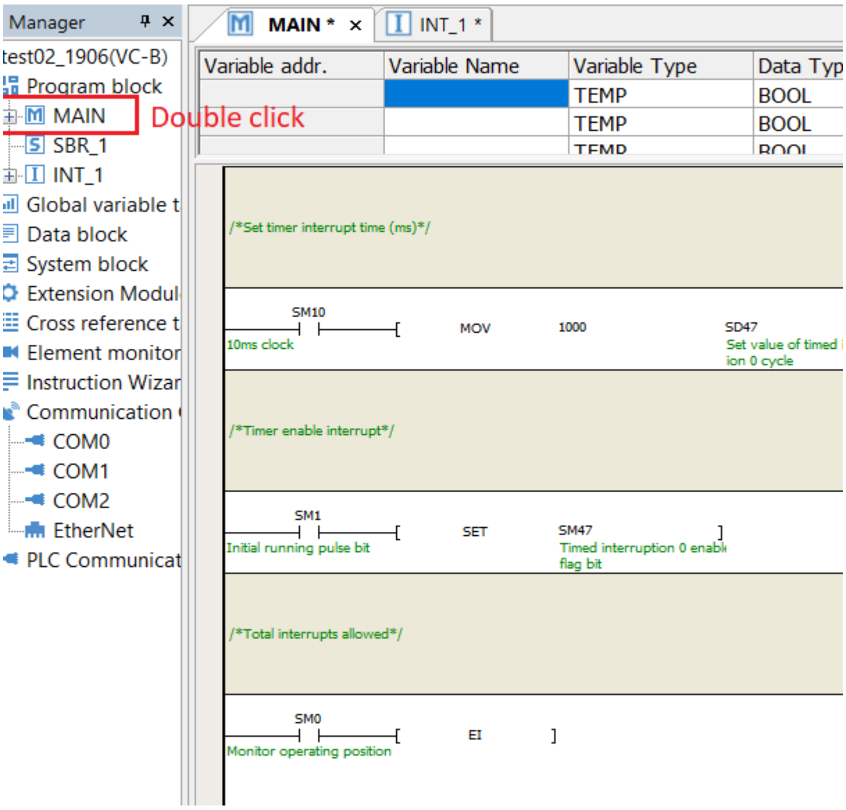
(1) Write the interrupt program, double-click "INT_1" to edit the processing code when the interrupt is triggered.



(2) Specify the corresponding interrupt event number for the interrupt program:



(3) Double-click ① in the main program, write the code for setting and enabling timing interrupts.



9.4 External Interrupt

A. Describtion of external interrupt

The interrupt subroutine is executed using the input signals of input X0~X7.

B. Applicable scenarios

Since external input signals can be processed without being affected by the operation cycle of the programmable controller, it is suitable for performing high-speed control and obtaining short-time pulses.

rush.

C. Matters needing attention

1. The maximum response frequency of the system to external signals is 1k. External events over 1k may be lost.

2. The rising edge and falling edge interrupts can be used at the same time for the same port. All external interrupts are valid only when the total interrupt control EI is valid and the corresponding interrupt enable SM is valid.

3. The single input pulse frequency of VC-B series X0~X7 is less than 10KHz; the single input pulse frequency of VC-S series X0~X7 is less than 200KHz.

The external interrupt numbers are shown in the following table:

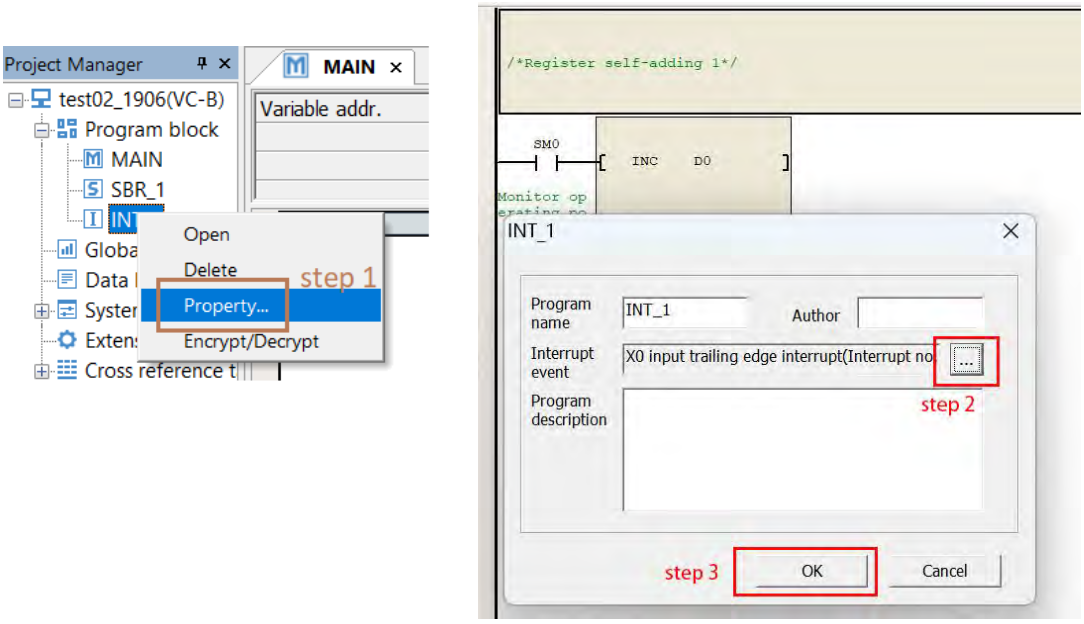
Rising edge interrupt				Falling edge interrupt			
Port	Interrupt number	Corresponding interrupt source	Interrupt Enable SM	Port	Interrupt number	Corresponding interrupt source	Interrupt Enable SM
X0	0	X0 rising edge interrupt	SM25	X0	8	X0 falling edge interrupt	SM33
X1	1	X1 rising edge interrupt	SM26	X1	9	X1 falling edge interrupt	SM34
X2	2	X2 rising edge interrupt	SM27	X2	10	X2 falling edge interrupt	SM35
X3	3	X3 rising edge interrupt	SM28	X3	11	X3 falling edge interrupt	SM36
X4	4	X4 rising edge interrupt	SM29	X4	12	X4 falling edge interrupt	SM37
X5	5	X5 rising edge interrupt	SM30	X5	13	X5 falling edge interrupt	SM38
X6	6	X6 rising edge interrupt	SM31	X6	14	X6 falling edge interrupt	SM39
X7	7	X7 rising edge interrupt	SM32	X7	15	X7 falling edge interrupt	SM40

D. Demonstration of external interrupt cases

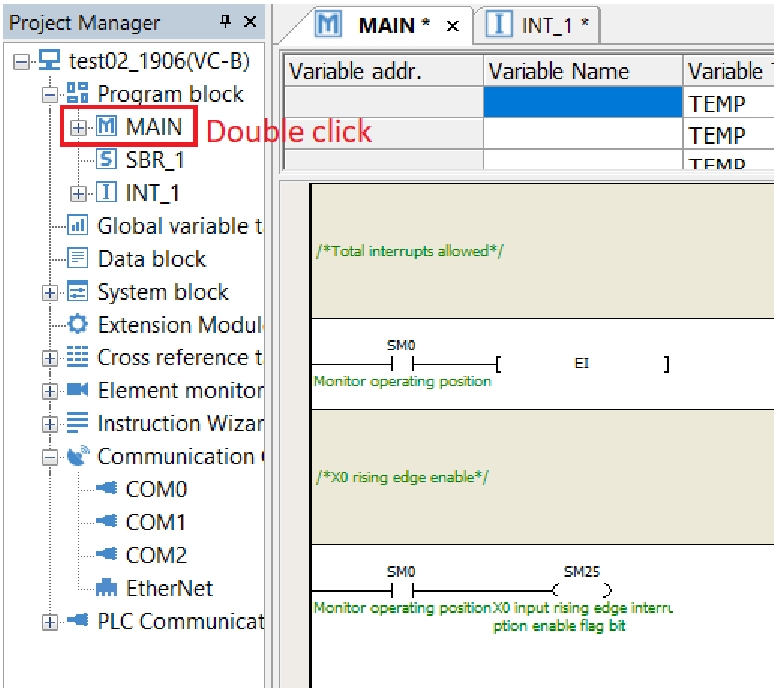
In this example, the external interrupt 0 function corresponding to X0 is used, and D0 is self-added according to the input event of the rising edge of X0.

(1) Write an interrupt program, and the D0 register will increase by 1 every time the interrupt is entered. For each interrupt, the corresponding interrupt number must be selected.

The specific operation is shown in the following figure.



(2) Double-click ① to write the EI instruction in the main program, and make the interrupt enable SM25 corresponding to the X0 input rising edge interrupt valid.



(3) Program description: When the rising edge of the X0 signal is valid, the interrupt service routine is executed, and the D0 register is incremented by 1.

9.5 High-Speed Counter Interrupt

A. High-speed counter interrupt description

Interrupt using the current value of the high-speed counter HCNT instruction. Used together with the DHSCI instruction, when the current value of the high-speed counter reaches the specified value of DHSCI

When the interrupt program is executed.

B. Conditions of use

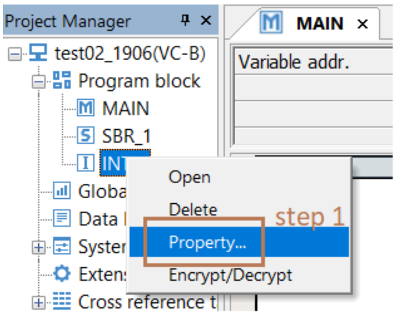
The high-speed counter interrupt must be used in conjunction with the high-speed HCNT drive instruction or the DHSCI instruction to generate a high-speed counter interrupt according to the count value of the high-speed counter. In the high-speed interrupt program, the user can write programs related to external pulse input. All high-speed counter interrupts (33 to 40) are only valid when the total interrupt control EI is valid and the corresponding interrupt enable Sign is valid. Interrupt numbers are shown in the table below

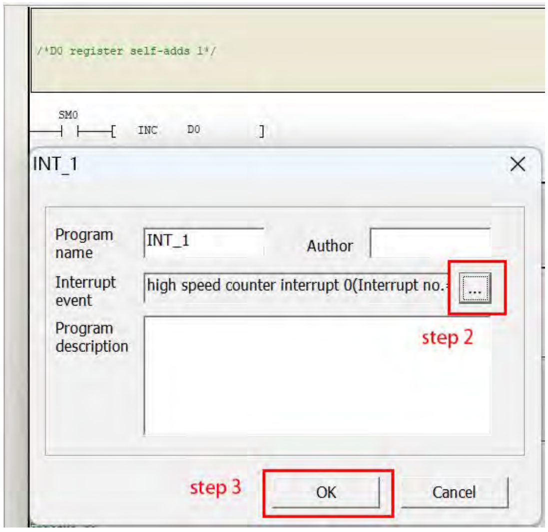
Interrupt event number	Corresponding to the interrupt event	Interrupt Enable Control SM
33	High-speed counter interrupt 0	SM58
34	High-speed counter interrupt 1	SM58
35	High-speed counter interrupt 2	SM58
36	High-speed counter interrupt 3	SM58
37	High-speed counter interrupt 4	SM58
38	High-speed counter interrupt 5	SM58
39	High-speed counter interrupt 6	SM58
40	High-speed counter interrupt 7	SM58

C. High-speed counter interrupt case demonstration

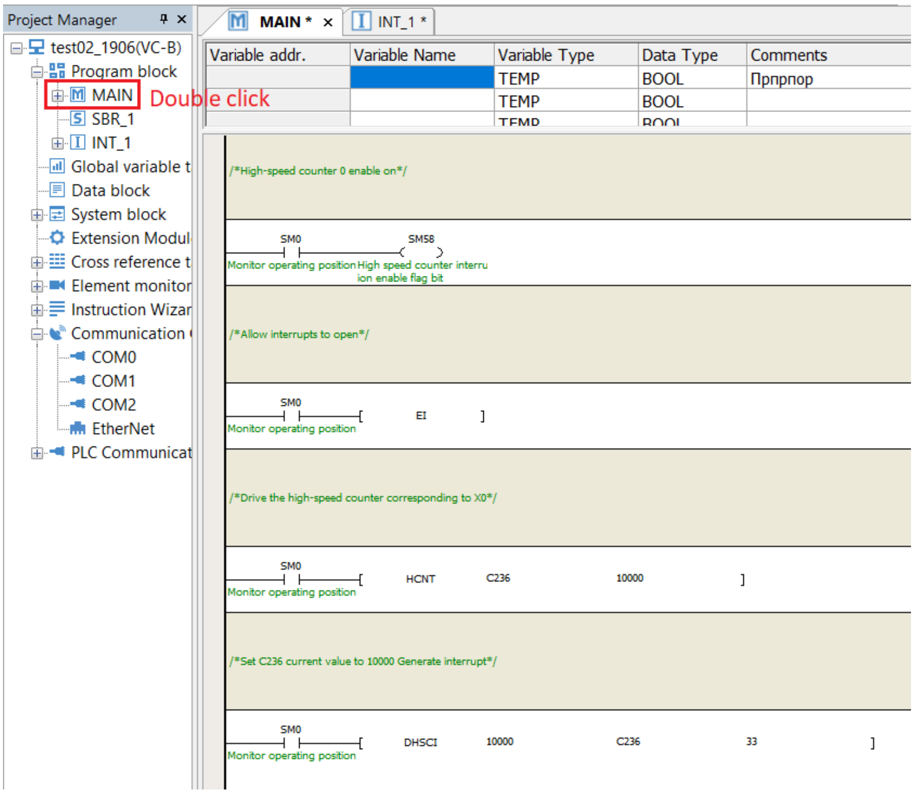
This example uses the interrupt instruction function of the high-speed counter corresponding to X0. When the value of the high-speed counter C236 reaches the data specified by DHSCI, the interrupt program with interrupt number 33 is responded to, and the D0 register in the interrupt program is incremented by 1.

1. Program the interrupt subroutine. The corresponding interrupt number must be selected for each interrupt subroutine, as shown in the following figure.





2.Double-click ① to enter the main program and write the EI instruction to enable the interrupt enable SM58 of the high-speed counter interrupt. Drive the high-speed counter C236, and drive the high-speed counter interrupt instruction.



3. Program description: When the high-speed counter C236=1000, an interrupt is generated, the interrupt service routine is executed, and the D0 register is incremented by 1.

9.6 Pulse Output Completion Interrupt

A. Pulse output completion interrupt

1. When the enable Signs SM50, SM51, and SM52 (corresponding to Y0~Y2 respectively) of VC-B series are ON, in the positioning commands such as PLSY, PLSR, DRVA, DRVI, etc., the pulse output completion can be interrupted. The related processing is carried out in the interrupt subroutine.

2. When the enable Signs SM50, SM51, SM52, SM53, SM54, SM56, and SM57 (corresponding to Y0~Y7) are ON, the VC-S series can implement pulses in positioning commands such as PLSY, PLSR, DRVA, and DRVI. The output completes interrupt, and the user can perform related processing in the interrupt subroutine.

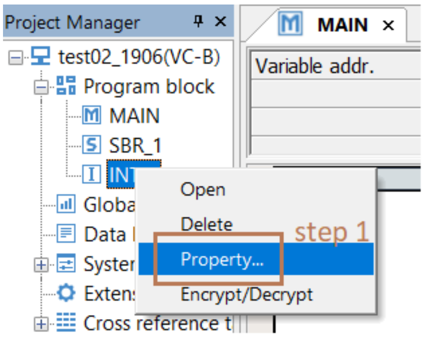
B. The interrupt enable relationship corresponding to the pulse completion interrupt is shown in the table:

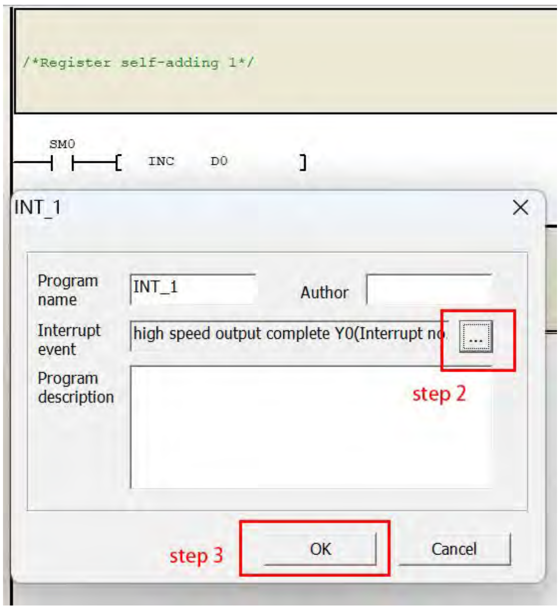
Port	Interrupt event number	Corresponding to the interrupt event	Interrupt Enable Control SM
Y0	25	High-speed output complete interrupt 0	SM50
Y1	26	High-speed output complete interrupt 1	SM51
Y2	27	High-speed output complete interrupt 2	SM52
Y3	28	High-speed output complete interrupt 3	SM53
Y4	29	High-speed output complete interrupt 4	SM54
Y5	30	High-speed output complete interrupt 5	SM55
Y6	31	High-speed output complete interrupt 6	SM56
Y7	32	High-speed output complete interrupt 7	SM57

A. Program demonstration⊠

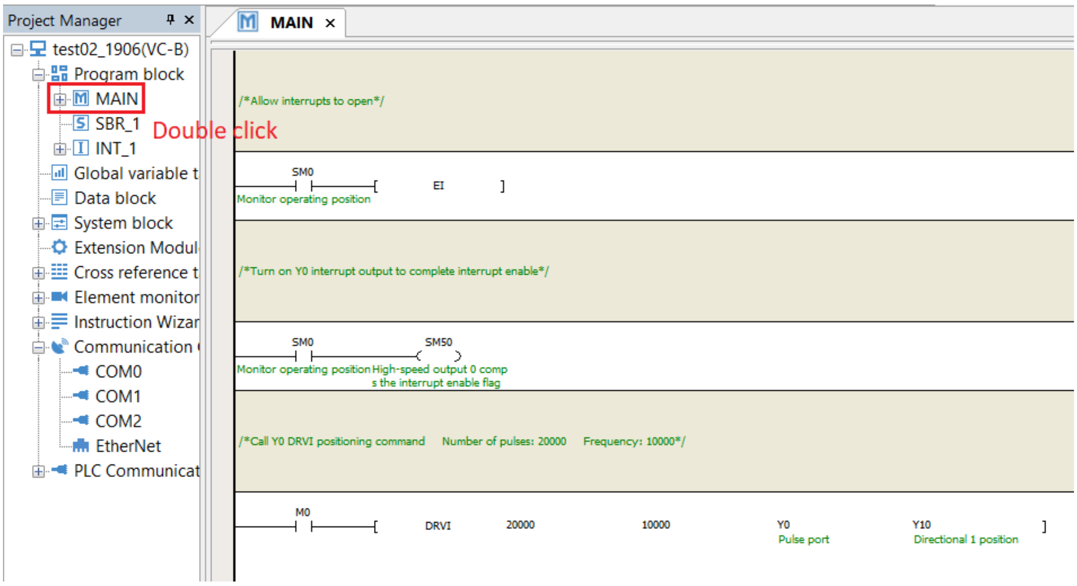
Using the interrupt command function of the high-speed pulse output corresponding to Y0, when the high-speed pulse output pulse of Y0 is completed, the interrupt program with the interrupt number 25 will be responded to, and the D0 register in the interrupt program will be incremented by 1.

1. Code function in the interrupt program (INT_1): programming required to control the code in the interrupt. As shown below;





2. Code function in the main program (MAIN): make the total interrupt EI valid, and at the same time make the Y0 output completion interrupt enable Sign SM50 valid, and call the DRVI instruction.



3. Program description: When M0 is ON, Y0 starts to send 20,000 pulses at a frequency of 10000HZ. When the pulse is sent, an interrupt is generated and the interrupt program is executed, so that the D0 register executes a self-increment by

9.7 Serial Port Interrupt

A. Serial port interrupt description

When the serial port is in the free port protocol mode, the system will generate interrupt events according to the sending and receiving events of the serial port.

B. Applicable occasions

For each serial port, the system provides the user with 2 interrupt resources. The serial port interrupt program is mainly used in occasions that require special processing of serial port frame receiving and sending operations, and the system is required to process it in time. It can respond immediately to sending and receiving without being affected by the scanning time Some processing of finished frames.

C. Matters needing attention

Set the ON/OFF state of the corresponding SM element to enable/disable the serial port interrupt. When the serial port interrupt is disabled, the serial port interrupt that has been added to the interrupt queue will still be executed. In the character sending interrupt processing subroutine, please do not call the serial port sending instruction (XMT) after the normal power flow, which may cause the interrupt subroutine to be nested and block the execution of the user program.

D. Frame receiving and frame sending interrupts refer to the interrupt events triggered after the completion of the serial port sending command (XMT) and the serial port receiving command (RCV).

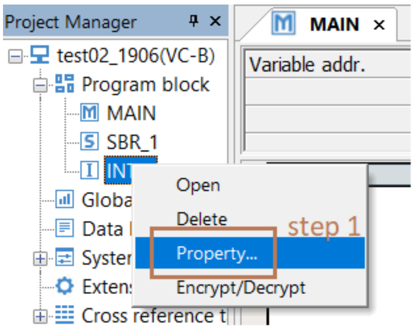
List of serial port interrupt resources:

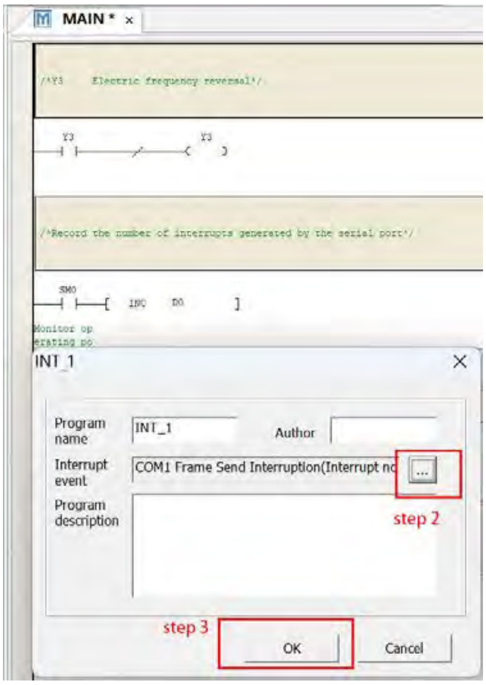
Interrupt event number	Corresponding to the interrupt event	Interrupt enable control SM element
16	Frame send interrupt of COM0	SM41
17	Frame receive interrupt on COM0	SM42
18	Frame send interrupt of COM1	SM43
19	Frame receive interrupt on COM1	SM44
20	Frame send interrupt of COM2	SM45
21	Frame receive interrupt on COM2	SM46

F. Demonstration of serial port interrupt program

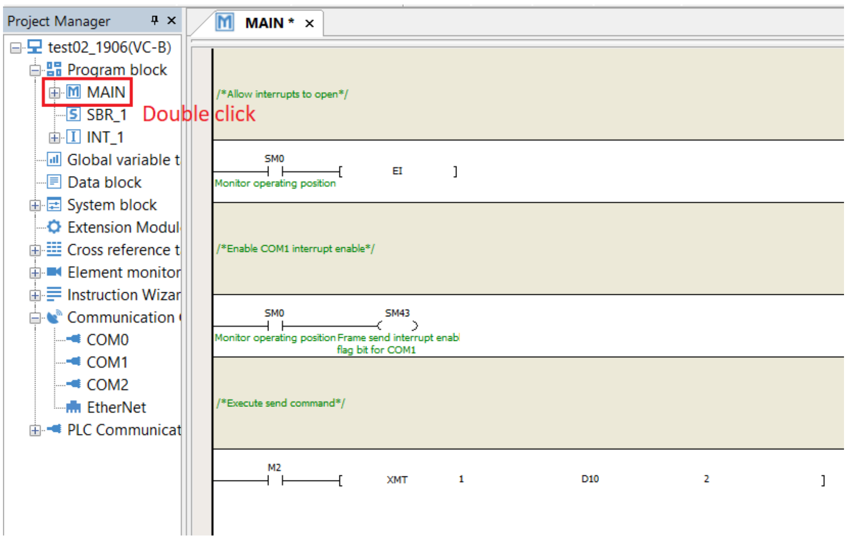
This example uses the serial port frame sending interrupt function. After each frame is sent, the output of Y3 is flipped once, so that Y3 has the effect of flickering according to the frequency of the character sending frame.

1. Write an interrupt program, write out the processing code when the serial port sends a frame and the interrupt is triggered, and configure the interrupt number event corresponding to the COM1 serial port. As shown below:





2. Double-click ① In the main program, make the total interrupt EI effective, and enable the serial port to send the frame interrupt code and send data command.



3. Program description: When M2 is ON, the COM1 port starts to send data. When the output transmission is completed, an interrupt is generated and the user interrupt program is executed, so that the output of Y3 is 0N, and the value of the D0 register is incremented by 1. (For details on the use of serial port interrupts, please refer to the tenth communication function guide)

Chapter 10 Communication Function

10.1 Communication Resources

A. VC-B series communication resources

VC-B series PLC main module has 2 integrated serial ports COM0, COM1, and 1 channel USB interface, among which COM0 and USB support programming port protocol and firmware upgrade; can support 1 channel COM2 expansion, 3 serial ports have Modbus communication protocol, N: N communication protocol, user-defined free port protocol can also be used. (Note: COM0 does not support N: N communication and free port protocol)

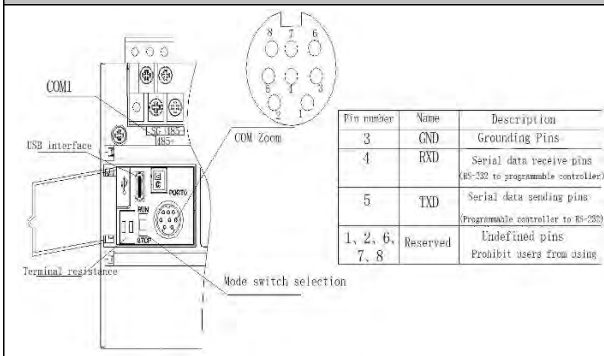
B. VC-S series communication resources

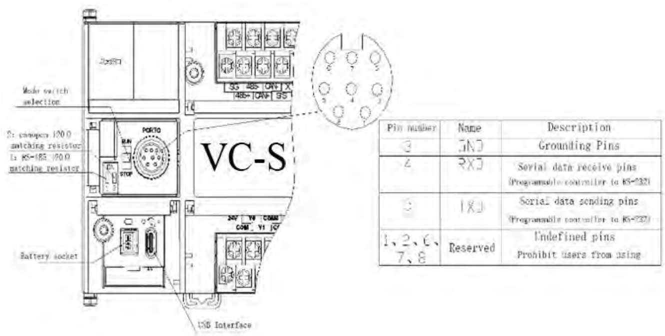
The VC-S series PLC main module has 2 integrated serial ports COM0, COM1, and 1 USB interface, as well as its own CAN communication interface and Ethernet interface, among which COM0 and USB support programming port protocol and firmware upgrade; can support expansion of 1 COM2 , 3 serial ports have Modbus communication protocol, N: N communication protocol, and can also use user-defined free port protocol. Support CANopen protocol, Modbus-TCP (Note: COM0 does not support N: N communication and free port protocol)

C. Hardware and communication connections

The COM0 hardware standard is RS232, and the interface end is an 8-hole round head female socket. The interface definition is as follows

VC-B series	VC-S series





D. Applicable baud rate of VEDA VC series small PLC

Letter of agreement	Applicable baud rate
Free mouth agreement,	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200
Modbus communication protocol	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200
N: N communication protocol	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200

E. Communication protocols supported by VEDA VC series small PLCs:

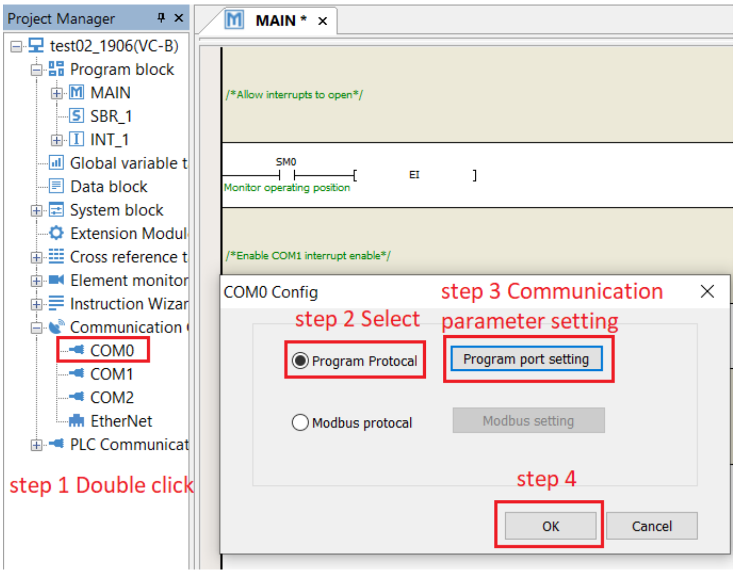
Main module	Communication port	Communication port type	Supported Protocols
VC-B	COM0	RS232	Programming port protocol, Modbus communication protocol (slave)
	COM1	RS485	Freeport protocol, Modbus communication protocol (master, slave), N: N communication protocol (master, slave)
	COM2	RS485	Freeport protocol, Modbus communication protocol (master, slave), N: N communication protocol (master, slave)
		USB	programming port protocol

In addition, RUN-STOP of VEDA VC series small PLC can force COM0 to be converted to programming port protocol.

10.2 Programming Port Communication Settings

The COM0 programming port protocol is a special protocol for the communication between the host computer software VEDA PCT and the main module, and is not open to the outside world. You can use USB, network port, serial port to communicate with the main module, only one of them can be selected for monitoring, and cannot be used at the same time. (VC-B series does not support network port)

(1) In the Connect, select the communication configuration option, and select the programming port protocol in the corresponding parameter setting. The communication parameters generally keep the default, as shown in the following figure:



10.3 Free Port Communication Settings

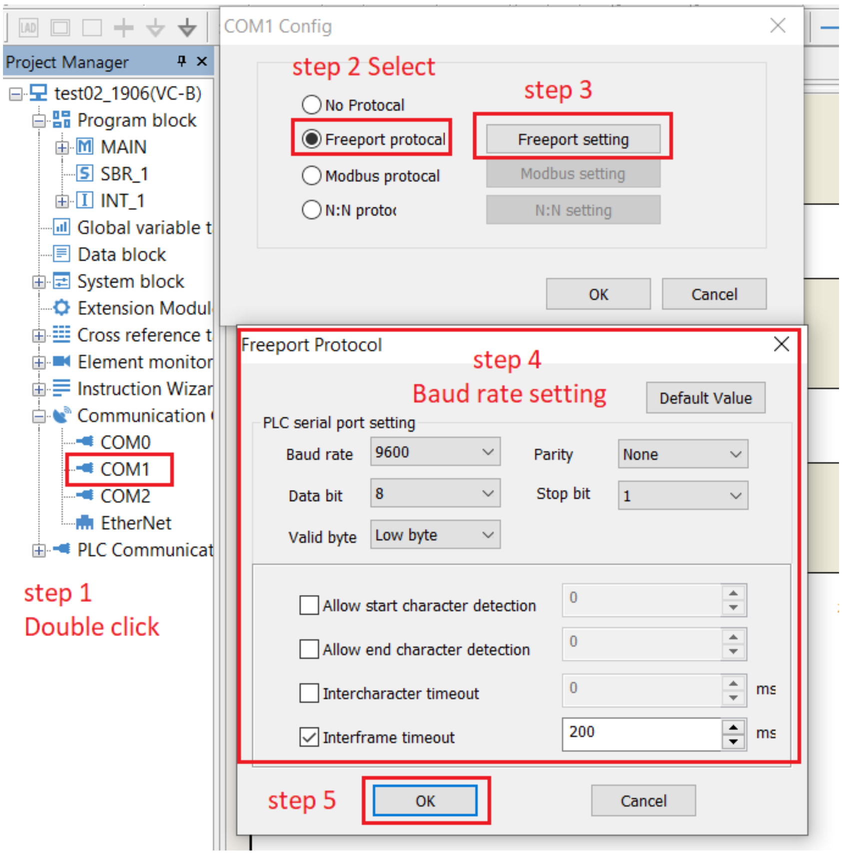
10.3.1 Introduction

(1) Free port protocol is a communication method of user-defined data file format, which can send and receive data by instructions. The Free port protocol supports both ASCII and binary data formats. Free port communication can only be used when the PLC is in RUN mode.

(2) COM1/COM2 supports the free port protocol; the communication commands of the free port include XMT (free port sending command) and RCV (free port receiving command).

10.3.2 Free mouth parameter setting

(1) Select the communication configuration in the Connect, and select the free port protocol in the corresponding parameter setting to activate the corresponding free port setting button, as shown in the following figure:



The configurable contents are as follows:

Options	Set content	Notes
Baud rate	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200, default is 9600	-
Data bits	Set 7 or 8, default is 8	-
Parity bit	Set to no parity, odd parity, even parity, the default is no parity	-
Stop bit	Set 1 or 2, default is 1	-
Allow start character detection	Allow or forbid, default is forbidden	-
Start character detection	0 to 255 (corresponding to 00 to FF)	Detect the starting character specified by the user, start receiving, and save the received character (including the starting character) to the buffer area specified by the user
Allow end character detection	Allow or forbid, default is forbidden	-
End character detection	0 to 255 (corresponding to 00 to FF)	When the end character set by the user is received, the reception is ended, and the end character is saved in the buffer area.
Timeout between characters allowed	Allow or forbid, default is forbidden	-
Inter-character timeout	0～65535ms	When the time between the received two characters exceeds the time-out time between characters set by the user, the reception is aborted
Frame Timeout Enable	Valid or invalid, default is invalid	-
Frame timeout	0～65535ms	When the power flow of the RCV is turned on and the communication conditions are met, start receiving from the communication serial port.

10.3.3 Free port Commands

- Precautions

The free port commands XMT and RCV can be used to send and receive data to the specified communication port. For the detailed usage of the free port commands, please refer to6.12.2 XMT: Free port send commandand6.12.3 RCV: Free port receive command.

It should be noted that if you use the free port command on a port, you need to set the communication port to use the free port protocol and set the communication parameters in the system settings of the VEDA PCT software. After the setting is completed, download the system settings to the PLC, and Reboot to take effect.

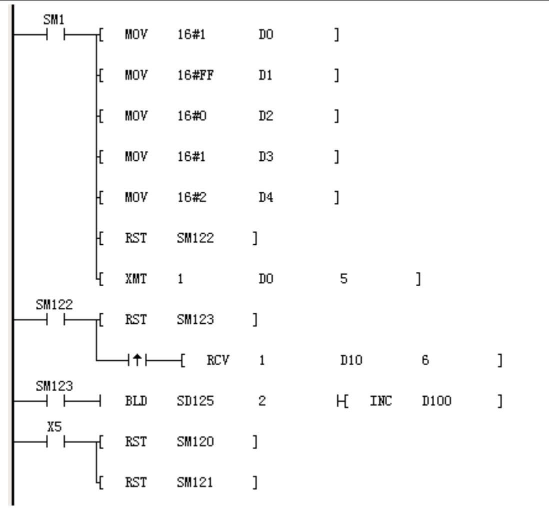
- Program example

Routine 1: Send data through COM1, and then receive data, the data sent is 5 bytes, and the received data is 6 bytes.

The data sent is: | 01 | FF | 00 | 01 | 02 | The received data is: | 01 | FF | 02 | 03 | 05 | FE |

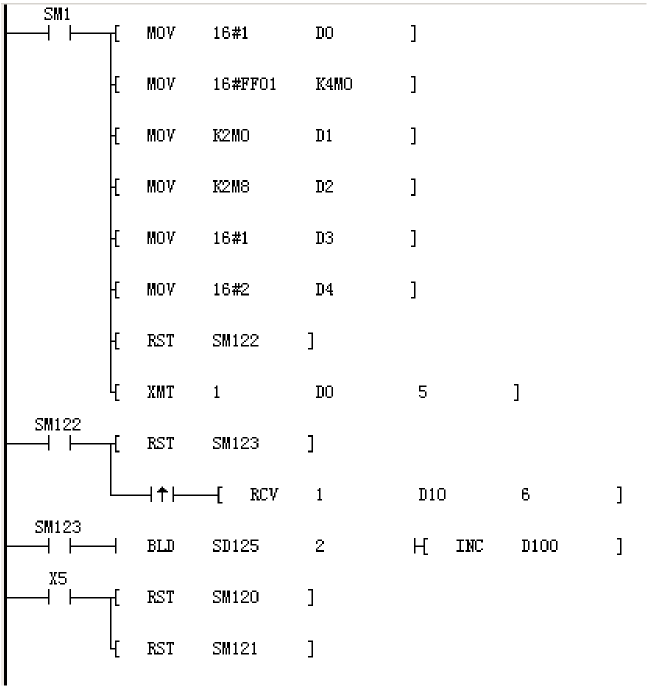
Save the received data to the address starting from D10, and save each byte to a D element. The way of saving is shown in the following table:

01	FF	02	03	05	FE
D10	D11	D12	D13	D14	D15



1. First of all, the setting of the communication port should be changed to free port communication in the system block, and the parameters such as baud rate and parity should be set.
2. When the primary power flow of SM1 is valid, save the data to be sent in the communication buffer area starting from D0, use the XMT instruction to send the data, and reset SM122 (the end of sending Sign) before sending.
3. After the transmission is completed, SM122 is set, and the rising edge is used to start receiving data. The maximum length received is 6.
4. When the reception is completed, SM123 is set, and the corresponding operation is performed according to the content of the reception completion information register (SD125).
5. Use X5 as the enable bit for interrupt transmission and reception

Routine 2: Send data through communication port 1, and then receive data.



Different from **routine 1**, when the high and low bytes of a word element are to be sent, the word element must first be split into two parts: high and low bytes. For example, to send the content of the word element D2, you can first store its high and low bytes in the low middle of D3 and D4, and then send D3 and D4. You can also use the method of first saving the data to a K4MX (such as K4M0 in this program), and then taking the high byte K2M0 as the high byte and K2M8 as the low byte.

10.4 Modbus Communication Protocol

10.4.1 Introduction

VEDA VC series small PLC serial communication can use Modbus communication protocol, support RTU communication mode, and can be set as a master station or a slave station.

10.4.2 Link characteristics

A. Physical layer: RS232, RS485

B. Link Layer: Asynchronous Transfer

(1) Data bits: 8 bits (RTU)

(2) Transmission rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

(3) Check mode: even check, odd check or no check

(4) Stop bit: 1 or 2 stop bits

D. Network configuration: up to 31 devices, with an address range of 1 to 247. Broadcast is supported.

10.4.3 RTU transmission mode

1. Hex data.
2. The inter-character spacing should be less than 1.5 character times.
3. There is no frame header and frame trailer, and the interval between frames is at least 3.5 character times.
4. Use CRC16 checksum.
5. The maximum frame length of the RTU frame is 256 bytes, and the frame structure is as follows:

Frame composition	Address	Function code	Data	CRC
Number of bytes	1	1	0～252	2

6. Character interval calculation:

The communication baud rate is 19200, then 1.5 character time=1/19200×11×1.5×1000=0.86ms

3.5 character interval = 1/19200×11×3.5×1000=2ms.

10.4.4 Modbus function code and data addressing

A. When VEDA VC series PLC is used as a slave station, it supports function codes 01, 02, 03, 04, 05, 06, 15, 16 in the Modbus communication protocol.

Function code (decimal)	Function code name	Modbus data address	Operable element type	Notes
01	Read coil	0 Note 1: xxxx	Y, X, M, SM, S, T, C	Read bit
02	Read discrete input	1 Note 2: xxxx	X	Read bit
03	Read register	4 Note 3: xxxx Note 4	D, SD, Z, T, C, R	Read characters
05	Write a single coil	0:xxxx	Y, M, SM, S, T, C	Write bit
06	Write a single register	4:xxxx	D, SD, Z, T, C, R	Write characters
15	Write multiple coils	0:xxxx	Y, M, SM, S, T, C	Write bit
16	Write multiple registers	4:xxxx	D, SD, Z, T, C, R	Write characters

Note:
1. 0 means coil
2. 1 Representing discrete input
3 .4 Representation register
4. xxxx represents the range from 1 to 9999. Each type has an independent logical address range of 1 to 9999 (protocol addresses start from 0).
5. 0, 1, 4 do not have physical meaning and do not participate in actual addressing.
6. User should not use function code 05, 15 to write to X element. If the X element is written, and the written operand and data are correct, the system will not return an error message, but the system will not perform any operation on the written command.

B. Modbus frame format (take Modbus-RTU as an example)

1. Function code: 0X01 (01) Read coil

Request frame format: slave address+0X01+coil start address+coil number+CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X01 (function code)	1 byte	read coil
3	Coil start address	2 bytes	Highs come first, lows come after
4	Number of coils	2 bytes	High order first, low order last (N)
5	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X01 + number of bytes + coil status + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X01 (function code)	1 byte	Read coil
3	Number of bytes	1 byte	Value [(N/7) /8] (N is the number of coils read)
4	Coil Status	[(N/7)/8] bytes	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 bit. And so on.
5	CRC check	2 bytes	Highs come first, lows come after

2. Function code: 0X02 (02) Read coil

Request frame format: slave address+0X02+coil start address+coil number+CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X02 (function code)	1 byte	Read coil
3	Coil start address	2 bytes	Highs come first, lows come after
4	Number of coils	2 bytes	High order first, low order last (N)
5	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X02 + number of bytes + coil status + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X02 (function code)	1 byte	Read coil
3	Number of bytes	1 byte	Value [(N/7) /8] (N is the number of coils read)
4	Coil Status	[(N/7)/8] bytes	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 bit. And so on.
5	CRC check	2 bytes	Highs come first, lows come after

3. Function code: 0X03 (03) Read register

Request frame format: slave address + 0X03 + register start address + register number + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X03 (function code)	1 byte	Read register
3	Register start address	2 bytes	Highs come first, lows come after
4	Number of registers	2 bytes	High order first, low order last (N)
5	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X03 + number of bytes + register value + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X03 (function code)	1 byte	Read register
3	Number of bytes	1 byte	Value: N*2
4	Register value	N*2 bytes	Every two bytes represent a register value, with the high-order bits in the front and the low-order bits in the back. Register address is smaller in front
5	CRC check	2 bytes	Highs come first, lows come after

4. Function code: 0X04 (04) Read register

Request frame format: slave address + 0X04 + register start address + register number + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X04 (function code)	1 byte	Read register
3	Register start address	2 bytes	Highs come first, lows come after
4	Number of registers	2 bytes	High order first, low order last (N)
5	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X04 + number of bytes + register value + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X04 (function code)	1 byte	Read register
3	Rumber of bytes	1 byte	Value: N*2
4	Register value	N*2 bytes	Every two bytes represent a register value, with the high-order bits in the front and the low-order bits in the back. Register address is smaller in front
5	CRC check	2 bytes	Highs come first, lows come after

5. Function code: 0X05 (05) Write single coil

Request frame format: slave address+0X05+coil address+coil status+CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X05 (function code)	1 byte	Write single coil
3	Coil address	2 bytes	Highs come first, lows come after
4	Coil Status	2 bytes	High-order first, low-order last The value of the effective write element is 0xFF00 (ON, 1) or 0x0000 (OFF, 0)
5	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X05 + coil address + coil status + CRC check

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X05 (function code)	1 byte	Write single coil
3	Coil address	2 bytes	Highs come first, lows come after
4	Coil Status	2 bytes	High order first, low order after FF00 is valid
5	CRC check	2 bytes	Highs come first, lows come after

6. Function code: 0X06 (06) Write a single register

Request frame format: slave address + 0X06 + register address + register value + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X06 (function code)	1 byte	write a single register
3	register address	2 bytes	Highs come first, lows come after
4	register value	2 bytes	Highs come first, lows come after
5	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X06 + register address + register value + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X06 (function code)	1 byte	Write a single register
3	Register address	2 bytes	Highs come first, lows come after
4	Register value	2 bytes	Highs come first, lows come after
5	CRC check	2 bytes	Highs come first, lows come after

7. Function code: 0X0F (15) Write multiple coils

Request frame format: slave address+0X0F(15)+coil start address+coil number+byte number+coil status+CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X0F (function code)	1 byte	Write multiple single coils
3	Coil start address	2 bytes	Highs come first, lows come after
4	Number of coils	2 bytes	High order first, low order last (N)
5	Number of bytes	1 byte	Value [(N/7) /8] (N represents the number of write coils)
6	Coil Status	[(N/7)/8] bytes	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 bit. And so on.
7	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave station address+0X0F(15)+coil start address+coil number+CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X0F (function code)	1 byte	Write multiple single coils
3	Coil start address	2 bytes	Highs come first, lows come after
4	Number of coils	2 bytes	Highs come first, lows come after
5	CRC check	2 bytes	Highs come first, lows come after

8. Function code: 0X10 (16) Write multiple registers

Request frame format: slave address+0X10(16)+register start address+register number+byte number+register value+CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X10 (function code)	1 byte	Write multiple registers
3	Register start address	2 bytes	Highs come first, lows come after
4	Number of registers	2 bytes	High order first, low order last (N)
5	Number of bytes	1 byte	Value [(N/7) /8] (N represents the number of write coils)
6	Register value	N*2 or (N*4)	
7	CRC check	2 bytes	Highs come first, lows come after

Response frame format: slave address + 0X10 + register start address + register number + CRC check;

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X010 (function code)	1 byte	Write multiple registers
3	Register start address	2 bytes	Highs come first, lows come after
4	Number of registers	2 bytes	Highs come first, lows come after
5	CRC check	2 bytes	Highs come first, lows come after

9. Error response frame

Error response: slave address + (function code + 0X80) + error code + CRC check

Serial number	Data byte meaning	Number of bytes	Illustrate
1	Slave address	1 byte	1~247 (communication setting interface)
2	0X80+ function code	1 byte	Error function code
3	Error code	1 byte	See appendix
4	CRC check	2 bytes	Highs come first, lows come after
The function code is the function code for intercepting the requested frame + 0x80			

Precautions:

1.Referring to the address division of soft elements, the type of soft element read each time is of the same type. For example, X and Y elements cannot be read back together in one frame.

2.The address and data range for reading this type of device cannot exceed the range specified in the protocol. An example is as follows:

The protocol address range of the known Y element is 0000～0255 (Y0～Y377):

①If the read start address is 1 and the number of read components is 256, an address error (exception code 02) will be returned, because there are only 255 Y components starting from 1;

②If the read start address is 0 and the number of read components is 257, a data error (exception code 03) will be returned, because the number of read components exceeds 256, and only 256 Y components are actually defined ;

③If the read start address is 0, and the number of read elements is 256, the status of 256 elements will be returned;

That is, it must be ensured that the element being read is actually defined (in scope). This is true for both read and write word elements and bit elements.

10.4.5 Modbus communication address

A. When the PLC is used as a Modbus communication slave, the corresponding relationship between the device and the Modbus address is as follows:

Element	Type	Physical element	Protocol address	Supported function codes	Notes
Y	Bit element	Y0~Y777 (octal code) a total of 512 points	0000～0511	01, 05, 15	The status of the output, the component numbers are Y0~Y7, Y10~Y17
X	Bit element	X0～X777 (octal code) a total of 512 points	1200～01711	01, 05, 15 02	The state of the input, supports two kinds of addresses, the component number is the same as above
M	Bit element	M0～M2047 M2048～M10239	2000～4047 12000-20191	01, 05, 15	
SM	Bit element	SM0～SM255 SM256～SM1023	4400～4655 30000-30767	01, 05, 15	
S	Bit element	S0～S1023 S1024～S4095	6000-7023 31000-34071	01, 05, 15	
T	Bit element	T0～T255 T256～T511	8000～8255 11000-11255	01, 05, 15	The state of the T element
C	Bit element	C0～C255 C256～C511	9200～9455 10000-10255	01, 05, 15	The state of the C element
D	Word element	D0～D7999	0000～7999	03, 06, 16	
SD	Word element	SD0～SD255 SD256～SD1023	8000～8255 12000-12767	03, 06, 16	
Z	Word element	Z0～Z15	8500～8515	03, 06, 16	
T	Word	T0～T255	9000～9255	03, 06, 16	Current value of T element

Element	Type	Physical element	Protocol address	Supported function codes	Notes
	element	T256～T511	11000-11255		
C	Word element	C0～C199	9500～9699	03, 06, 16	Current value of C element (INT)
C	Double word element	C200～C255	9700～9811	03, 16	Current value of C element (DINT)
C	Double word element	C256～C263	10000-10101	03, 16	Current value of C element (DINT)
R	Word element	R0～R32767	13000-45767	03, 06, 16	

10.4.6 Read and write components

In addition to function code 08, other supported function codes are all read and write operations for components. In principle, a maximum of 2000 bit components can be read in one frame, 1968 bit components can be written, 125 word components can be read, and 120 word components can be written. However, since the actual protocol addresses are separate and discontinuous for different types of components (for example, the protocol address of Y377 is 255, and the protocol address of X0 is 1200), when reading and writing components, the components read at one time can only be read and written. It is a type of component, and the maximum number of read components is also related to the actual number of components of this type. For example, reading Y components, Y0~Y377 (256 points in total), the protocol address range is 0~255, corresponding to the logical address of the Modicon data is 1 to 256, and it is allowed to read up to 256 elements when reading the Y element.

An example is as follows:

1. Master send: 01 01 00 00 01 00 3D 9A

   01 address, function code 01, 00 00 start address, 01 00 read the number of components 3D 9A check

   Slave answer: will return the correct answer

2. Master send: 01 01 00 00 01 01 FC 5A

   The master station reads 01 01 (257) elements from the starting address of 0000, which exceeds the defined number of Y elements

   Slave reply: 01 81 03 00 51

   The slave reply is an illegal data value because 257 is greater than 256, and 256 is the maximum allowed number of Y elements

3. Master send: 01 01 00 64 00 A0 7D AD

   The master station reads the starting address 00 64 (decimal 100), the number of components 00 A0 (decimal 160)

   Slave reply: 01 81 02 C1 91

   The slave station responds to the illegal data address. There are only 156 Y elements starting from the protocol address 100, and reading 160 is illegal.

4. Master send: 01 04 00 02 00 0A D1 CD

   The master station sends the frame of function code 04

   Slave reply: 01 84 01 82 C0

   The slave station responds to an illegal function code, VC2L does not support function code 04

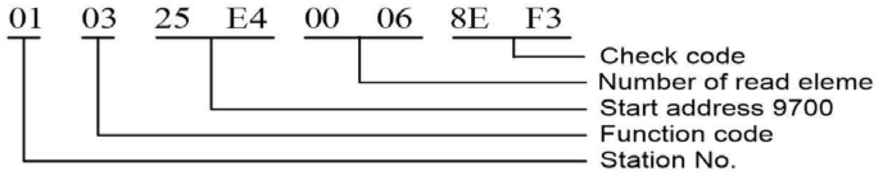
📖 Notice

1. The X element does not support writing (that is, writing to the X element is an invalid operation). SM, SD component writable properties please refer to Chapter 13

10.4.7 Handling of double word components

The current count value of C element is word element or double word element, C200～C255 are double word elements. The read and write of C200～C255 is also completed by the function codes (03, 16) of the read and write registers. The address of each two registers corresponds to a C double word element, and only pairs of registers can be read and written when reading and writing.

For example: to read the RTU frame of three C double word elements from C200 to C202:



In the returned data, the two addresses 9700 and 9701 represent the content of C200, 9700 is the upper 16 bits, and 9701 is the lower 16 bits.

When reading a double-word element, if the read start address is not an even number, an illegal address with an exception code will be returned. If the number of registers read is not an even number, an illegal data with an exception code will be returned.

An example is as follows:

Master send: 01 03 25 E5 00 04 5E F2

The master sends a four-word element whose read start address is 25 E5 (9701 in decimal)

Slave response: 01 83 02 C0 F1

Slave reply: Illegal data address

Master send: 01 03 25 E4 00 05 CE F2

The master station reads 5 word elements whose start address is 25 E4

Slave reply: 01 83 03 01 31

Slave returned illegal data

10.4.8 Handling of dint

For the storage of a DINT type data, there may be two D elements, for example: D3, D4 store a DINT type number, VEDA VC series PLC thinks that D3 stores the upper 16 bits, D4 stores the lower 16 bits, when the main When the station reads DINT data through Modbus, after reading back the data, it should also reorganize the 32-bit data according to the storage principle of VEDA VC series PLC for DINT. The storage principle of FLOAT is equivalent to the storage principle of DINT.

10.4.9 Diagnostic function code

The diagnostic function code is used to test the communication between the master station and the slave station, or various internal error states of the slave station. The supported diagnostic sub-function codes are shown in the following table:

Function code	Sub function code	Sub-function code name		Function code	Sub function code	Sub-function code name
08	00	Return query data		08	12	Returns the bus communication error count
08	01	Restart communication options		08	13	Returns the bus exception error count
08	04	Force listen-only mode		08	14	Returns the slave message count
08	10	Clear counter		08	15	Returns the slave no response count
08	11	Returns the bus message count		08	18	Returns the bus character overrun count

10.4.10 Exception code

When the master sends a command, in a normal response, the slave returns data or statistics in the data field. In the abnormal response, the server returns the abnormal code in the data field. The abnormal code is as follows:

Exception code	Exception code meaning
0x01	Illegal function code
0x02	Illegal register address
0x03	Illegal data

In addition, the slave station will not return a response message when it receives data in the following situations:

(1) There are errors in the broadcast frame, such as data errors, address errors, etc.

(2) The character limit is not returned, for example, the RTU frame is larger than 256 bytes.

(3) In RTU transmission mode, the interval time between characters is overtime, which is equivalent to receiving an error frame and does not return.

(4) The slave does not return in listen-only mode.

(5) Slave received bad ASCII error frame, including end of frame error, wrong character range in frame.

📖 Notice

The reading station has a forced element, and what is read is only the value of the program running, which may not match the forced value

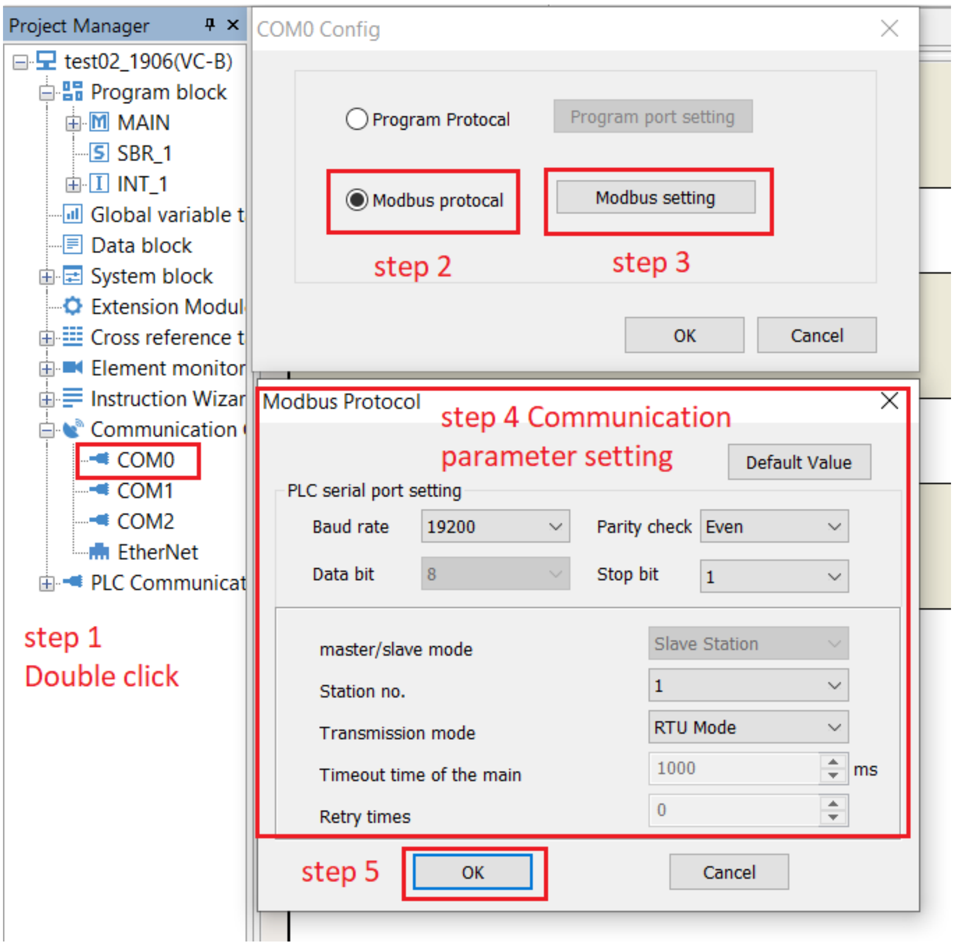
10.4.11 Modbus slave communication settings

(1) In industrial applications, PLC, as the industrial automation control layer, needs to be monitored by the automation control network. When the PLC communication port needs to set the Modbus slave mode to communicate with the host computer. VEDA VC series PLC has built-in Modbus-RTU slave protocol, and the slave protocol can be run on COM0, COM1 and COM2 ports.

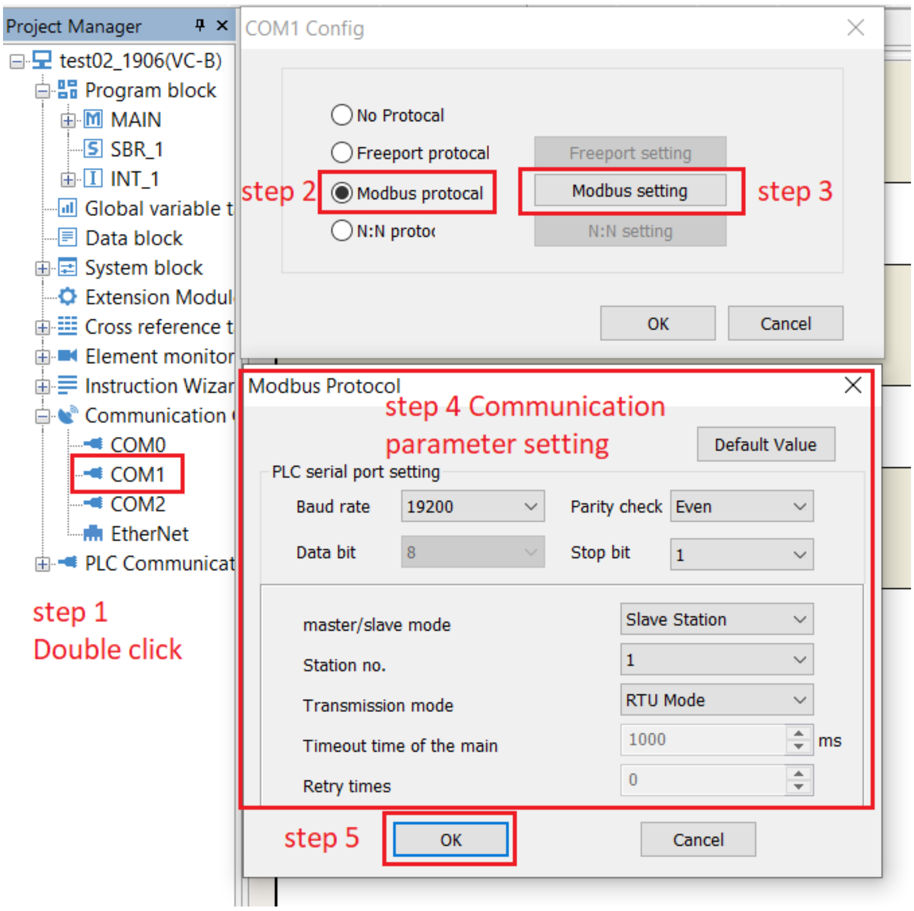
(2) When the PLC acts as a Modbus slave station, it does not actively send any message, and only after receiving the message for local addressing will it check whether it responds to the master station according to the specific situation. Slave only supports Modbus Function codes 01, 02, 03, 05, 06, 08, 15, 16, and the rest of the responses are "illegal function codes" (except broadcast frames).

A. Software setting slave station

(1) COM0 is shown in the figure below



(2) COM1/CM02 Slave settings;



10.4.12 Modbus master communication settings

- Set the communication port of【Communication Configuration】

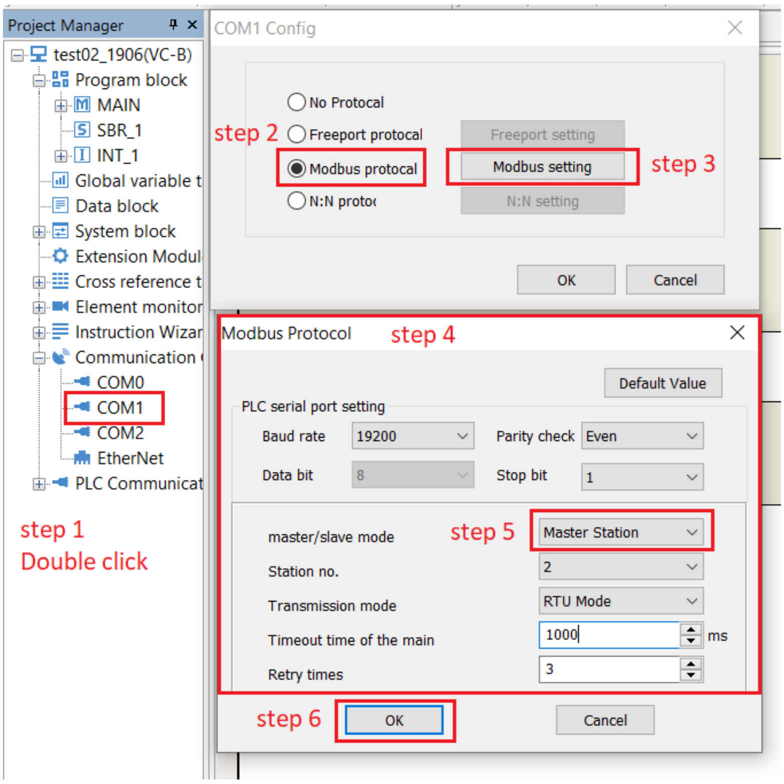
  There are three serial port options in the communication interface interface, COM0, COM1 and COM2, among which COM0 only supports Modbus slave station, and COM1 and COM2 support Modbus master station or slave station.

- Set Modbus communication protocol parameters

  In the Modbus communication protocol operand interface, there is a default value button, and the default value is the communication setting recommended by the Modbus communication protocol. The parameter setting options are shown in the table below.

Options	Set content
Station No	0～247
Baud rate	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200
Data bits	Set 7 or 8, 7 bits in ASCII mode, 8 bits in RTU mode
parity bit	Set to no parity, odd parity, even parity
stop bit	Set 1 or 2, set to 1 for odd and even parity, set to 2 for no parity
Modbus Master/Slave	Can be set as master station or slave station, communication port 1 can be set as master station or slave station, communication port 0 cannot be set as master station
transfer mode	Select RTU mode or ASCII mode
main mode timeout	Timeout for the master to wait for the slave to respond
Note: After the operand is set in the system block and downloaded, it is not effective immediately, and it must be run once to be effective.	

- The software sets the master station COM1/COM2 to set the master station as shown in the figure below



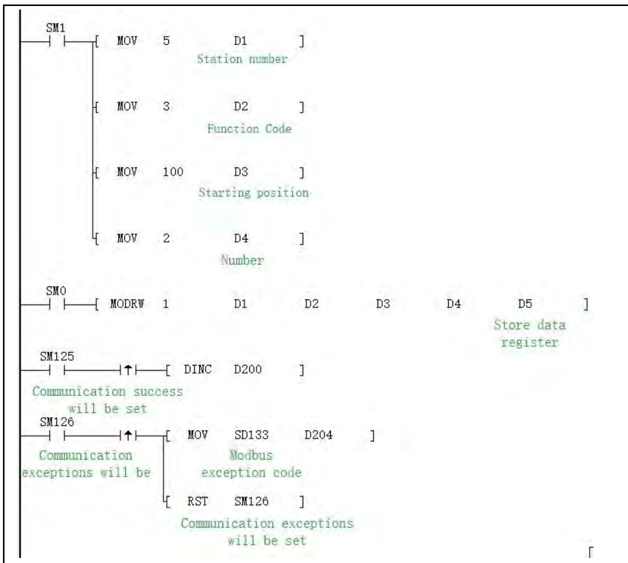
10.4.13 Instructions for the use of MODRW instructions

(1) When the PLC is used as a Modbus master station, it can send Modbus data frames and receive replies through the MODRW command, Modbus command and Modbus table configuration provided by the system. (For the detailed usage of MODRW command and Modbus command, please refer to6.12.1 Modbus: Master communication commandand MODRW instructions)

(2) When setting the PLC as the master station, when setting the Modbus parameters in [Communication Configuration], there is a timeout time for the master mode. In order to ensure the correctness of the received data, it should be ensured that this time should be longer than that of the Modbus slave station. The scanning period of one cycle of VC-B should be long and there is a margin. For example, VC-B is a slave station. If a scanning period of VC-B is 300ms, the master mode timeout time of the master station should be more than 300ms, and it is more suitable to set 350ms.

A: Demonstration of MODRW instruction

Routine 1: VC-B PLC is the Modbus master station, the slave station is also a VC-B PLC and the slave station number is set to No. 5, the master station reads the D register value of the slave station protocol address of 100~101 (decimal), and saves it to the two starting from D5 in the register. Program the Modbus master as follows:



⊠Program Description:

1. The program specifies to use the COM1 channel as the communication interface.
2. The slave address to be accessed is specified in the program as 5 (stored in D0).
3. The function code specified in the program is 03 (saved to D2).
4. The starting address of the register specified by the program to read is 100 (stored in D3).
5. The program specifies that the number of registers to be read is 2 (stored in D4).
6. The program specifies that the received data is stored in registers D5/D6.
7. If the communication is normal, SM125 is set to ON, and the D200 register value is incremented by 1.
8. If communication fails, SM126 is turned ON, and the error code is stored in D204.

According to the exception code provided by Modbus, troubleshoot the problem.

(Note: The slave only needs to configure the correct communication format, no programming is required)

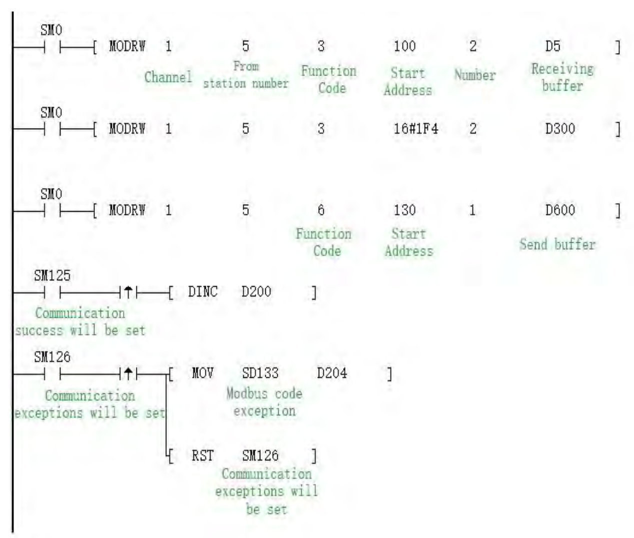
Routine 2: Use multiple MODRW instructions

VC PLC (1#) is the Modbus master station, the slave station is also VC PLC (2#) and the slave station number is set to 5, the master station performs the following operations on the slave station

(1) The master station reads the D register value of the slave station protocol address of 100~101 (decimal), and stores it in the 2 registers starting from D5;

(2) The master station reads the D register value of the slave station protocol address of 1F4~1F5 (hexadecimal), and stores it in the 2 registers starting from D300;

(3) The master station writes the value D600=100 to the D130 register whose slave station address is 130 (decimal); the programming of the Modbus master station is as follows:



⊠Program Description:

1. The program specifies to use the COM1 channel as the communication interface.
2. The slave address to be accessed is specified as 5 in the program.
3. The function code specified in the program is 03.
4. The program specifies that the starting address of the register to be read is 100.
5. The program specifies that the number of registers to be read is 2.
6. The program specifies that the received data is stored in registers D5/D6.
7. The starting address of the register specified by the program to read is 16#1F4.
8. The program specifies that the number of registers to be read is 2.
9. The received data specified by the program is stored in registers D300/D301.
10. The program specifies to write 100 to the slave register starting address 130.
11. The program specifies that the number of registers to be read is 1.
12. The data to be sent by the program is stored in the register D600.
13. If the communication is normal, SM125 is set to ON, and the D200 register value is incremented by 1.

	14. If communication fails, SM126 is turned ON, and the error code is stored in D204. According to the exception code provided by Modbus, troubleshoot the problem. (Note: The slave only needs to configure the correct communication format, no programming is required)

📖 Notice

1. When using the logical address to address the bit components of VC-B PLC, logical address 1 is protocol address 0. Also in the above example, to read the bit component values from 11 to 39 (protocol address) of the slave station, the logical address should start from 12.

2. When an error occurs in this communication, it does not affect the next communication. That is to say, there are two Modbus commands in a user program to send data. When the first communication fails and there is an error code, this does not affect the second communication. One Modbus command sends data, the second can continue. So in this example SD133The error code in is put into D204, can pass D204View error codes.

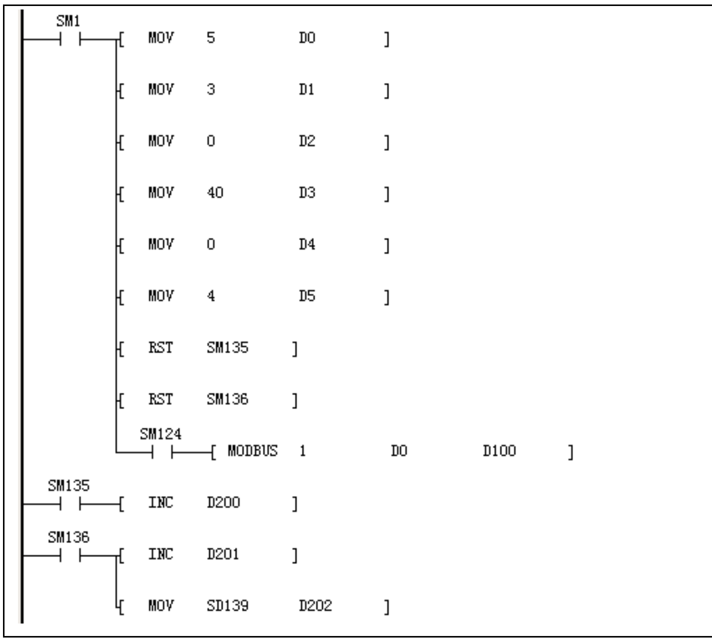
3. When the master station is in the listen-only mode, the slave station sends no data, so the error Sign will be set, so when using VC-BWhen forming a Modbus network, VC-BAs the master station, the user should clearly know which PLC slave station is in the listen-only mode to ensure that the communication error is not because the slave station is in the listen-only mode.

Example 3: Communication using Modbus commands

VC-B is the Modbus master station, and the slave station is also VC-B, which reads the word element value of the protocol address of station 5 from 40 to 43.

The read data is as follows, the received frame starts from D100, D100 saves the address, D101 saves the function code, D102 saves the number of registers, and D103 starts to save the read register value.

40 elements upper 8 bits	40 elements lower 8 bits	41 elements upper 8 bits	41 elements lower 8 bits	42 elements upper 8 bits	42 elements lower 8 bits	43 elements upper 8 bits	43 elements lower 8 bits
D103	D104	D105	D106	D107	D108	D109	D110



1. The address of the slave station to be accessed is specified in the program as 5 (stored in D0).
2. The function code specified in the program is 3 (saved to D1).
3. The starting address of the register to be read is 40 (the high and low bytes are stored in D2/D3).
4. The number of registers to read is 4.
5. The received data is saved to D100.
6. If the reception is completed (SM135 is set), D200 counts up by one.
7. If the communication fails (SM136 is set), the count of D201 is incremented by one, and the error code is stored in D202.
8. SM124 is the idle sign of the communication port.

10.4.14 Modbus table configuration instructions

The Modbus command method is flexible in programming, and the user program is easy to understand. However, in the case of a slave station communication drop, it will affect the PLC program scan time, resulting in poor control effects, and may even cause program scan timeout warnings. method, which makes this shortcoming ameliorated.

Store the communication content and data in the user program in the configuration unit, define it as a table in advance in the form of a Modbus configuration table, and download the "Modbus configuration" to the PLC when downloading the user program. When the PLC executes the user program , the system software automatically performs the communication operation of the Modbus master station. What needs to be done when programming the user program is:

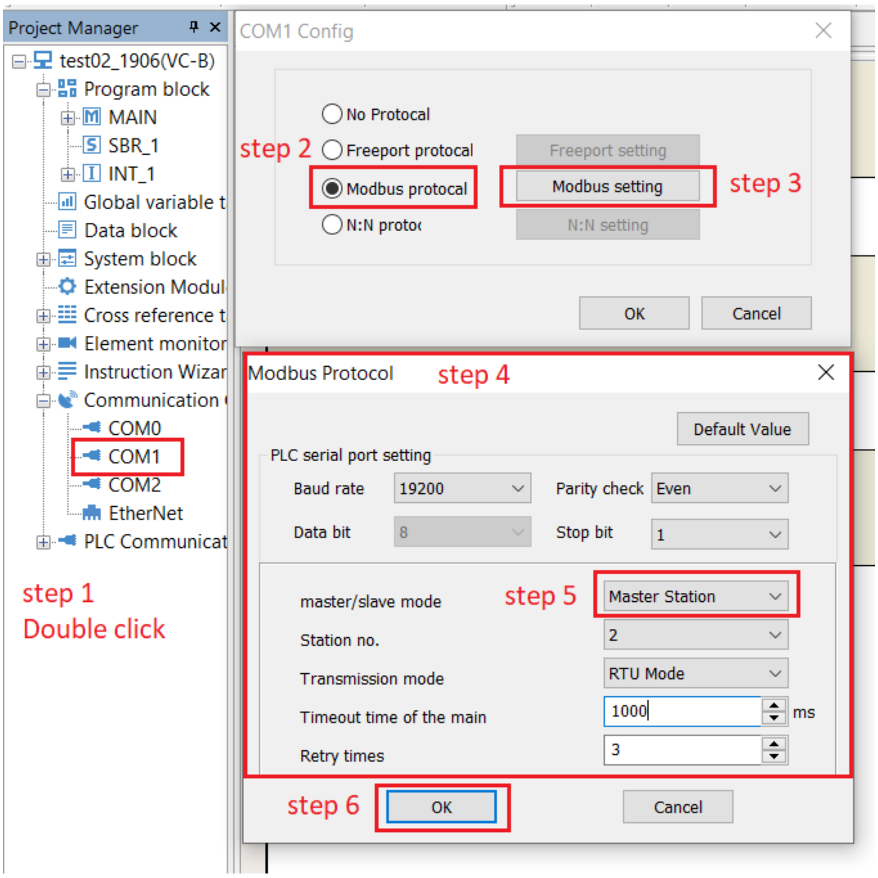
1. Configure the specified communication port as a Modbus master station, and set the communication data format;

2. Fill in the configuration table according to the data frequency characteristics, data storage address, communication trigger conditions, etc. required for communication interaction; in the user program, refresh and send the data of the D unit, trigger the M Sign, and use the received data of the D unit for control calculate;

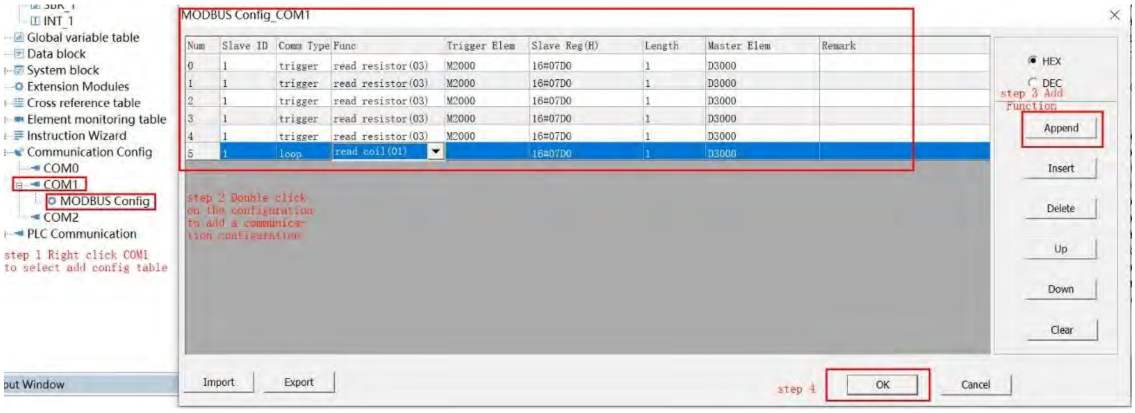
3. The master station PLC regularly checks the communication status of each Modbus slave station, judges the influence degree of the system corresponding to the communication failure of the slave station, and makes a warning or shutdown.

Protocol settings for the Modbus configuration table

1: Set the communication format of the master station as shown in the figure below



2: Modbus configuration table settings



As shown in the MODBUS configuration window above, in this window, you can add communication configuration items by clicking the "Add" button;

The information of each column in the configuration window can be edited and set. It can be seen from the configuration table that the information filled in the column is the operand required by the Modbus ladder diagram instruction. According to the desired communication operation, the D of sending and receiving data is Fill in the variable definition. After filling in, click "Confirm". The configuration is saved in the project of the user program. After the compilation is correct, download the user program to complete the operation.

Notes and suggestions for filling out the Modbus configuration table

1: When selecting 【Hexadecimal】, only the slave station 【register address】 is expressed in hexadecimal.

If reading the address of the 18th register of slave station No. 2, fill in 16#12 in the slave station register column of the form;

2: The communication method is divided into two types: 【cycle】 and 【trigger】. It is recommended to classify the required communication interaction data according to the frequency of need;

(1) Cyclic communication

It is necessary to repeatedly read and write the data that changes rapidly in the slave station as soon as possible, such as reading the running frequency of the inverter, running status, input port status, etc., and you can choose the 【cycle】 communication mode. When the PLC executes the user program, it will repeatedly scan and execute all the "loop" configuration items in the communication configuration table;

(2) Trigger communication

It is necessary to regularly read or write the data with slow refresh speed of the slave station, such as reading the output current, output power, current fault information of the inverter, etc., you can select the communication mode of [Trigger]. In the user program, each setting triggers If the Flag bit is set once, it will trigger the communication operation of the corresponding communication item in the communication configuration table once, and the Flag bit is set regularly in the user program to realize the required frequency of communication read and write operations.

3: Suggestions for setting the communication method

Reasonable configuration according to the characteristics of the interactive parameter refresh required can greatly improve the communication performance. Do not set all communication items to 【cycle】 communication for the sake of simplicity in programming. The interaction timeliness is reduced, which affects the control effect of the system. Setting some unimportant data access as 【trigger】 communication can greatly improve the real-time communication.

Based on RS485, the common Modbus communication rate is 9600bps. According to experience, the 【cycle】 communication item is limited to less than 10, and there are about 10 trigger items per second, and the communication timeliness is good.

4: Suggestions on setting trigger variable M

When the communication mode selected "trigger" mode, when the trigger conditions of the bit components set to ON, the communication operation is triggered, when the PLC will trigger the communication success, the system will automatically clear the trigger flag bit, so the M flag can also be used as a successful communication judgment flag. Therefore, when setting the communication configuration table, do not use an M variable as the trigger flag bit to trigger multiple communication configurations, so as to avoid the system clearing the M flag bit operation, which affects the communication operation of other items.

5: Modbus communication operation type

In the "Function" column of the configuration table, you can select an operation type for each configuration item, namely read register, write register, read coil, write coil, where "register" is expressed as a word variable (16bit or INT type) variable), while "coil" is represented as a bit variable (1bit variable has only 0 or 1). The commands for these two different types of communication operations are different, and you should select them according to the type of variables to be accessed when filling in. (Note: To access the variables inside the slave, you need to understand the rules for defining the slave register address)

10.5 N: N Communication Protocol

10.5.1 Introduction to N: N

N: N is a small PLC network developed by VEDA MC N: N uses RS485 at the physical layer, and PLC can be connected directly through communication port 1 or through RS232/RS485 converter through communication port 0. PLCs connected to N: N can automatically exchange the values of some D elements and M elements with each other, which makes accessing other PLC elements in the network as simple and convenient as accessing their own elements. In N: N Data access between PLCs is completely equal (N: N communication network).

N: N Convenient configuration, most parameters only need to configure No. 0 PLC. Support online modification of network parameters. Can automatically detect new PLCs that join the network. When any one PLC is disconnected from the network, other PLCs will continue to exchange data. Through the relevant SM components of any PLC in N: N, the communication situation of the entire network can be monitored.

10.5.2 The transmission form of N: N network data

There are two kinds of messages in N: N: the token issued by the main station; the broadcast of each PLC's own data.

The token is issued uniformly by the master station. The master station first holds the token, and after broadcasting the data, the token is circulated and issued to each slave station in turn. Only the slave station that receives the token can broadcast to other PLCs (including the master station).

Figure 10-1 to Figure 10-5 show the main process of network communication. The 1# station in the figure is the master station. It should be pointed out that, under normal circumstances, the default 0# is the master station, and 1# is the standby master station (when the master station has a communication failure or power failure, it will switch to the master station).

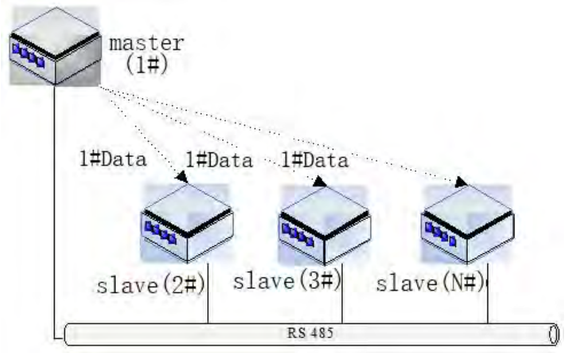


Figure 10-1 Master station broadcast

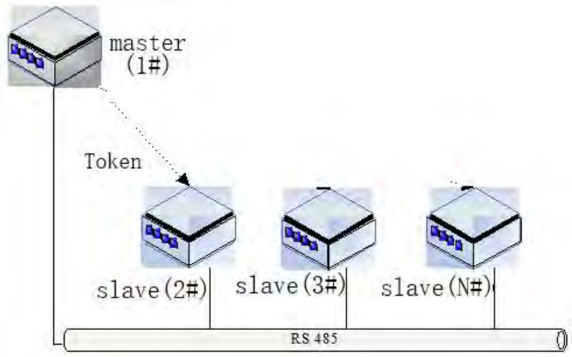


Figure 10-2 The master issues a token to the 2# slave

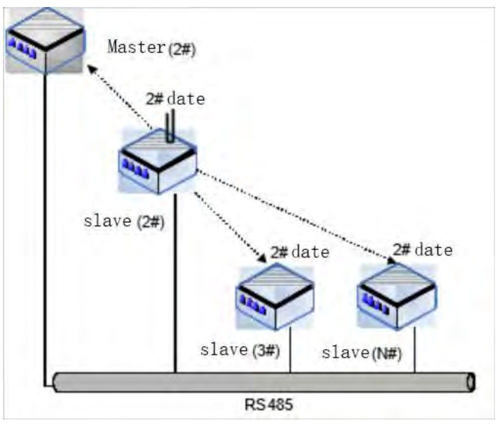


Figure 10-3 Slave Broadcast

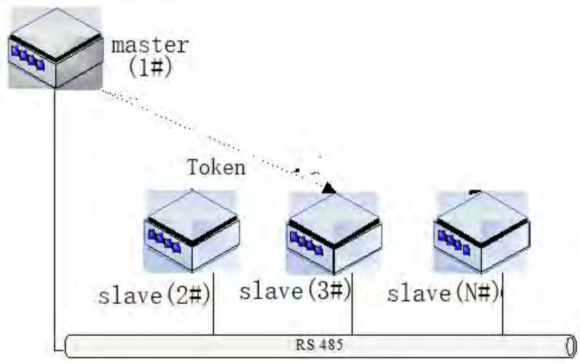


Figure 10-4 The master issues a token to the 3# slave

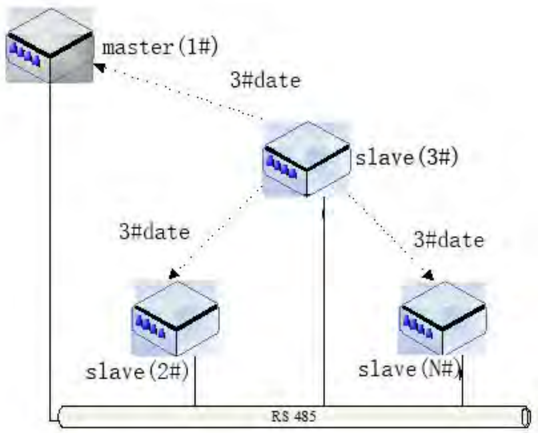


Figure 10-5 Slave Broadcast

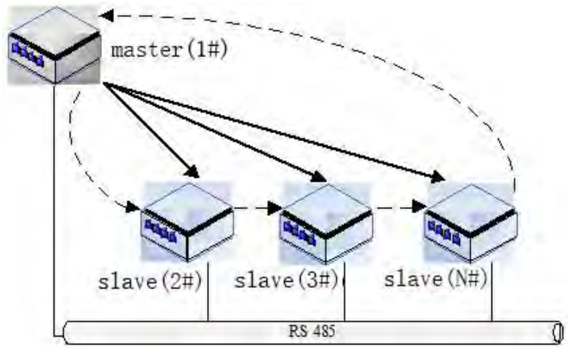


Figure 10-6 Token issuance and flow sequence

Figure 10-6 shows the order in which the tokens flow. The thick solid line shows the actual token issuance process, and the dashed line shows the sequence of stations that hold the token and broadcast it. It should be noted that the token is not passed from one slave station (such as 2# PLC) to another slave station (such as 3# PLC), but firstly by the master station to issue the token to 2# PLC, and then by the master station issues it to 3#PLC.

10.5.3 N: N network architecture

N: N can be connected into two types of networks: single-layer network and multi-layer network. As shown below:

Figure 10-1 N: N single layer network

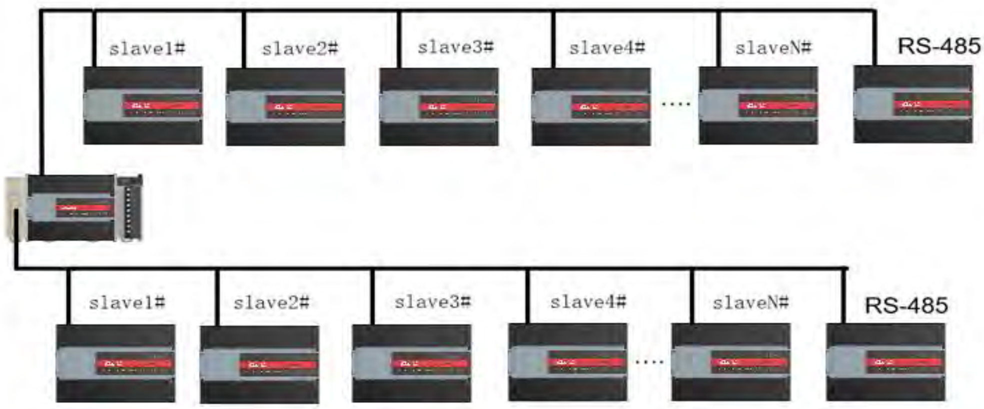
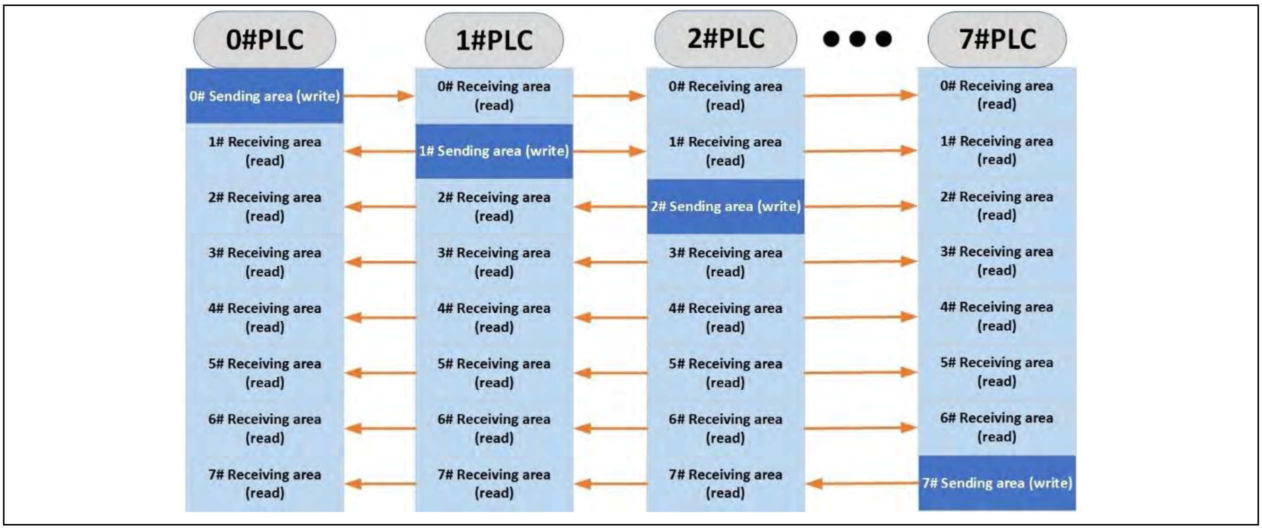


Figure 10-2 N: N multi-layer network

In a single-layer network, each PLC is connected to N: N through only one PORT port. In the multi-layer network, it is necessary to connect the intermediate node PLC of the layer and the layer, and the two communication ports of the intermediate node PLC are respectively connected to different layers. A single-layer network can support up to 32 PLCs, and each layer of a multi-layer network can support up to 16 PLCs.

10.5.4 N: N Refresh mode

Multiple PLCs connected to N: N can automatically exchange some D elements and M elements in the network. The number and number of these D elements and M elements are fixed, and these elements are called "shared element areas". Once the PLC uses N: N, the value of the shared component area will be automatically refreshed continuously, so that the value of the shared component area of each PLC in the network remains equal.



As shown in the figure above, each PLC connected to N: N has its own writable sending area in this shared component area, and N: N will automatically transfer the contents of this writable sending area (The values of D elements and M elements of specific numbers) are broadcast to other PLCs, and at the same time, other PLCs are also received to broadcast their contents to themselves, and store them in the corresponding read-only sending area.

Since the number of components in the shared component area is fixed (a total of 64 D components and 512 M components can be shared), these components are allocated to multiple PLCs. Therefore, the less the number of PLCs connected to the network, the more components are allocated to each PLC. This correspondence is defined by the N: N refresh mode table:

- N: N Single-layer network D element assignment:

<table>
<tr><th>Send Area D Component Assignment</th><th>Mode 1</th><th>Mode 2</th><th>Mode 3</th><th>Mode 4</th><th>Mode 5</th></tr>
<tr><td>D7700～D7701</td><td>#0</td><td rowspan="2">#0</td><td rowspan="4">#0</td><td rowspan="8">#0</td><td rowspan="16">#0</td></tr>
<tr><td>D7702～D7703</td><td>#1</td></tr>
<tr><td>D7704～D7705</td><td>#2</td><td rowspan="2">#1</td></tr>
<tr><td>D7706～D7707</td><td>#3</td></tr>
<tr><td>D7708～D7709</td><td>#4</td><td rowspan="2">#2</td><td rowspan="4">#1</td></tr>
<tr><td>D7710～D7711</td><td>#5</td></tr>
<tr><td>D7712～D7713</td><td>#6</td><td rowspan="2">#3</td></tr>
<tr><td>D7714～D7715</td><td>#7</td></tr>
<tr><td>D7716～D7717</td><td>#8</td><td rowspan="2">#4</td><td rowspan="4">#2</td><td rowspan="8">#1</td></tr>
<tr><td>D7718～D7719</td><td>#9</td></tr>
<tr><td>D7720～D7721</td><td>#10</td><td rowspan="2">#5</td></tr>
<tr><td>D7722～D7723</td><td>#11</td></tr>
<tr><td>D7724～D7725</td><td>#12</td><td rowspan="2">#6</td><td rowspan="4">#3</td></tr>
<tr><td>D7726～D7727</td><td>#13</td></tr>
<tr><td>D7728～D7729</td><td>#14</td><td rowspan="2">#7</td></tr>
<tr><td>D7730～D7731</td><td>#15</td></tr>
<tr><td>D7732～D7733</td><td>#16</td><td rowspan="2">#8</td><td rowspan="3">#4</td><td rowspan="3">#2</td><td rowspan="3">#1</td></tr>
<tr><td>D7734～D7735</td><td>#17</td></tr>
<tr><td>D7736～D7737</td><td>#18</td><td>#9</td></tr>
</table>

Send Area D Component Assignment	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
D7738～D7739	#19				
D7740～D7741	#20	#10	#5		
D7742～D7743	#21				
D7744～D7745	#22	#11			
D7746～D7747	#23				
D7748～D7749	#24	#12	#6	#3	
D7750～D7751	#25				
D7752～D7753	#26	#13			
D7754～D7755	#27				
D7756～D7757	#28	#14	#7		
D7758～D7759	#29				
D7760～D7761	#30	#15			
D7762～D7763	#31				

For example:

（1）In mode 1, the D components of the sending area allocated by station 0# are D7700~D7701. The PLC of station 0# can write values to D7700 and D7701, and other stations (1#-~31#) can directly read the values of D7700 and D7701. value.

（2）In mode 2, the D components of the sending area allocated to the 0# station are D7700~D7703. The 0# station PLC can write values to D7700, D7701, D7702, D7703, and other stations (1#-~15#) can directly read Values of D7700 to D7704.

- N: N Single-layer network M element assignment:

Send area M component assignment	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
M1400～M1415	#0	#0	#0	#0	#0
M1416～M1431	#1				
M1432～M1447	#2	#1			
M1448～M1463	#3				
M1464～M1479	#4	#2	#1		
M1480～M1495	#5				
M1496～M1511	#6	#3			
M1512～M1527	#7				
M1528～M1543	#8	#4	#2	#1	
M1544～M1559	#9				
M1560～M1575	#10	#5			
M1576～M1591	#11				
M1592～M1607	#12	#6	#3		#0
M1608～M1623	#13				
M1624～M1639	#14	#7			
M1640～M1655	#15				
M1656～M1671	#16	#8	#4	#2	#1
M1672～M1687	#17				
M1688～M1703	#18	#9			
M1704～M1719	#19				
M1720～M1735	#20	#10	#5		
M1736～M1751	#21				
M1752～M1767	#22	#11			
M1768～M1783	#23				

Send area M component assignment	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
M1784～M1799	#24	#12	#6	#3	
M1800～M1815	#25				
M1816～M1831	#26	#13			
M1832～M1847	#27				
M1848～M1863	#28	#14	#7		
M1864～M1879	#29				
M1880～M1895	#30	#15			
M1896～M1911	#31				

For example:

（3）In mode 1, the M components in the sending area allocated to station 0# are M1400~M1415. The PLC of station 0# can write values to M1400~M1415, and other stations (1#-~31#) can directly read the values of M1400~M1415. value.

（4）In mode 2, the M components in the sending area allocated to station 0# are M1400~M1431. The PLC of station 0# can write values to M1400~M1431, and other stations (1#-~31#) can directly read the values of M1400~M1431. value.

- N: N Multilayer network D element assignment (layer 0):

Send Area D Component Assignment	Mode 6	Mode 7	Mode 8	Mode 9
D7700～D7701	#0	#0	#0	#0
D7702～D7703	#1			
D7704～D7705	#2	#1		
D7706～D7707	#3			
D7708～D7709	#4	#2	#1	
D7710～D7711	#5			
D7712～D7713	#6	#3		
D7714～D7715	#7			
D7716～D7717	#8	#4	#2	#1
D7718～D7719	#9			
D7720～D7721	#10	#5		
D7722～D7723	#11			
D7724～D7725	#12	#6	#3	
D7726～D7727	#13			
D7728～D7729	#14	#7		
D7730～D7731	#15			

For example:

（5）In mode 6, D7700~D7701 are assigned to the D components in the sending area of station 0# (layer 0). The PLC of station 0# can write values to D7700~D7701, and other stations (1#-~15#) can read directly. Take the values from D7700 to D7701.

- N: N Multilayer network D element assignment (layer 1):

Send Area D Component Assignment	Mode 10	Mode 11	Mode 12	Mode 13
D7732～D7733	#0	#0	#0	#0
D7734～D7735	#1			
D7736～D7737	#2	#1		

<table>
<tr><th>Send Area D Component Assignment</th><th>Mode 10</th><th>Mode 11</th><th>Mode 12</th><th>Mode 13</th></tr>
<tr><td>D7738～D7739</td><td>#3</td><td></td><td></td><td rowspan="5"></td></tr>
<tr><td>D7740～D7741</td><td>#4</td><td rowspan="2">#2</td><td rowspan="4">#1</td></tr>
<tr><td>D7742～D7743</td><td>#5</td></tr>
<tr><td>D7744～D7745</td><td>#6</td><td rowspan="2">#3</td></tr>
<tr><td>D7746～D7747</td><td>#7</td></tr>
<tr><td>D7748～D7749</td><td>#8</td><td rowspan="2">#4</td><td rowspan="4">#2</td><td rowspan="8">#1</td></tr>
<tr><td>D7750～D7751</td><td>#9</td></tr>
<tr><td>D7752～D7753</td><td>#10</td><td rowspan="2">#5</td></tr>
<tr><td>D7754～D7755</td><td>#11</td></tr>
<tr><td>D7756～D7757</td><td>#12</td><td rowspan="2">#6</td><td rowspan="4">#3</td></tr>
<tr><td>D7758～D7759</td><td>#13</td></tr>
<tr><td>D7760～D7761</td><td>#14</td><td rowspan="2">#7</td></tr>
<tr><td>D7762～D7763</td><td>#15</td></tr>
</table>

For example:

（6）In mode 10, D7732~D7733 are assigned to the sending area D components of station 0# (layer 1). The PLC of station 0# can write values to D773~D7733, and other stations (1#-~15#) can read directly. Take the values from D7732 to D7733.

- N: N Multilayer network M element assignment (layer 0):

<table>
<tr><th>Send area M component assignment</th><th>Mode 6</th><th>Mode 7</th><th>Mode 8</th><th>Mode 9</th></tr>
<tr><td>M1400～M1415</td><td>#0</td><td rowspan="2">#0</td><td rowspan="4">#0</td><td rowspan="8">#0</td></tr>
<tr><td>M1416～M1431</td><td>#1</td></tr>
<tr><td>M1432～M1447</td><td>#2</td><td rowspan="2">#1</td></tr>
<tr><td>M1448～M1463</td><td>#3</td></tr>
<tr><td>M1464～M1479</td><td>#4</td><td rowspan="2">#2</td><td rowspan="4">#1</td></tr>
<tr><td>M1480～M1495</td><td>#5</td></tr>
<tr><td>M1496～M1511</td><td>#6</td><td rowspan="2">#3</td></tr>
<tr><td>M1512～M1527</td><td>#7</td></tr>
<tr><td>M1528～M1543</td><td>#8</td><td rowspan="2">#4</td><td rowspan="4">#2</td><td rowspan="8">#1</td></tr>
<tr><td>M1544～M1559</td><td>#9</td></tr>
<tr><td>M1560～M1575</td><td>#10</td><td rowspan="2">#5</td></tr>
<tr><td>M1576～M1591</td><td>#11</td></tr>
<tr><td>M1592～M1607</td><td>#12</td><td rowspan="2">#6</td><td rowspan="4">#3</td></tr>
<tr><td>M1608～M1623</td><td>#13</td></tr>
<tr><td>M1624～M1639</td><td>#14</td><td rowspan="2">#7</td></tr>
<tr><td>M1640～M1655</td><td>#15</td></tr>
</table>

For example:

（7）In mode 6, the M components in the sending area allocated to station 0# (layer 0) are M1400~M1415. PLC of station 0# can write values to M1400~M1415, and other stations (1#-~15#) can read directly Take the values from M1400 to M1415.

- N: N Multilayer network M element assignment (layer 1):

<table>
<tr><th>Send area M component assignment</th><th>Mode 10</th><th>Mode 11</th><th>Mode 12</th><th>Mode 13</th></tr>
<tr><td>M1656～M1671</td><td>#0</td><td rowspan="2">#0</td><td rowspan="2">#0</td><td rowspan="2">#0</td></tr>
<tr><td>M1672～M1687</td><td>#1</td></tr>
</table>

<table>
<tr><th>Send area M component assignment</th><th>Mode 10</th><th>Mode 11</th><th>Mode 12</th><th>Mode 13</th></tr>
<tr><td>M1688～M1703</td><td>#2</td><td rowspan="2">#1</td><td rowspan="2"></td><td rowspan="6"></td></tr>
<tr><td>M1704～M1719</td><td>#3</td></tr>
<tr><td>M1720～M1735</td><td>#4</td><td rowspan="2">#2</td><td rowspan="4">#1</td></tr>
<tr><td>M1736～M1751</td><td>#5</td></tr>
<tr><td>M1752～M1767</td><td>#6</td><td rowspan="2">#3</td></tr>
<tr><td>M1768～M1783</td><td>#7</td></tr>
<tr><td>M1784～M1799</td><td>#8</td><td rowspan="2">#4</td><td rowspan="4">#2</td><td rowspan="8">#1</td></tr>
<tr><td>M1800～M1815</td><td>#9</td></tr>
<tr><td>M1816～M1831</td><td>#10</td><td rowspan="2">#5</td></tr>
<tr><td>M1832～M1847</td><td>#11</td></tr>
<tr><td>M1848～M1863</td><td>#12</td><td rowspan="2">#6</td><td rowspan="4">#3</td></tr>
<tr><td>M1864～M1879</td><td>#13</td></tr>
<tr><td>M1880～M1895</td><td>#14</td><td rowspan="2">#7</td></tr>
<tr><td>M1896～M1911</td><td>#15</td></tr>
</table>

For example:

（8）In mode 10, the M components in the sending area allocated to station 0# (layer 1) are M1656~M1671. The PLC of station 0# can write values to M1656~M1671, and other stations (1#-~15#) can read directly Take the values from M1656 to M1671.

🕮 Notice

Once the PLC is configured with the N: N communication protocol, the D components D7700~D7763 and the M components M1400~M1911 will be used as public resources for network data exchange. Please pay attention when using these components in the program!

10.5.5 Enhanced refresh mode

In order to support more component sharing, VEDA VC series PLC provides modes 14~18. These modes are only applicable to single-layer structures with many shared components. M element and D element are expanded on the original basis (M1400-M1911, D7500-D7755)

The M element area (512) is shown in the following table:

<table>
<tr><th>M element assignment</th><th>Mode 14</th><th>Mode 15</th><th>Mode 16</th><th>Mode 17</th><th>Mode 18</th></tr>
<tr><td>M1400-M1415</td><td>#0</td><td rowspan="2">#0</td><td rowspan="4">#0</td><td rowspan="8">#0</td><td rowspan="16">#0</td></tr>
<tr><td>M1416-M1431</td><td>#1</td></tr>
<tr><td>M1432-M1447</td><td>#2</td><td rowspan="2">#1</td></tr>
<tr><td>M1448-M1463</td><td>#3</td></tr>
<tr><td>M1464-M1479</td><td>#4</td><td rowspan="2">#2</td><td rowspan="4">#1</td></tr>
<tr><td>M1480-M1495</td><td>#5</td></tr>
<tr><td>M1496-M1511</td><td>#6</td><td rowspan="2">#3</td></tr>
<tr><td>M1512-M1527</td><td>#7</td></tr>
<tr><td>M1528-M1543</td><td>#8</td><td rowspan="2">#4</td><td rowspan="4">#2</td><td rowspan="8">#1</td></tr>
<tr><td>M1544-M1559</td><td>#9</td></tr>
<tr><td>M1560-M1575</td><td>#10</td><td rowspan="2">#5</td></tr>
<tr><td>M1576-M1591</td><td>#11</td></tr>
<tr><td>M1592-M1607</td><td>#12</td><td rowspan="2">#6</td><td rowspan="4">#3</td></tr>
<tr><td>M1608-M1623</td><td>#13</td></tr>
<tr><td>M1624-M1639</td><td>#14</td><td rowspan="2">#7</td></tr>
<tr><td>M1640-M1655</td><td>#15</td></tr>
<tr><td>M1656-M1671</td><td>#16</td><td rowspan="2">#8</td><td rowspan="4">#4</td><td rowspan="4">#2</td><td rowspan="4">#1</td></tr>
<tr><td>M1672-M1687</td><td>#17</td></tr>
<tr><td>M1688-M1703</td><td>#18</td><td rowspan="2">#9</td></tr>
<tr><td>M1704-M1719</td><td>#19</td></tr>
</table>

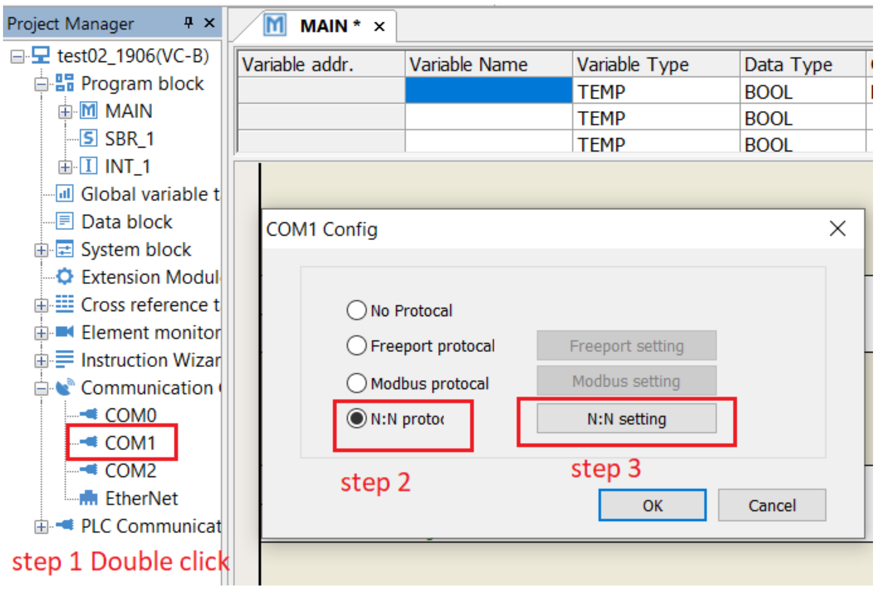
M1720-M1735	#20	#10	#5		
M1736-M1751	#21				
M1752-M1767	#22	#11			
M1768-M1783	#23				
M1784-M1799	#24	#12	#6	#3	
M1800-M1815	#25				
M1816-M1831	#26	#13			
M1832-M1847	#27				
M1848-M1863	#28	#14	#7		
M1864-M1879	#29				
M1880-M1895	#30	#15			
M1896-M1911	#31				

The D element area (256) is shown in the following table:

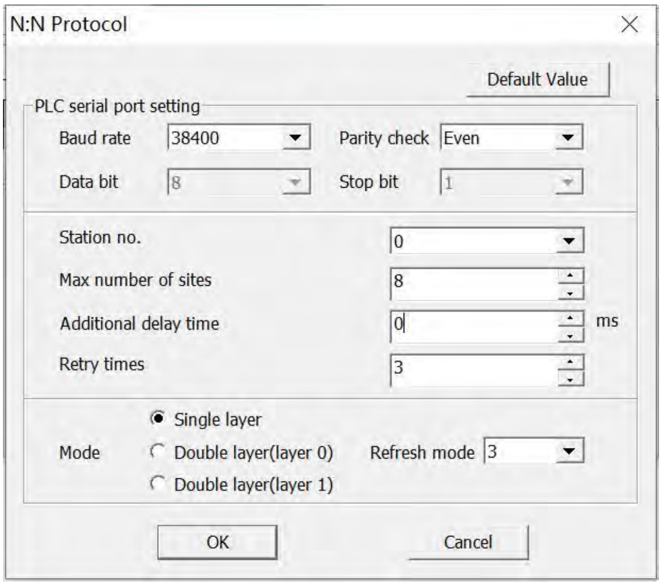
D component assignment	Model14	Model15	Model16	Model17	Model18
D7500～D7507	#0	#0	#0	#0	#0
D7508～D7515	#1				
D7516～D7523	#2	#1			
D7524～D7531	#3				
D7532～D7539	#4	#2	#1		
D7540～D7547	#5				
D7548～D7555	#6	#3			
D7556～D7563	#7				
D7564～D7571	#8	#4	#2	#1	
D7572～D7579	#9				
D7580～D7587	#10	#5			
D7588～D7595	#11				
D7596～D7603	#12	#6	#3		
D7604～D7611	#13				
D7612～D7619	#14	#7			
D7620～D7627	#15				
D7628～D7635	#16	#8	#4	#2	#1
D7636～D7643	#17				
D7644～D7651	#18	#9			
D7652～D7659	#19				
D7660～D7667	#20	#10	#5		
D7668～D7675	#21				
D7676～D7683	#22	#11			
D7684～D7691	#23				
D7692～D7699	#24	#12	#6	#3	
D7700～D7707	#25				
D7708～D7715	#26	#13			
D7716～D7723	#27				
D7724～D7731	#28	#14	#7		
D7732～D7739	#29				
D7740～D7747	#30	#15			
D7748～D7755	#31				

10.5.6 N: N Parameter settings

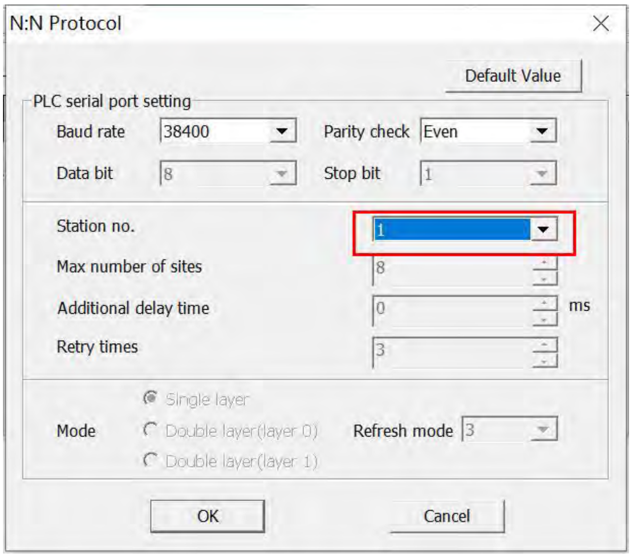
Select the **communication config** option in the Connect, double-click **COM1**, select the **N: N protocol** in the **COM1 config**, and activate the corresponding **N: N setting button**, as shown in the following figure:



Click the **N: N setting button** to enter the **N: N protocol** setting interface, as shown in the following figure:



Set the N: N parameters as shown above. The **station no.** should be set from0Start with the number and set it in sequence, you cannot connect multiple **PLC** is set to the same station number. Station 0 is the start-up and setup site of the network. Parameters such as the **max number of sites**, **additional delay time**, **retry times**, and **mode settings** only need to be set for station 0. The stations of other station numbers only need to set their own station numbers, except that the baud rate and parity must be the same as those of No. 0, as shown in the following figure:



The **max number of sites** refers to the total number of PLCs used in the network. If a total of 6 PLCs are used, please set them to 6, and set the station number of these 6 PLCs to 0~5. If you want to add 2 new PLCs to the network in the future without network interruption, you can set the max number of sites to 8, and the number of PLC stations to be added in the future to 6 and 7, respectively. When 6 and 7 are connected to the network, they will be automatically detected by N: N within 1 second and included in the data exchange with 0 to 5. .

10.6 Several Control Strategies

10.6.1 Master station determination

Station 0 is the default master station. Only station 0 can initialize and start the entire network. N: N related settings, such as refresh mode, additional delay time, number of retries, etc., must and can only be configured through station 0. During the online modification of the relevant configuration of station 0 and the download of system blocks, the standby master station will take over the network. When station No. 0 completes the system block download, the standby master station will give up the master station status to No. 0.

Master strategy in the network: The station with the smallest station number acts as the master.

10.6.2 Max number of sites

When setting the max number of sites, it is recommended to set the max number of sites to the total number of PLCs included in the actual network, and to program the station numbers in sequence starting from 0. When the max number of sites is set to N, the network only manages the stations from No. 0 to No. N-1. In particular, when the max number of sites set by the user is incorrect, that is, when the max number of sites is less than the actual number of PLCs included in the 485 network, PLCs with a station number greater than or equal to the max number of sites will not be able to broadcast However, it can receive broadcast data whose station number is less than the max number of sites.

10.6.3 Multi-master-slave (M:N)

N: N can be used to build a multi master and multi slave network. The meaning of "master" and "slave" here is: "master" is a PLC that can write its own M and D components and read M and D components from other sites; "Slave" is a PLC that can only read M and D components from other stations. Under the set max number of sites (the number of stations is also subject to the refresh mode), PLCs with station numbers less than the number of stations can be used as "master", while PLCs with station numbers greater than the number of stations can only be used as "slave". The slave station can only read the relevant M and D elements of the master station. These M and D elements correspond to each master station according to the refresh mode in the master station. You can refer to the N: N shared M and D element table. The slave station has no corresponding M and D elements in these tables.

10.6.4 Example of using N: N

There are 5 PLCs in total, the refresh mode is 3, and the station numbers are 0#～4#. It is hoped that the sum of D100 of 0# PLC and D305 of 2# PLC is stored in D500 of 4# PLC.

Programming to 0#: **MOV D100 D7700**

Programming for 2#: **MOV D305 D7716**

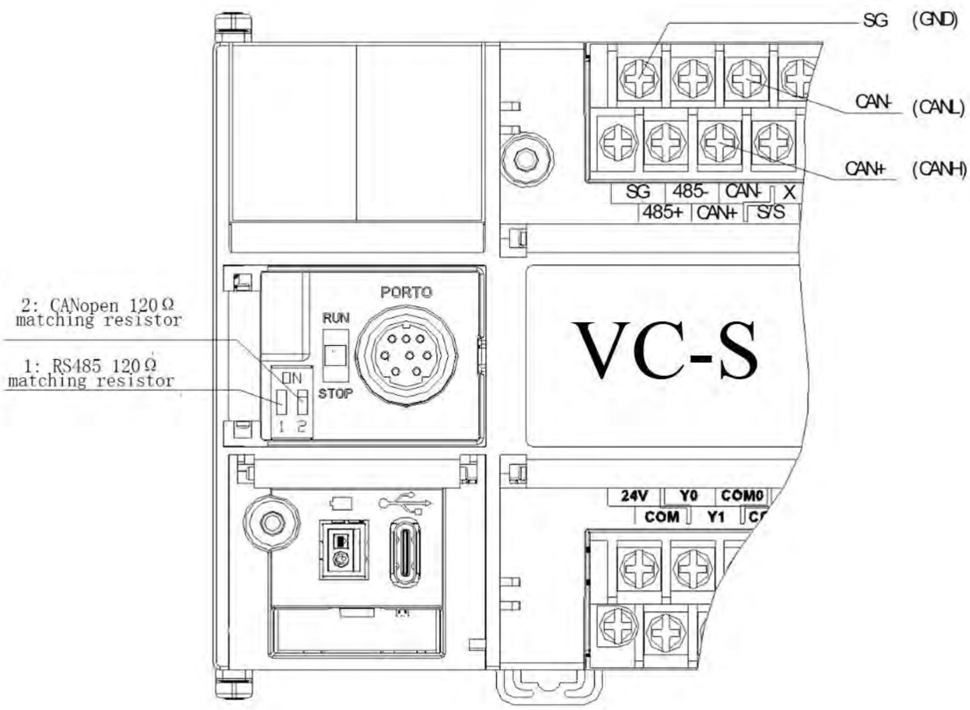
Programming to 4#: **ADD D7700 D7716 D500**

Note: This example is an N: N single-layer network, there are 5 PLC stations on the network, and the refresh mode is 3: each station can be assigned 8 D elements and 64 M elements. The D elements assigned to station 0# are D7700 to D7707, the D elements assigned to station 2# are D7716 to D7723, and the D elements assigned to station 4# are D7732 to D7739. Store the value of 0# station D100 in a write public area D7700 assigned to it on the network, and store the value of 2# station D305 in a write public area D7716 assigned to it on the network. In 4# PLC, add and store the common unit D7700 and D7716 to the local element D500.

10.7 CANopen Communication Settings

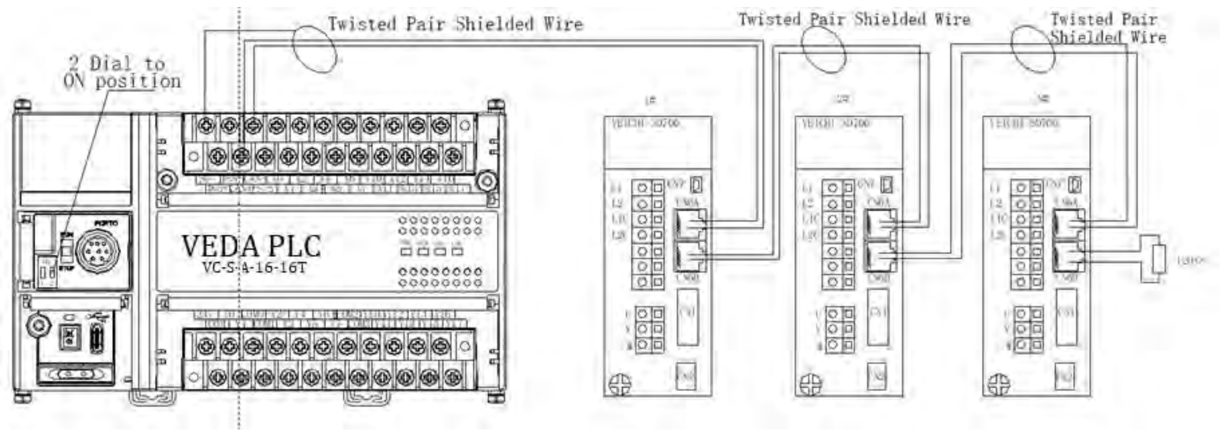
A. CANopen hardware port connection:

1. The VC-S series host has its own CAN hardware interface, and the corresponding interface pins are as follows:



B. CANopen 120Ω matching resistance

When forming a CAN network, the CAN+, CAN-, and SG lines of the device must be in one-to-one correspondence, and a 120-ohm CAN bus matching resistor should be added to both ends of the bus. The CAN bus wiring diagram is shown in the following figure:



10.7.1 CANopen Protocol selection

VC-S series supports CANopen communication standard protocol DS301

Software function	Main site	Slaves
Supporting agreement	DS301V4.02	DS301V4.02
Maximum number of TPDO	64	4
Maximum number of RPTO	64	4
Slave node	30	/
Baud rate and communication distance	1Mbps/25m 800kbps/50m 500kbps/100m 250Kbps/250m 125Kbps/500m 50kbps/1000m 20kbps/2500m 100Kbps 10Kbps	1Mbps/25m 800kbps/50m 500kbps/100m 250Kbps/250m 125Kbps/500m 50kbps/1000m 20kbps/2500m 100Kbps 10Kbps
Data exchange device	D0~d7999 (configurable)	SD400~SD415 (receiving area) ;SD432~SD447 (sending area)

10.7.2 CANopen Indicator

LED light display	CAN (green)	ERR (red)
Extinguish	No configuration	No errors
Bright	Working status	System error message
Flicker	Communication exception	Can communication abnormal or system error

10.7.3 CANopen Function explanation

NMT: Network Management

Network management services, application layer management, network status management and node ID allocation management, etc. The service mode is the master-slave communication mode: in the CAN network, there can only be one NMT master station and one or more slave stations. The master is used to control the slave status.

SDO: Server Data Object

A service data object that can access data in the slave device object dictionary through Indexes and sub-Indexes. This is mainly used in the slave configuration process. Every frame of SDO needs to reply to the confirmation.

PDO: Process Data Object

Process data objects, mainly used to transmit real-time data. Data transfers are limited to 1 to 8 bytes. The transmission of PDO data is divided into two ways: synchronous and asynchronous. The PDO frame is the main data exchange frame after the slave is started.

SYNC: Synchronous

The synchronization service adopts the master-slave communication mode. The master node sends the SYNC object regularly, and the SYNC slave node receives it and executes the task synchronously. This frame is mainly used for synchronous transmission of PDO.

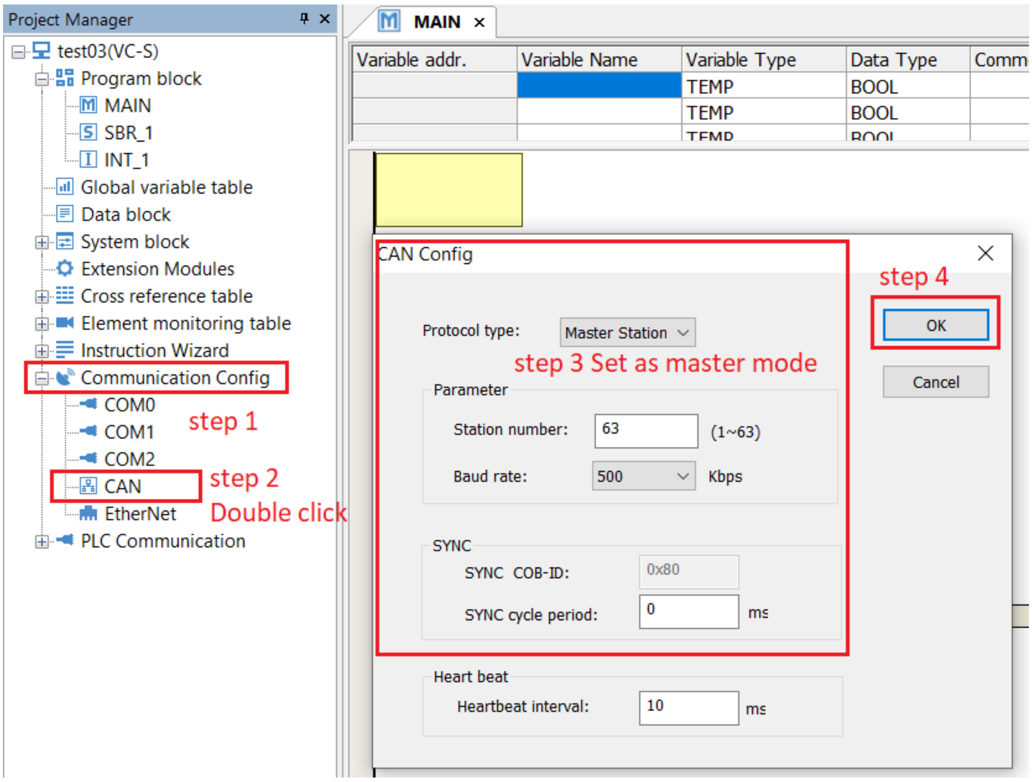
COB_ID: Communication Object Identifier

Each CANopen frame starts with a COB_ID, which is used as the communication object identifier of the CAN frame. COB_ID is not equal to the slave station number. However, it is generally associated with the slave station number by default.

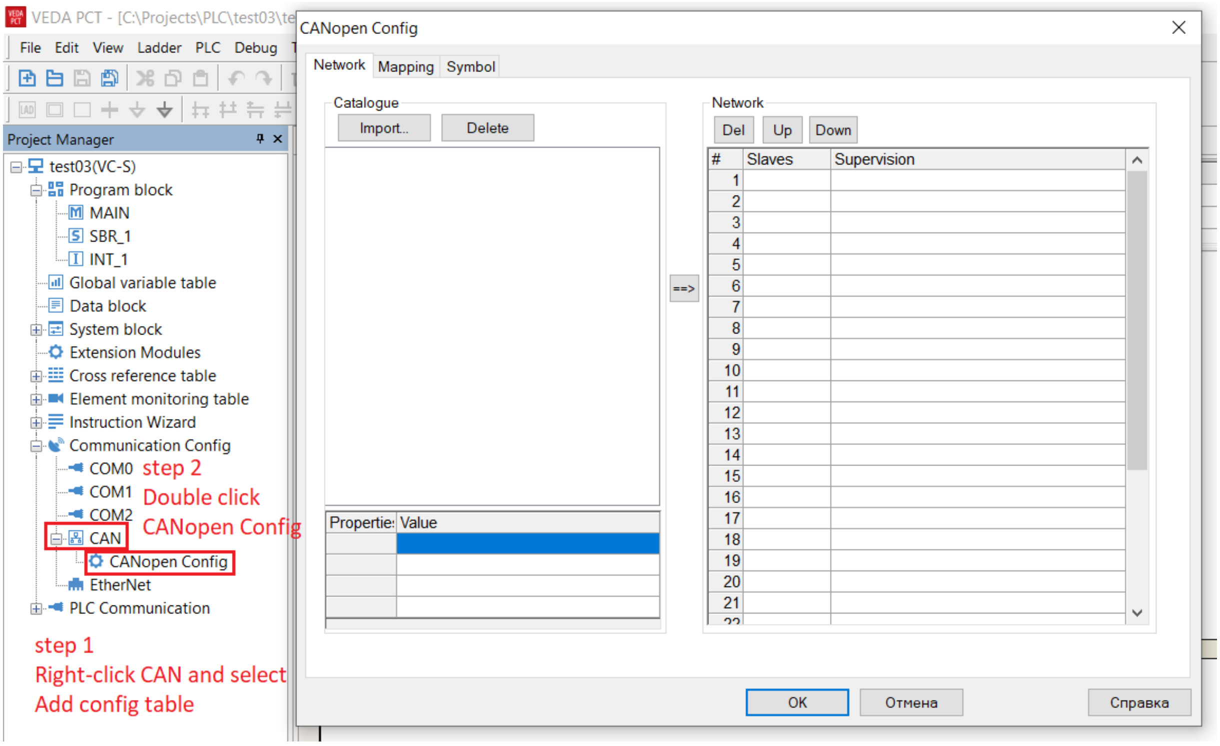
10.7.4 CANopen Master/slave configuration

A. The configuration of the CANopen master station is shown in the figure below

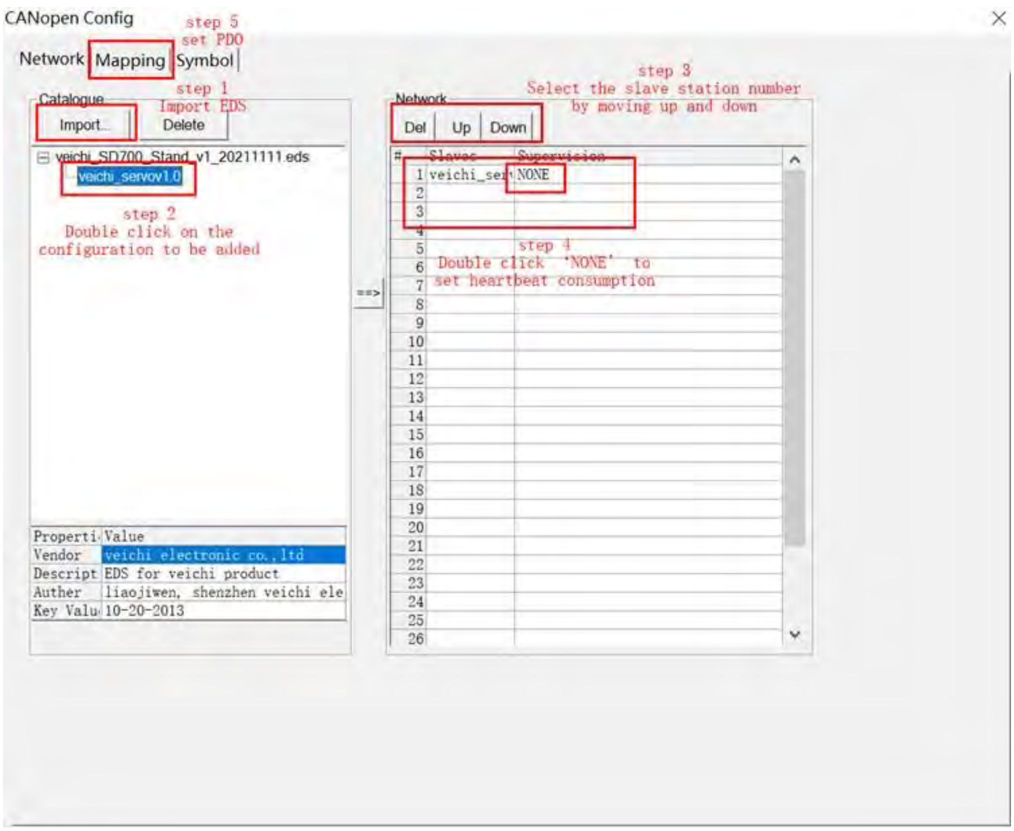
(1) Open the VEDA PCT software, select CAN in the communication configuration, double-click the "CAN" protocol type and select the CANopen master station.



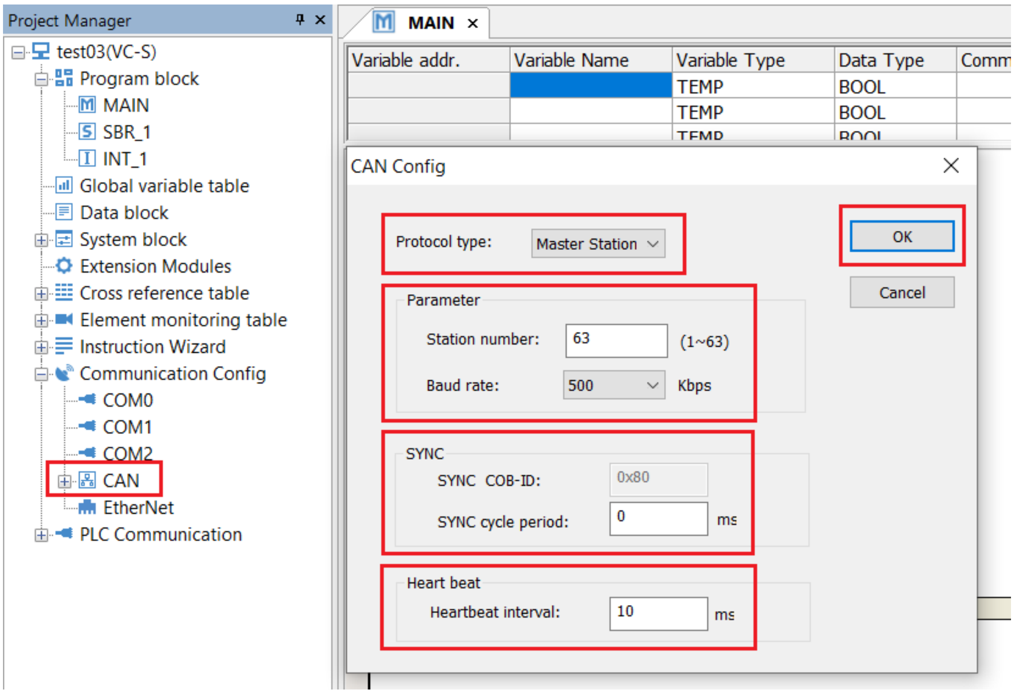
(2) Right-click CAN to add CANopen configuration, and then double-click CANopen configuration to configure the slave device as shown in the figure below:



(3) Import the EDS file of the CANopen slave device (the EDS file is obtained from the device supplier)



B. **Master interface information configuration**



① Double-click "CAN" to configure the master station;

② Select the protocol type [Master];

③ Set the station number of the master station and the baud rate at which the master station takes effect;

C. Synchronize

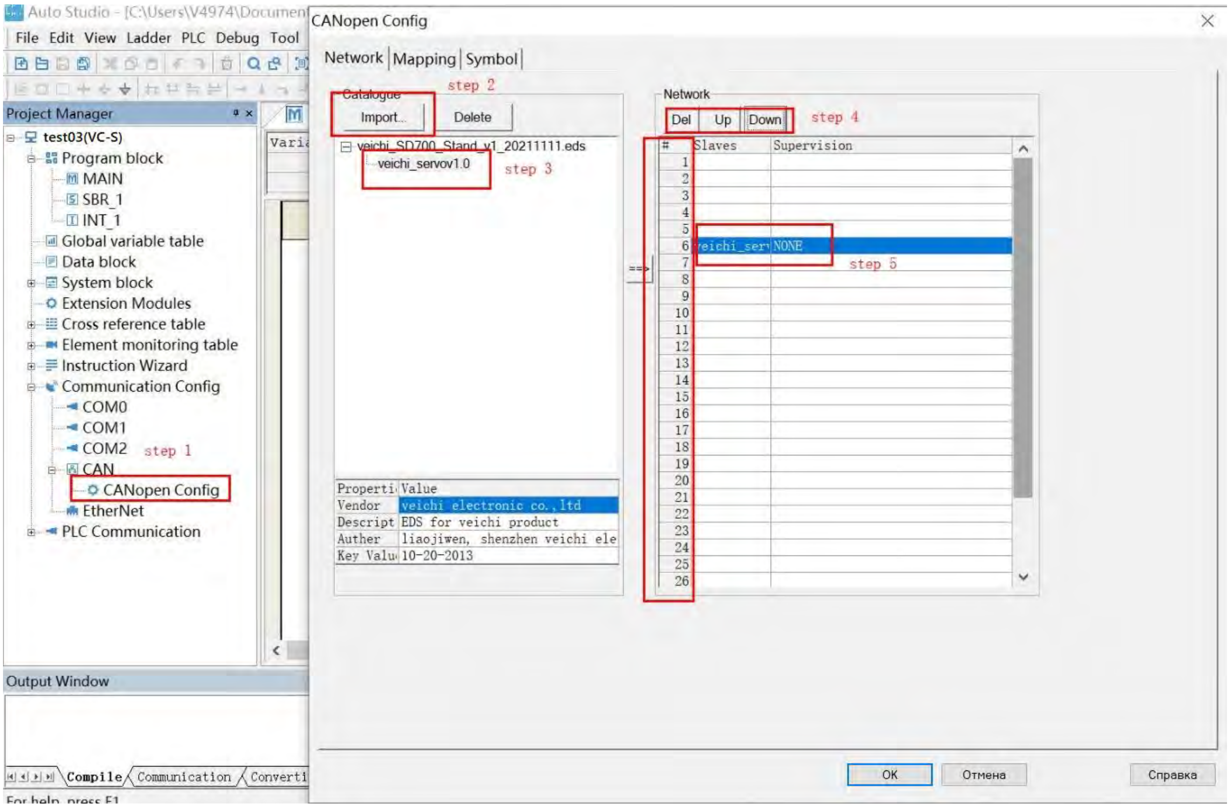
(1) COB-ID: Synchronous frame sending D, this item uses the default value of 0x80, which is not allowed to be set.

(2) Set the synchronization time, the station will send the synchronization frame cyclically according to the time set in "Synchronization Period (ms)", you need to check "Synchronization Cycle" in the corresponding PDO transmission type.

D. Heartbeat

(1) Set the heartbeat time, the station will send heartbeat frames cyclically according to the time set by "production time (ms)". Production time (ms): The cycle period for sending heartbeats. The default is 10, the unit is ms (the default is 0, no heart is sent).

E. Slave parameter setting (take VC-S slave as an example) as shown in the figure below:



a) Slave network configuration

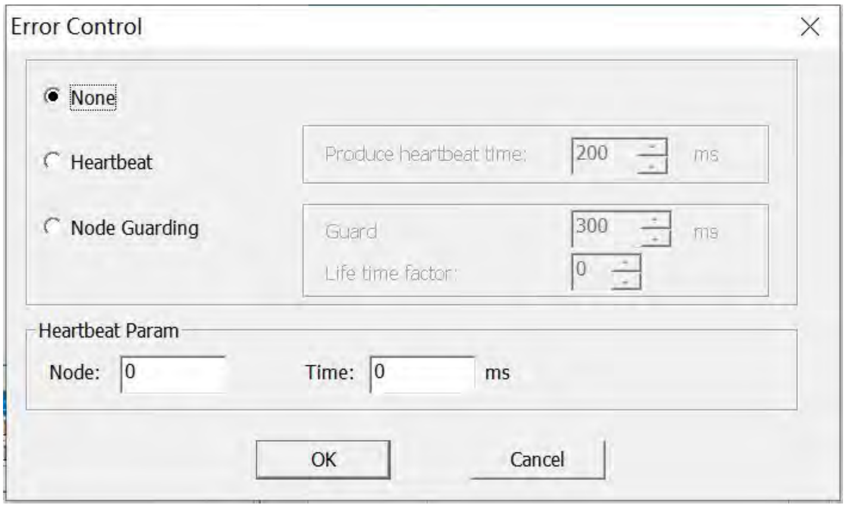
① Double-click [CANopen Configuration] to enter the network configuration;

② Import the EDS file of the VC-S slave;

③ Select the VC-S slave device that needs to be added to the network, double-click [VC Series PL] to add the slave to the network, the software will automatically join the network in sequence (1-31),

④ Slave device station number setting, select the name of the slave station whose station number needs to be adjusted, and realize through Up (move up) and Down (move down) or Del (delete) the slave station device from the network. (The configuration in the above figure indicates that the station number of VC-S is 4)

⑤ Double-click [NONE] to enter the error control setting, and select None by default. As shown below:



(1) [Use Heartbeat protocol]: The slave station will generate a heartbeat according to the set time, which is not selected by default. After the slave station selects [Use Heartbeat protocol], the master station monitors the heartbeat status of the slave station by default.

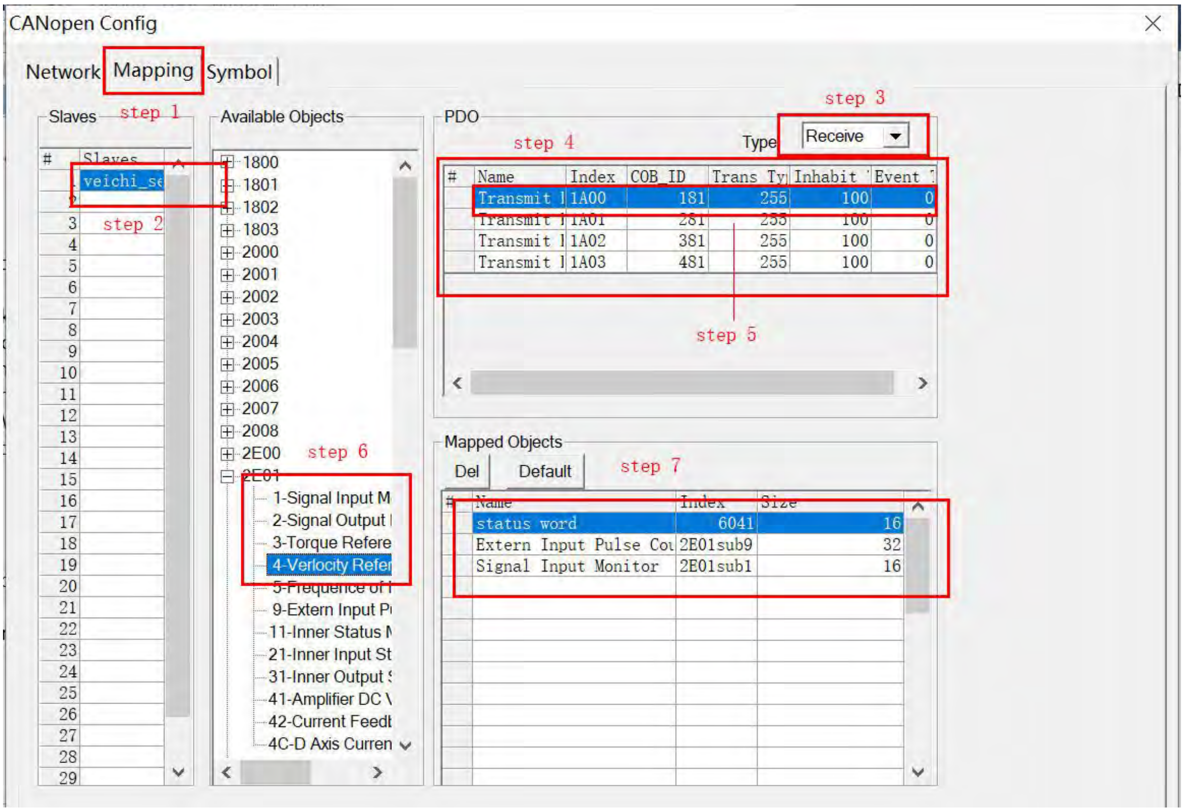
(2) [Node Heartbeat production time]: The time when the slave station heartbeat is sent cyclically. Unit (ms)

(3) [Use Node Guarding Protocol]: It is a network evaluation function that monitors each other between the master station and the slave station that return the frame. Only one of the heartbeat and node guard functions can be selected. temporarily reserved

(4) [Heartbeat consumption parameter]: This function is used to set the heartbeat of other sites that this slave will monitor. Nodes represent sites that need to be monitored. This feature is not selected by default. (This function also requires the slave to support the heartbeat monitoring function).

b) Send PDO/Receive PDO mapping settings

① Click [Map] to select the slave that needs to configure PDO, and the interface will appear as shown below:

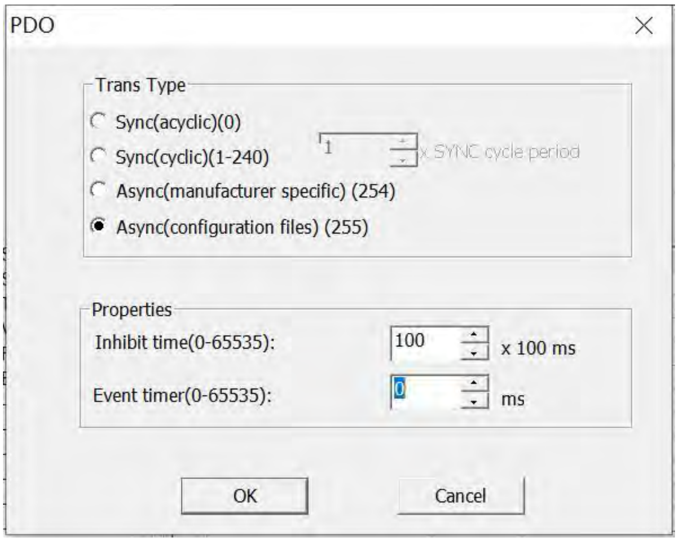


② Select the slave station that will set POD;

③ Select POD type: **receive (PDO):** data sent by the master to the slave; **send (PDO):** data sent from the slave to the master;

④ The PDO that the slave EDS takes effect by default; (EDS file provided by the equipment manufacturer)

⑤ PDO property settings; double-click a PDO, the following interface will appear:



Transmission type:

Type	Data sending conditions	Data validation conditions
Synchronous acyclic (0)	Data changes and a sync frame is received	It does not take effect immediately after receiving the data, it needs to receive a synchronization frame to take effect
Synchronous loop (1-240)	Data is sent after receiving the corresponding "sync period" frame synchronization	It does not take effect immediately after receiving the data, it needs to receive a synchronization frame to take effect
Asynchronous (manufacturer event) (254)	Not support	Not support
Asynchronous (profile events) (255)	The data changes or meets the event time and the frequency of change is less than the suppression time	Effective immediately

【Synchronization period】: It is valid after selecting cycle-synchronization (1-240), and set the number of synchronization periods;

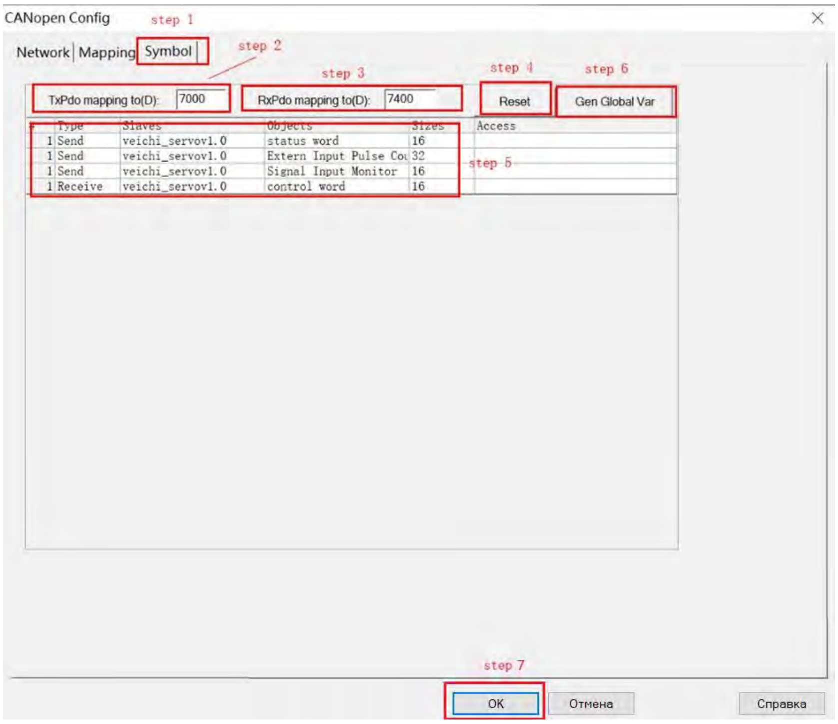
【Suppression time】: It can be set after selecting asynchronous-device configuration file specification (Type255), if it is 0, this function is invalid. When not 0, it is the minimum interval for frame transmission.

【Event timer】: It can be set after selecting asynchronous-device configuration file specification (Type255), if it is 0, this function is invalid. When it is not 0, it indicates the period of timed transmission. (This sending situation is also limited by the suppression time)

⑥ Add object operation; select the PDO type【receive (POD) or send (POD)】, double-click the selected object dictionary to add a mapping object as shown in ⑦.

⑦ The mapped object can be selected and then [Delete] to map the object.

c) 【Symbol】Master PDO maps D register configuration. Selecting【Symbol】will bring up the following interface:



① Select 【Symbol】 to map PDO to D register configuration;

② Set 【Send PDO mapping start address (D】: 7000 (default); users can freely choose D0~D7999;

③ Set 【Receive PDO mapping start address (D】: 7400 (default); users can freely choose D0~D7999;

④ After the PDO mapping address is set, click 【Reset Access Address】 and the system will automatically allocate the D register as shown in ⑤;

⑤ The system automatically allocates the D register according to the set starting address;

⑥ Click 【Generate Global Variables】, the system will generate the automatically allocated D registers in the global variable table, and automatically add comments such as station number, type, slave station name, object dictionary, etc. to the D register, which is convenient for user program debugging.

⑦ After completing the above a), b), and c) operations, click 【OK】 to complete the CANopen configuration.

10.7.5 CANopen SDO Read and write commands

A. CANNMT state switching command

<table>
<tr><td colspan="15" rowspan="2">Ladder Diagram:
├──┤ ├──[CANNMT S1]</td><td>Applicable models</td><td>VC-S VC-P</td></tr>
<tr><td>Affect the flag</td><td></td></tr>
<tr><td colspan="15">Command list: CANNMT (S1)</td><td>Step size</td><td>3</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="13">Applicable devices</td><td colspan="2">Index</td></tr>
<tr><td>S1</td><td>INT</td><td></td><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">√</td></tr>
</table>

- Operand Description

S1: Switch state, value range 1-4, 1 reset CANopen communication; 2 reset CANopen node; 3 switch to preprocessing mode; 4 switch to run mode.

- Function Description

When the power flow is valid, a message is sent to make the CANopen network enter the specified state.

- Precautions

When the instruction is being executed, when encountering PLC RUN to STOP, this instruction may not be executed.

- Example of use

SM440 CANopen command execution is completed (=1 is completed, =0 for the rest).
SM441 CANopen command execution error (=1 command error, =0 no error).

SM442 CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

B. CANSDORD read command

Ladder Diagram:	Applicable models	VC-S VC-P
[CANSDORD S1 S2 S3 S4 D1]	Affect the flag	
Command list: CANSDORD *(S1) (S2) (S3) (S4) (D1)*	**Step size**	**12**

Operand	Type	Applicable devices													Index
S1	INT	Constant	D												√
S2	INT	Constant	D												√
S3	INT	Constant	D												√
S4	INT	Constant	D												√
D1		D													√

- Operand Description

S1: Device address range 1-126.

S2: SDO Index.

S3: SDO sub Index.

S4: read data length, (1, 2, 4 respectively, BYTE, WORD, DWORD).

D1: The data storage address read back, (for BYTE, WORD only occupies 16 bits and is stored in the lower 16 bits).

- Function Description

When the power flow is valid, send a message to read the Index data of the specified node.

- Precautions

When the instruction is being executed, when encountering PLC RUN to STOP, the instruction may not be executed. Please make sure that the Index and sub-Index read are valid, otherwise an error will be returned.

- Example of use

SM440 CANopen command execution is completed (=1 is completed, =0 for the rest).
SM441 CANopen command execution error (=1 command error, =0 no error).

SM442 CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

C. CANSDOWR write command

Ladder Diagram:	Applicable models	VC-S VC-P
[CANSDOWR S1 S2 S3 S4 D1]	Affect the flag	
Command list: CANSDOWR(S1)(S2)(S3)(S4)(D1)	**Step size**	**12**

Operand	Type	Applicable devices	Index

S1	INT	Constant	D												√
S2	TIN	Constant	D												√
S3	TIN	Constant	D												√
S4	TIN	Constant	D												√
D1	DINT	D													√

- Operand Description

S1: Device address range 1-126.

S2: SDO Index.

S3: SDO sub Index.

S4: Write data length, (1, 2, 4 respectively, BYTE, WORD, DWORD).

D1: The written data storage address, (for BYTE, WORD only occupies 16 bits and is stored in the lower 16 bits).

- Function Description

When the power flow is valid, send a message and write the Index data of the specified node.

- Precautions

When the command is being executed, when encountering PLC RUN to STOP, it may cause the command to fail to be executed. Please make sure that the written Index and sub-Index are valid, otherwise an error will be returned.

- Example of use

SM440 CANopen command execution is completed (=1 is completed, =0 for the rest).
SM441 CANopen command execution error (=1 command error, =0 no error).

SM442 CANopen command is being executed (=1 command is being executed, =0 no command is being executed), mainly to prevent multiple CANopen commands from being executed at the same time.

10.7.6 CANopen communication troubleshooting

A. Routine inspection steps

1) Check if the device supports CANopen

Equipment	Inspection method
PLC	VC-S VC-P series
Servo/Inverter	Check whether the software version supports

2) Check the matching resistance

All devices are powered off. Use a multimeter to measure the resistance between CAN+ and CAN- at either end of the network. It should be around 60Ω. If it is too small, it means that not only two ends of the network are connected to matching resistors, but there are also wrong connections at other locations. Input, disconnect the wrong matching resistor. If the network is only connected to a single-ended matching resistance of about 120Ω, the quality of network communication is very poor. If the matching resistance is not connected at all, the network cannot communicate. Please access the matching resistors of the first and last sites of the network.

3) Check baud rate and station number

The baud rate and station number settings do not match. Check whether the baud rate and station number are configured correctly. The device baud rate or station number will take effect only after power-on or reset operation. Baud rate The communication distance is related to the baud rate. Please refer to Chapter 10 10.7.1

4) Check the wiring

Whether the connection between the CAN communication port of the PLC and the servo or inverter is correct, and ensure that the shielding layers of all devices are connected together.

5) Other

If the on-site interference is very large, it is recommended to add a magnetic ring to the PLC communication port or try to reduce the communication baud rate.

B. See the following table for the special components of CAN communication:

CANopen special SM auxiliary relay

Address number	Name	Function	R/W
SM440	Canopen instruction completed	=1 completion of execution, =0 the rest	R/W
SM441	Canopen command error	=1 command error, =0 no error	R/W
SM442	Canopen instruction is being executed	=1 instruction is being executed, =0 no instruction is being executed	R
SM443	VEDA slave device encryption enable	0: no encryption1:encryption	R/W

The CANopen special registers are shown in the following table:

Address number	Data length	Initial value	Function	R/W
SD340	16	0	The configured network node (1-16) indicates whether sites 1-16 are configured. When the bit is 1, it indicates that the corresponding site is configured. Bit 0 represents station 1 and bit 15 represents station 16.	R
SD341	16	0	Indicates whether the 17-32 site is configured, when the bit is 1, it means it is configured, and the small site is low. Bit 0 represents station 17, bit 15 represents station 32	R
SD342	16	0	Network baud rate, 1-8, Corresponding to 10k, 20k, 50k, 125k, 250k, 500k, 800k, 1m	R
SD343	16	0x7f	Cob-id synchronization	R
SD344	16	0	Synchronization period (1-1000ms)	R
SD345	16	0	The first address of the image area (eg: d1000 displays 1000)	R
SD346	16	0	The first address of the image area (eg: d1000 displays 1000)	R
SD350	16	0	Online node in the network, when the bit is 1, it means online. For stations 1-16, bit 0 represents station 1, and bit 15 represents station 16.	R
SD351	16	0	When the online node in the network is 1, it means online. Stations 17-32, bit 0 represents station 17, bit 15 represents station 32	R

<table>
<tr><th>Address number</th><th>Data length</th><th>Initial value</th><th>Function</th><th>R/W</th></tr>
<tr><td>SD352</td><td>16</td><td>0</td><td>Canopen network status
<table>
<tr><th>Bit</th><th>Error type</th><th>Remark</th></tr>
<tr><td>Bit0</td><td>Optional module error</td><td>0: no error; 1 at least one mo</td></tr>
<tr><td>Bit1</td><td>Required module error</td><td>0: no error; 1 at least one conf</td></tr>
<tr><td>Bit2</td><td>The required module has an error in network monitoring</td><td>Reserve</td></tr>
<tr><td>Bit3</td><td>Configuration process error</td><td>0: no error; 1 with error</td></tr>
<tr><td>Bit4</td><td>Network communication error</td><td>0: no error; 1 with error</td></tr>
<tr><td>Bit5</td><td>One or more slaves have errors and are not in operation</td><td>0: no error; 1 with error</td></tr>
<tr><td>Bit6</td><td>Canopenthe length of the pdo received by the master is too short</td><td>0: no error; 1 with error</td></tr>
<tr><td>Bit7~bit10</td><td>Reserve</td><td></td></tr>
<tr><td>Bit11</td><td>Whether the master is alone on the bus</td><td>0: no; 1 yes</td></tr>
<tr><td>Bit15~bit12</td><td>Reserve</td><td></td></tr>
</table></td><td>R</td></tr>
<tr><td>SD353</td><td>16</td><td>0</td><td>Canopen instruction error status</td><td>R</td></tr>
<tr><td>SD354</td><td>16</td><td>0</td><td>Emcy id</td><td>R</td></tr>
<tr><td>SD355</td><td>16</td><td>0</td><td>Emcy data</td><td>R</td></tr>
<tr><td>SD359</td><td>16</td><td></td><td>Communication error status with canopen master module,
<table>
<tr><td>Bit0</td><td>The PLC cannot detect the canopen master. Canopen is configured, but cannot communicate</td><td>0: ok; 1 not detected</td></tr>
<tr><td>Bit1</td><td>PLC download canopen configuration error</td><td>1 error,
Configuration error occurred 3 times,
Set, and canopen
Main module communication stopped</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Bit2
Bit3</td><td colspan="2">PLC data refresh error
01 an error occurred during canopen data refresh
10 canopen data refresh process timed out</td></tr>
<tr><td>Bit4
Bit5</td><td>PLC reads canopen master network status error</td><td>01 an error occurred
10 timeout occurred reading status</td></tr>
</table></td><td>R</td></tr>
</table>

Address number	Data length	Initial value	Function			R/W
SD360	16		Canopen master status information			R
			Bit0	System self-check succeeded	0: uninitialized successfully; 1: succ	
			Bit0	System self-check succeeded	0: uninitialized successfully; 1: succ	
			Bit1	Network initialization/configuration start	0: unsuccessful; 1: start	
			Bit2	An error occurred while configuring the slave	0: no error; 1: at least one module does not m	
			Bit3	Critical error sign	0: no error ;1: critical error, must restart	
			Bit4	Error code	=0 ok =1 download error =2 initialization error	
			Bit5			
			Bit6			
			Bit7			
			Bit8	Master status	=0x01, initialize =0x02, reset node =0x04, reset communication =0x10, pre-operation =0x20, operate =0x30, stop	
			Bit9			
			Bit10			
			Bit11			
			Bit12			
			Bit13			
			Bit14			
			Bit15	Reserve	Reserve	

<table>
<tr><th>Address number</th><th>Data length</th><th>Initial value</th><th>Function</th><th>R/W</th></tr>
<tr><td>SD361~SD392</td><td>16</td><td></td><td>Canopen slave 1 status information tocanopen slave 32 status information
<table>
<tr><th>Bit</th><th>Error type</th><th>Remark</th></tr>
<tr><td>Bit0</td><td>Is the user configured</td><td>=1 configuration
=0 not configured</td></tr>
<tr><td>Bit1</td><td>Slave online</td><td>=0 No such slave on canopen network
=1 has this slave</td></tr>
<tr><td>Bit2</td><td>Slave ready to start</td><td>=0 not ready
=1 ready</td></tr>
<tr><td>Bit3</td><td>Slave configuration is complete</td><td>=0 Configuration not complete
=1 configuration complete</td></tr>
<tr><td>Bit4
Bit5
Bit6
Bit7</td><td>Error code</td><td>=0 OK
Bit4=1 EMCY error
Bit5=1 configuration error
Bit6=1 PDO length is too short
Bit7=1 Life Guard or Heartbeat Mistake
=F other errors
= other reserved</td></tr>
<tr><td>Bit8~bit15</td><td>Slave Status</td><td>=0x00 is in initialization state
=0x04 is in stop state
=0x7f is in pre-operational state
=0x05 is in operation
=0xff Unknown (Supervision status is configu</td></tr>
</table>
</td><td></td></tr>
<tr><td>SD400~SD415</td><td>16</td><td></td><td>Do Canopen slave data receiving area</td><td>R</td></tr>
<tr><td>SD432~SD447</td><td>16</td><td></td><td>Do Canopen slave data transmission area</td><td>R/W</td></tr>
</table>

10.7.7 Summary of axis control instructions

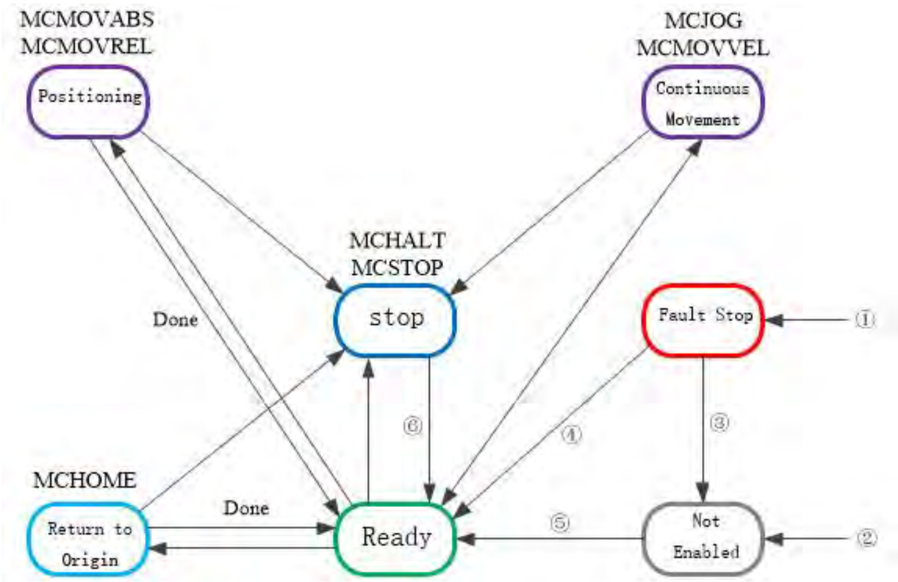
Command name	Function	VC-S VC-P
MCPOWER	Enable	√
MCRESET	Reset	√
MCSTOP	Stop	√
MCHALT	Pause	√
MCRDPOS	Read the current actual position	√
MCRDVEL	Read the current actual speed	√
MCRDPAR	Read parameter	√
MCWRPAR	Write parameters	√
MCHOME	Return to origin	√
MCMOVABS	Absolute positioning	√
MCMOVREL	Relative positioning	√

MCMOVVEL	Speed mode	√
MCJOG	Jog (speed mode)	√

10.7.8 Axis control command state machine description

(1) Axis state machine

Each servo execution unit acts as a motion control axis, and the control of the axis is based on the following state machine.



(2) Axis state description

State	Describe
Disable (disable)	Power-on initialization state In this state, the motion control commands are invalid, and the servo execution unit is not enabled State transition of no. ⑤: the mcpower command is valid, the host sends 0x06, 0x07, 0x0f control commands to the servo 0x6040 object dictionary successively, and the servo is in the enabled state after completion. State transition of no. ②: the command mcpower in other states (non-faults) is invalid, the host sends 0x00 to the servo 0x6040 to disable the servo execution unit, and when the servo 0x6041 reports that it is in a non-operational state, the state transition is completed. No. ③ state transition: mcreset is executed in the fault state. At this time, the servo 402 state machine is in the fault state, sending 0x80 to the servo 0x6040, and the servo 0x6041 feedback fault reset and is not in the enabled state.
Error stop	Highest priority ① State transition: In other states, the axis itself has a fault or the servo 402 state is switched to the fault state. Certain axis control faults do not cause servo shutdown
Ready (standstill)	Servo execution unit is enabled and fault-free No other valid commands ④ state transition: mcreset is executed in this fault state and the servo execution unit is in the enabled state State transition of no. ⑥: the shutdown is completed, the mcstop execution completion sign is valid and the mcstop busy sign is invalid
Stopping	The execution unit is executing the stop command according to the set stop mode
Position (discrete motion)	Mcmovabs is being executed Mcmovrel is being executed When these commands are executed, 0x0f and 0x1f commands are sent to the servo 0x6040 successively. This state servo is in pp control mode

Continuous motion	MCMOVVEL is being executed MCJOG is being executed When these commands are executed, 0x0f and 0x1f commands are sent to the servo 0x6040 successively. This state servo is in PV control mode
Return to the origin (Homing)	MCHOME is being executed When these commands are executed, 0x0f and 0x1f commands are sent to the servo 0x6040 successively. This state servo is in HM control mode

10.7.9 CANopen Axis control instruction description

If you only use the CANopen axis control commands other than "MCHOME origin return", you only need to configure the CANopen master station. If you want to use the "MCHOME origin return" axis control commands, the origin return method and speed need to be set in the object dictionary of the CANopen configuration interface. . For the specific configuration process, please refer to "10.7.4 CANopen Master/Slave Configuration".

10.7.9.1 MCPOWER: Enable

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCPOWERR S1 D1 D2	**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
D1	BOOL				M	S										√
D2	INT								D						R	√

■ Overview

Control the servo axis to enable or disable.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

D1: Axis state: output of the actual state of the axis, ON means the axis is enabled, OFF means the axis is not enabled.

D2: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

Note: Only one MCPOWER instruction can be used per axis.

The MCPOWR instruction writes the corresponding control word (6040h) according to the read status word (6041h), so that the axis enters the enabled state. The writing correspondence between the status word (6041h) and the control word (6040h) is shown in the table below.

Energy flow	Status word (6041h)		Control word (6040h)	
ON	Not ready to switch on	xxxx xxxx x0xx 0000b	Shutdown	0000 0000 0000 0110b
	Switch on disabled	xxxx xxxx x1xx 0000b		
	Ready to switch on	xxxx xxxx x01x 0001b	Switch on	0000 0000 0000 0111b
	Switched on	xxxx xxxx x01x 0011b	Switch on + enable operation	0000 0000 0000 1111b
	Fault reaction active	xxxx xxxx x0xx 1111b	-	xxxx xx00 xx00 xxxxb

	Fault	xxxx xxxx x0xx 1000b		
	other		-	xxxx xxxx xxxx xxxxb
OFF	Ready to switch on	xxxx xxxx x01x 0001b	Disable voltage	0000 0000 0000 0000b
	Switched on	xxxx xxxx x01x 0011b		
	Operation enabled	xxxx xxxx x01x 0111b		
	other		-	xxxx xx00 xx00 xxxxb

where x represents an arbitrary value (status word) or remains unchanged (control word).

10.7.9.2 MCRESET: Reset

Ladder Diagram:		**Applicable models**	**VC-S VC-P**
		Affect the flag	
Command list: MCRESET S1 D1 D2		**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
D1	BOOL				M	S										√
D2	INT								D						R	√

■ Overview

Resets axis-related errors, putting the axis into a "ready" or "disabled" state.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

D1: Complete: Reset operation execution complete output.

D2: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

The writing correspondence between MCRESET status word (6041h) and control word (6040h) is shown in the following table.

Energy flow	Status word (6041h)		Control word fault reset (6040h.bit7)
ON	Switch on disabled	xxxx xxxx x1xx 0000b	0
	Operation enabled	xxxx xxxx x01x 0111b	
	Fault	xxxx xxxx x0xx 1000b	1
	-	other	x
OFF	-	xxxx xxxx xxxx xxxxb	0
	-	xxxx xxxx xxxx xxxxb	x

where x represents an arbitrary value (status word) or remains unchanged (control word).

10.7.9.3 MCSTOP: Stop

Ladder Diagram:		**Applicable models**	**VC-S VC-P**
		Affect the flag	
Command list: MCSTOP S1 D1 D2 D3		**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
D1	BOOL				M	S										√
D2	BOOL				M	S										√
D3	INT								D						R	√

■ Overview

The control axis stops and enters the "stop" state, no longer responding to any command to move the axis.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

D1: Completion: The execution of the instruction is completed, and the axis has stopped.

D2: Busy: the instruction is being executed.

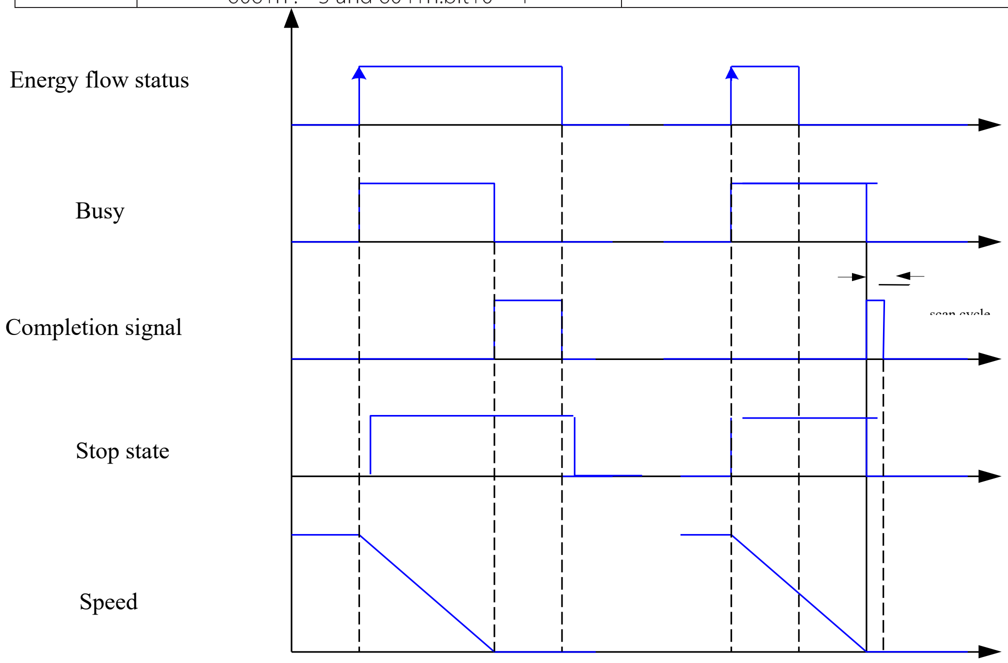
D3: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

Note: ①After the MCSTOP instruction is executed, the power flow of the MCMOVABS, MCMOVREL, MCMOVVEL, MCJOG instructions must be re-conducted if the original ON state is turned on.

② The deceleration at the time of stop is the deceleration set by the previous axis motion.

MCSTOP instruction CANOpen object operation steps

Step	Action/condition	Illustrate
1	6040 = 0x0f	Reset control word
2	6040 = 0x10f 6060h = 1 60ffh = 0	Control word triggers motion stop Switch to speed mode Write zero for target speed
3	606Ch = 0	Wait for stop to complete
	6061h = 3 and 6041h.bit13 = 1	
	6061h != 3 and 6041h.bit10 = 1	



MCSTOP timing diagram

10.7.9.4 MCHALT: Pause

Ladder Diagram:	Applicable models	VC-S VC-P
	Affect the flag	

Command list: MCHALT S1 D1 D2 D3									**Step size**			**11**				
Operand	Type	Applicable devices														Index
S1	INT	Constant														
D1	BOOL				M	S										√
D2	BOOL				M	S										√
D3	INT								D						R	√

■ Overview

The control terminates the current motion and continues to respond to other commands to move the axis when the power flow is disconnected.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

D1: Complete: The command execution is completed, and the axis has stopped.

D2: Busy: the instruction is being executed.

D3: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

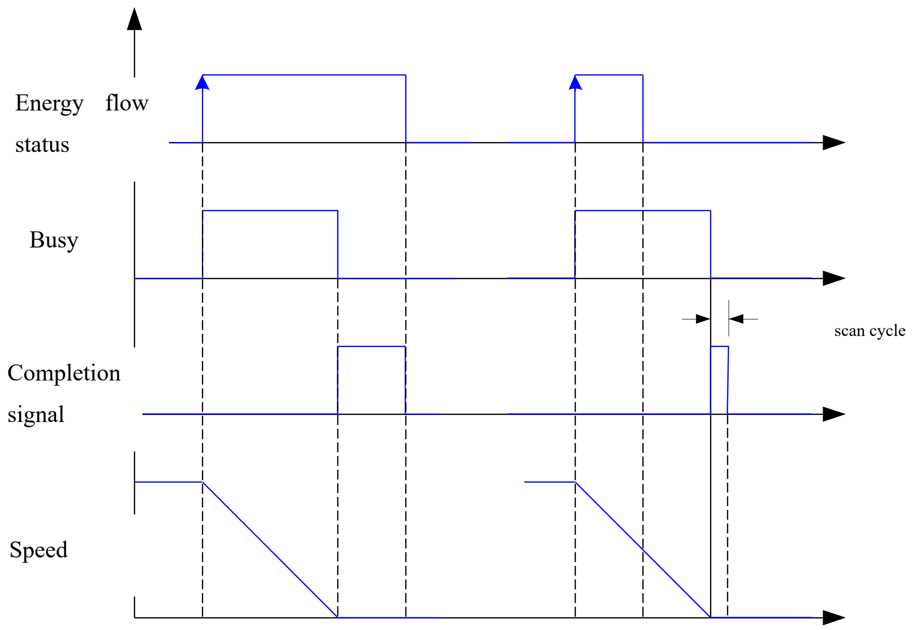
Note:

① During the execution of MCMOVABS, MCMOVREL, MCMOVVEL and MCJOG, the axis stops moving when the power flow of the MCHALT instruction is turned on. When the power flow of the MCHALT instruction is disconnected, the above instructions continue to be executed (the final position after the execution of the MCMOVREL instruction is the current stop position increase the set target position).

② The deceleration at stop is the deceleration set last to make the axis move.

MCHLAT instruction CANOpen object operation steps.

Step	Action/condition	Illustrate
1	6040 = 0x0f	Reset control word
2	6040 = 0x10f 6060h = 1 60ffh = 0	Control word triggers motion stop Switch to speed mode Write zero for target speed
3	606Ch = 0 6061h = 3 and 6041h.bit13 = 1 6061h != 3 and 6041h.bit10 = 1	Wait for stop to complete



MCHALT Timing Diagram

10.7.9.5 MCRDPOS: Read current actual position

Ladder Diagram:	Applicable models	VC-S VC-P
	Affect the flag	
Command list: MCRDPOS S1 D1	Step size	11

Operand	Type	Applicable devices														Index
S1	INT	Constant														
D1	DINT							D						R		√

■ Overview

Read the current actual position.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

D1: Position: the current actual position of the axis.

10.7.9.6 MCRDVEL: Read current actual speed

Ladder Diagram:	Applicable models	VC-S VC-P
	Affect the flag	
Command list: MCRDVEL S1 D1	Step size	11

Operand	Type	Applicable devices														Index
S1	INT	Constant														
D1	DINT							D						R		√

■ Overview

Read the current actual speed.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

S2: Speed: the current actual speed of the axis.

10.7.9.7 MCRDPAR: Read parameter

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCRDPAR S1 S2 D1	**Step size**	**11**

Operand	Type	Applicable devices													Index
S1	INT	Constant													
S2	INT	Constant													
D1	DINT							D						R	√

■ Overview

Read parameter command.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

S2: Parameter No.: Please refer to the "Parameter No. List" below.

S3: Numerical output: parameter value output element, 32-bit data.

10.7.9.8 MCWRPAR: Write parameters

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCWRPAR S1 S2 S3	**Step size**	**11**

Operand	Type	Applicable devices													Index
S1	INT	Constant													
S2	INT	Constant													
S3	DINT							D						R	√

■ Overview

Write parameter command.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured on the host computer (the node number should be less than or equal to 31 when using axis control instructions).

S2: Parameter No.: Please refer to the "Parameter No. List" below.

S3: Value: new parameter value, 32-bit data.

■ List of parameter numbers

Parameter	Name	Type of data	Read/write	Description
K1000	Interrupt mode	Uint32	Read/write	Positioning interrupt mode 0 (default): execute a new positioning command when the power flow of the previous positioning command is disconnected, and the current positioning will be interrupted immediately (note: when the same positioning command is interrupted, the command that triggered the positioning last time must be used to start it. Break); 1: the axis cannot be interrupted during the positioning process, and the
K1001	Di input status	Uint32	Read	Di input status [31:16]: factory custom [15:3]: reserved [1]: positive limit 0: invalid 1:efficient [0]: reverse limit 0: invalid 1:efficient
K1010	Axis status	Int32	Read	Current axis state -1: not configured 0: disabled 1: ready (standstill) 2: stopping 3: homing 4: continue motion 5: positioning/discrete motion 15: error stop (error stop)

10.7.9.9 MCHOME: Home return

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCHOME S1 S2 D1 D2 D3	**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	DINT	Constant							D						R	√
D1	BOOL				M	S										√
D2	BOOL				M	S										√
D3	INT								D						R	√

■ Overview

Perform an automatic search for the origin.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 32 when using axis control instructions).

S2: Position: the target position after origin return.

D1: Complete: The origin return is completed.

D2: Busy: Origin return is being performed.

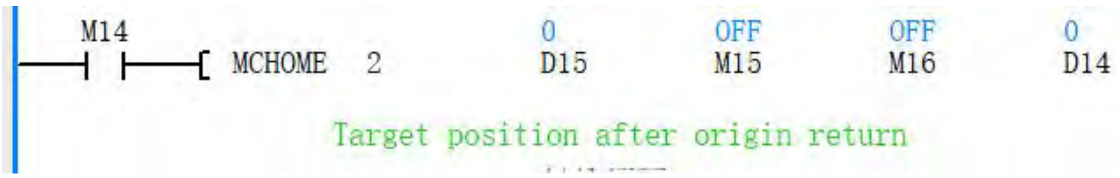
D3: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

MCHOME instruction CANOpen object operation steps

Step	Operation/condition	Description
1	6060h = 6	Switch to origin return mode
2	6061h = 6	Wait for the switch to origin return mode to complete
3	6040h.bit4 = 0	Reset control word
4	607ch = origin offset	Set the origin offset
5	6040h.bit4 = 1	Start return to origin
6	6041h.bit10 = 1 and 6041h.bit13 = 1	Return to origin failed
	6041h.bit10 = 1 and 6041h.bit12 = 1	Return to origin is successful

■ Program example:

Take our SD700 servo drive as an example for the slave station



Description:

1、 After the servo is enabled, the command will be executed when M14 is turned ON, and the origin return will be performed.

2、 When performing positioning, the instruction Sign M16 is set; after the positioning is completed, the instruction completion Sign M15 is set.

3、 When an error occurs during operation, the error will be stored in D14. For the error code, please refer to "10.7.10 Instruction Error Code Definition".

■ Note:

When using the origin return axis command, the origin return mode and speed need to be set in the object dictionary of the CANopen configuration interface. For the description of each origin return method, please refer to the manual of the servo/motor driver.

10.7.9.10 MCMOVREL: Relative positioning

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCMOVREL S1 S2 S3 S4 S5 D1 D2 D3	**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	DINT	Constant							D						R	√
S3	DINT	Constant							D						R	√
S4	DINT	Constant							D						R	√
S5	DINT	Constant							D						R	√

D1	BOOL				M	S										√
D2	BOOL				M	S										√
D3	INT								D						R	√

■ Overview

Relative positioning of the axis, the control axis continues to move the specified position (distance) at the current position.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 32 when using axis control instructions).

S2: Position: Specify the positioning target position.

S3: Speed: Specify the maximum speed for positioning.

S4: Acceleration: Specify the positioning acceleration.

S5: Deceleration: Specify the positioning deceleration.

D1: Complete: The positioning is completed, and the axis has moved to the specified position.

D2: Busy: Locating.

D3: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

Note: The relative positioning position is actually an incremental position, that is, the corresponding target position is added to the current position; when the relative positioning command is used to interrupt other positioning commands, the final position of the axis is the addition of the target positions of the two positioning commands.

MCMOVREL instruction CANOpen object operation steps

Step	Operation/condition	Description
1	6060h = 1	Switch to location mode
2	6061h = 1	Wait for the switch location mode to complete
3	6040h.bit4 = 0	Reset control word
4	607ah = position	Write (relative) target location
5	6083h = acceleration	Write acceleration
6	6084h = deceleration	Write deceleration
7	6081h = speed	Write positioning speed
9	6040h.bit4 = 1 6040h.bit5 = m 6040h.bit6 = 1 6040h.bit8 = 0 6040h.bit9 = 0	The control word is written to the corresponding mode. Interrupt mode (parameter number: k1000) = 0, then m = 1; otherwise, m = 0. Trigger positioning
10	607ah < 0 and 6041h.bit11 = 1 and 60fdh.bit0 = 1	When the negative movement meets the negative limit, the positioning ends.
	607ah > 0 and 6041h.bit11 = 1 and 60fdh.bit1 = 1	When the forward movement meets the positive limit, the positioning ends
	6041h.bit10 = 1	The target position is reached, the positioning is completed

■ Program example:

Take our SD700 servo drive as an example for the slave station

The slave station number is 1, the target position is 5000000, the speed is 1000r/min, and the acceleration and deceleration is 300.

Illustrate:

1、 After the servo is enabled, the command starts to be executed when M7 is turned ON, and the current position is used as the starting point to run the set distance.

2、 When performing positioning, the instruction Sign M109 is set; after the positioning is completed, the instruction completion Sign M108 is set.

3、 The speed unit is r/min, and the acceleration and deceleration time = acceleration and deceleration × (set speed/maximum motor speed). According to the above example, the acceleration and deceleration time is calculated as 50ms.

4、 When an error occurs during operation, the error will be stored in D32. For the error code, please refer to "10.7.10 Instruction Error Code Definition".

■ Note:

If the slave takes our SD710 servo drive as an example:

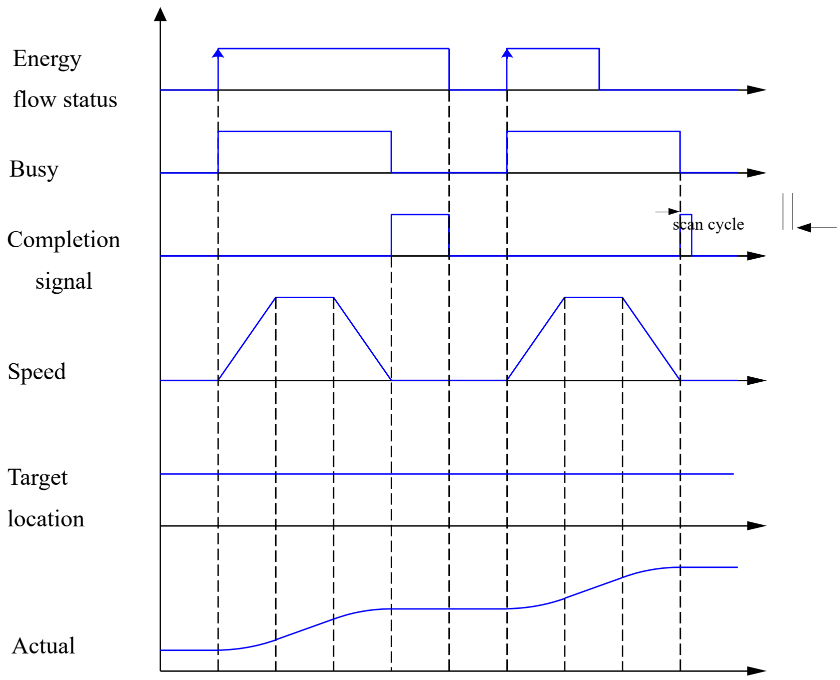
The slave station number is 2, the target position is 5000000, the speed is 100KHz, and the acceleration and deceleration is 1000000.

It should be noted here that the speed unit and acceleration/deceleration unit of SD700 and SD710 are different. The speed unit of SD700 is r/min, and the acceleration/deceleration unit is time unit; the speed unit of SD710 is frequency, and the acceleration/deceleration unit is pulse unit. So SD710 needs to convert the frequency into r/min, and then calculate the acceleration and deceleration time.

Acceleration and deceleration time: $Accelerationtime = \frac{(Setspeed/Pulserequiredforonerevolutionofthemotor)\times 60}{Maximummotorspeed} \times \frac{Maximummotorspeed}{(Setacceleration\wedge deceleration/Pulserequiredforonerevolutionofthemotor)\times 60}$

Calculate the acceleration and deceleration time according to the above example: $\frac{(100K/10K)\times 60}{6000} \times \frac{6000}{(1000K/10K)\times 60} = 100ms$

To sum up, when using our SD700 and SD710 servo drives, this place needs to be paid attention to.



MCMOVREL Timing Diagram

10.7.9.11 MCMOVABS: Absolute positioning

Ladder Diagram:	Applicable models	VC-S VC-P
	Affect the flag	
Command list: MCMOVABS S1 S2 S3 S4 S5 D1 D2 D3	Step size	11

Operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	DINT	Constant							D						R	√
S3	DINT	Constant							D						R	√
S4	DINT	Constant							D						R	√
S5	DINT	Constant							D						R	√
D1	BOOL				M	S										√
D2	BOOL				M	S										√
D3	INT								D						R	√

■ Overview

Absolute positioning, control the axis to move to the specified position.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

S2: Position: Specify the positioning target position.

S3: Speed: Specify the maximum speed for positioning.

S4: Acceleration: Specify the positioning acceleration.

S5: Deceleration: Specify the positioning deceleration.

D1: Complete: The positioning is completed, and the axis has moved to the specified position.

D2: Busy: Locating.

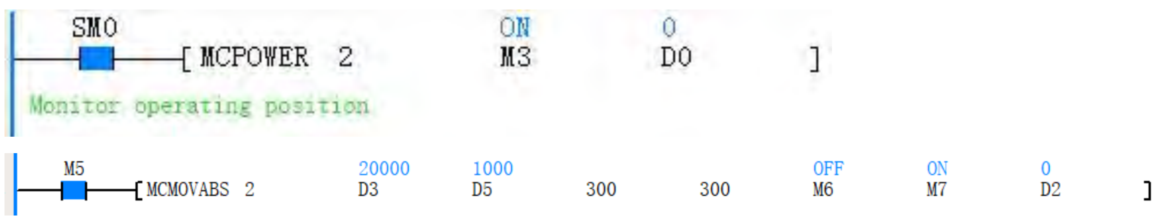
D3: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

MCMOVABS instruction CANOpen object operation steps

Step	Operation/condition	Description
1	6060h = 1	Switch to location mode
2	6061h = 1	Wait for the switch location mode to complete
3	6040h.bit4 = 0	Reset control word
4	607Ah = position	Write (absolute) target location
5	6083h = acceleration	Write acceleration
6	6084h = deceleration	Write deceleration
7	6081h = speed	Write positioning speed
9	6040h.bit4 = 1 6040h.bit5 = m 6040h.bit6 = 0 6040h.bit8 = 0 6040h.bit9 = 0	The control word is written to the corresponding mode. Interrupt mode (parameter number: K1000) = 0, then m = 1; otherwise, m = 0. Trigger positioning
10	607Ah < 0 and 6041h.bit11 = 1 and 60fdh.bit0 = 1	When the negative movement meets the negative limit, the positioning ends.
	607Ah > 0 and 6041h.bit11 = 1 and 60fdh.bit1 = 1	When the forward movement meets the positive limit, the positioning ends
	6041h.bit10 = 1	The target position is reached, the positioning is completed

■ Program example:

Take our SD700 servo drive as an example for the slave station



Illustrate:

4、 After the servo is enabled, the command starts to be executed when M5 is turned ON, and runs to the set end position.

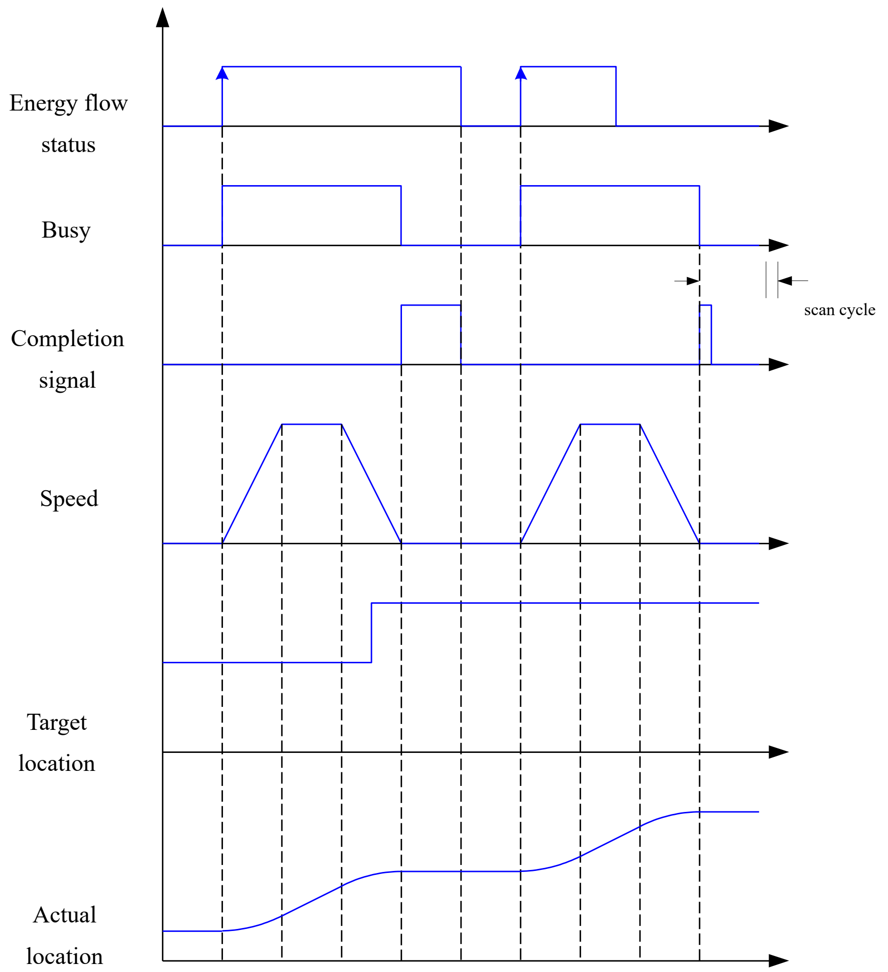
5、 When performing positioning, the instruction Flag bit M7 is set; after the positioning execution is completed, the instruction completion Flag bit M6 is set.

6、 The speed unit is r/min, and the acceleration and deceleration time = acceleration and deceleration × (set speed/maximum motor speed). According to the above example, the acceleration and deceleration time is calculated as 50ms.

7、 When an error occurs during operation, the error will be stored in D2. For the error code, please refer to "10.7.10 Instruction Error Code Definition".

■ Note:

Our SD700 and SD710 servo drives have different parameter units, so there are differences in calculating the acceleration and deceleration time. Attention is required. For details, please refer to "10.7.9.10 Note"



MCMOVABS Timing diagram

10.7.9.12 MCMOVVEL: Velocity mode

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCMOVVEL S1 S2 S3 S4 D1 D2 D3	**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	DINT	Constant							D						R	√
S3	DINT	Constant							D						R	√
S4	DINT	Constant							D						R	√
D1	BOOL				M	S										√
D2	BOOL				M	S										√
D3	INT								D						R	√

■ Overview

Speed mode, control the axis to move at the specified speed.

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

S2: Speed: Specify the movement speed.

S3: Acceleration: Specify the acceleration.

S4: Deceleration: Specify the deceleration.

D1: Speed reached: The speed Sign output specified by the command has been reached.

D2: Busy: The instruction is being executed.

D3: Error code: Please refer to "Instruction Error Code Definition in 10.7.10".

MCMOVVEL instruction CANOpen object operation steps

Step	Operation/condition	Description
1	6040h = 0x0f	Reset control word
2	6083h = acceleration	Write acceleration
3	6084h = deceleration	Write deceleration
4	6060h = 3	Switch to speed mode
5	6061h = 3	Wait for the switching speed mode to complete
6	60ffh = target speed	Set target speed
7	6041h.bit10 = 1	Target speed reached
8	60ffh < 0 and 6041h.bit11 = 1 and 60fdh. Bit0 = 1 Or 60ffh = 0	When the negative movement meets the negative limit, the movement ends
9	607ah > 6040h and 6041h.bit11 = 1 and 60fdh.bit1 = 1 Or 60ffh = 0	When the forward movement meets the positive limit, the movement ends
10	60ffh = 0	The command energy flow is invalid, and the movement ends

■ Program example:

Take our SD700 servo drive as an example for the slave station

Illustrate:

1、 After the servo is enabled, the command starts to be executed when M8 is turned ON, and runs at the set speed. The set speed can be changed during operation.

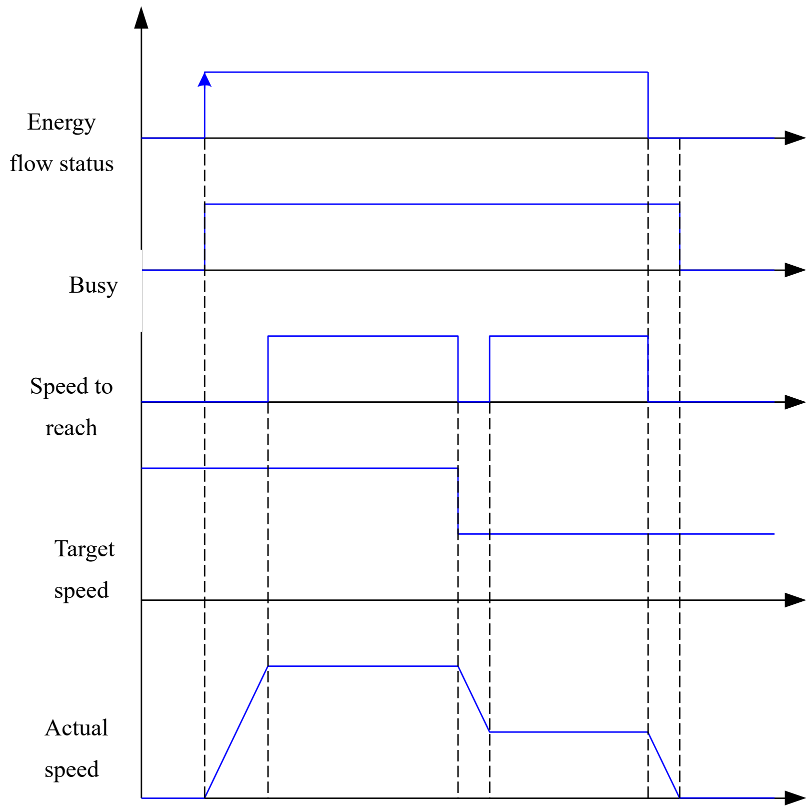
2、 When executing the command, the command Flag bit M10 is set; when the speed reaches the set value, the Flag bit M9 is set.

3、 The speed unit is r/min, and the acceleration and deceleration time = acceleration and deceleration × (set speed/maximum motor speed). According to the above example, the acceleration and deceleration time is calculated as 50ms.

4、 When an error occurs during operation, the error will be stored in D9. For the error code, please refer to "10.7.10 Instruction Error Code Definition".

■ Note:

Our SD700 and SD710 servo drives have different parameter units, so there are differences in calculating the acceleration and deceleration time. Attention is required. For details, please refer to "10.7.9.10 Note"



MCMOVVEL Timing Diagram

10.7.9.13 MCJOG: Jog

Ladder Diagram:	**Applicable models**	**VC-S VC-P**
	Affect the flag	
Command list: MCJOG S1 S2 S3 S4 S5 D1 D2 D3	**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	INT	Constant														
S2	BOOL				M	S										√
S3	BOOL				M	S										√
S4	DINT	Constant							D						R	√
S5	DINT	Constant							D						R	√
D1	BOOL				M	S										√
D2	INT								D						R	√

■ Overview

■ Instruction parameter description:

S1: Axis number: specify the number of the control axis, which corresponds to the node number of each slave station configured in the host computer (the node number should be less than or equal to 31 when using axis control instructions).

S2: Forward jog: Forward jog.

S3: Reverse jog: Reverse jog.

S4: Speed: Specify the speed.

S5: Acceleration/Deceleration: Specify the acceleration/deceleration.

D1: busy: the instruction is being executed.

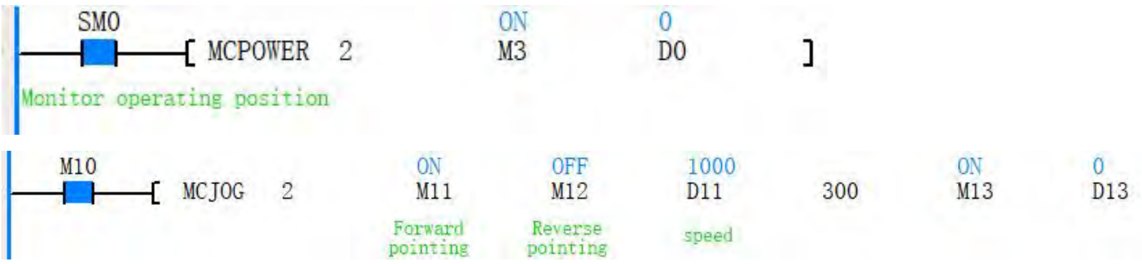
D2: Error code: Please refer to "Instruction Error Code Definition in 4".

MCJOG instruction CANOpen object operation steps

Step	Operation/condition	Description
1	6040h = 0x0f	Reset control word
2	6083h = acceleration/deceleration	Write acceleration
3	6084h = acceleration/deceleration	Write deceleration
4	6060h = 3	Switch to speed mode
5	6061h = 3	Wait for the switching speed mode to
6	Jog forward: 60ffh = target speed Jog reverse: 60ffh = - target speed Other: 60ffh = 0	Forward and reverse jog
7	60ffh < 0 and 6041h.bit11 = 1 and 60fdh.bit0 = 1 Or 60ffh = 0	When the negative movement meets the negative limit, the jog ends
8	607ah > 0 and 6041h.bit11 = 1 and 60fdh.bit1 = 1 Or 60ffh = 0	The forward movement meets the positive limit, and the jog ends
9	60ffh = 0	The command energy flow is invalid, and the

■ Program example:

Take our SD700 servo drive as an example for the slave station



Illustrate:

1、 After the servo is enabled, the command will be executed when M10 is turned ON, forward jog when M11 is turned ON, reverse jog when M12 is turned ON, run at the set speed, and the set speed can be changed during operation.

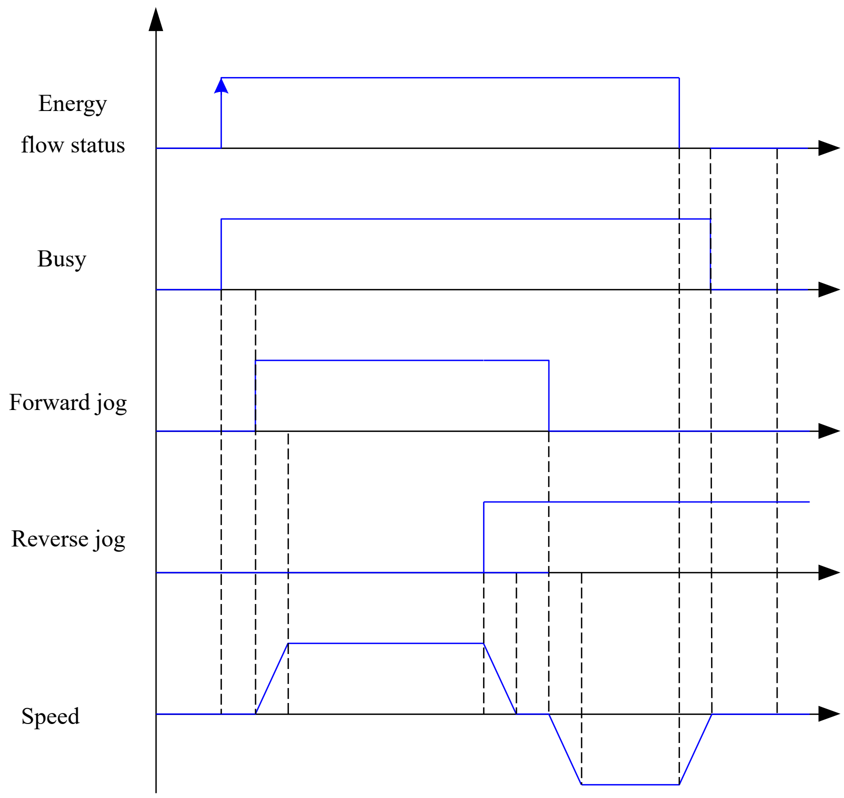
2、 When the instruction is executed, the instruction Flag bit M13 is set.

3、 The speed unit is r/min, and the acceleration and deceleration time = acceleration and deceleration × (set speed/maximum motor speed). According to the above example, the acceleration and deceleration time is calculated as 50ms.

4、 When an error occurs during operation, the error will be stored in D13. For the error code, please refer to "10.7.10 Instruction Error Code Definition".

■ Note:

Our SD700 and SD710 servo drives have different parameter units, so there are differences in calculating the acceleration and deceleration time. Attention is required. For details, please refer to "10.7.9.10 Note"



MCJOG timing diagram

10.7.10 Instruction Error Code Definition

Code	Description
0	No errors.
1	Wrong axis number. The axis number does not exist in the canopen configuration or the pdo configuration is incorrect.
2	Command parameter error. Mcmovabs, mcmovrel, mcmovvel, mcjog command acceleration/deceleration is less than or equal to 0; mcmovabs, mcmovrel command speed is less than or equal to 0;
3	The value of the command parameter (position, origin position offset) is out of range. ※1
4	The command parameter (speed) value is out of range. ※2
5	The command parameter (acceleration) value is out of range. ※2
6	The command parameter (deceleration) value is out
8	The current instruction is interrupted by other instructions during the execution process, the enable is lost, or the connection is dropped, resulting in the instruction not being completed and the execution being stopped.
9	Forward overtravel prevents the instruction from completing and stops execution. ※3
10	Reverse overtravel prevents the instruction from completing and stops execution. ※3
11	Return to origin failed.
16	The axis is not enabled and the current command cannot be executed.
17	If it is not in "fault stop" state, the mcreset instruction cannot be executed.
18	The axis is in the "stop" state, and the current command cannot be executed.
19	The axis is returning to the origin, and the current command cannot be executed.
20	The axis is moving continuously, and the current command cannot be executed.

21	The axis is being positioned and the current command cannot be executed.
31	The axis is in the "fault stop" state and the current command cannot be executed. ※3
33	The axis is still in the "stop" state or the drive is disconnected during the execution of the command, and the current command cannot be executed*4
250	Axis enable timeout.
251	Servo/motor driver error. ※3
255	Servo/motor drive dropped. ※3

※1 The value cannot exceed the 32-bit integer range.

※2 The value cannot exceed 30000.

※3 Overtravel during motion, the axis will enter the "fault stop" state, and the axis can only be triggered to move in the opposite direction after reset by the MCRESET instruction.

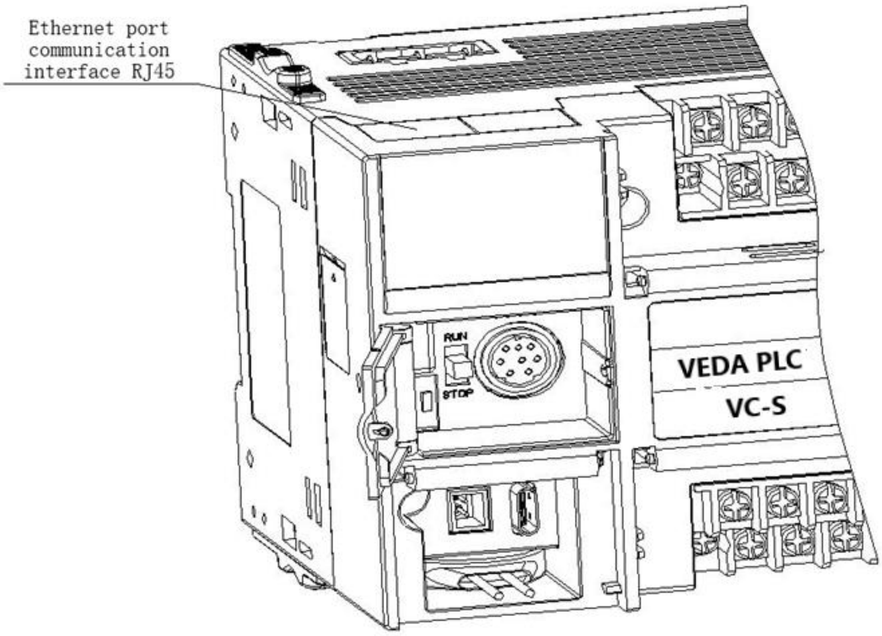
※4 Need to turn on the power flow againcommand will be executed.

10.8 Ethernet Communication Settings

VC-S series main module comes with Ethernet communication interface, supports 10M/100M adaptive rate, supports Modbus TCP function, VC-S supports 16 connections (the same connection with the same IP and port number) for data exchange, the same site can be used as Master and Slave.

The Ethernet sending and receiving frames are processed in each user program scan cycle, and the read and write speed is affected by the user program scan cycle.

10.8.1 Hardware interface

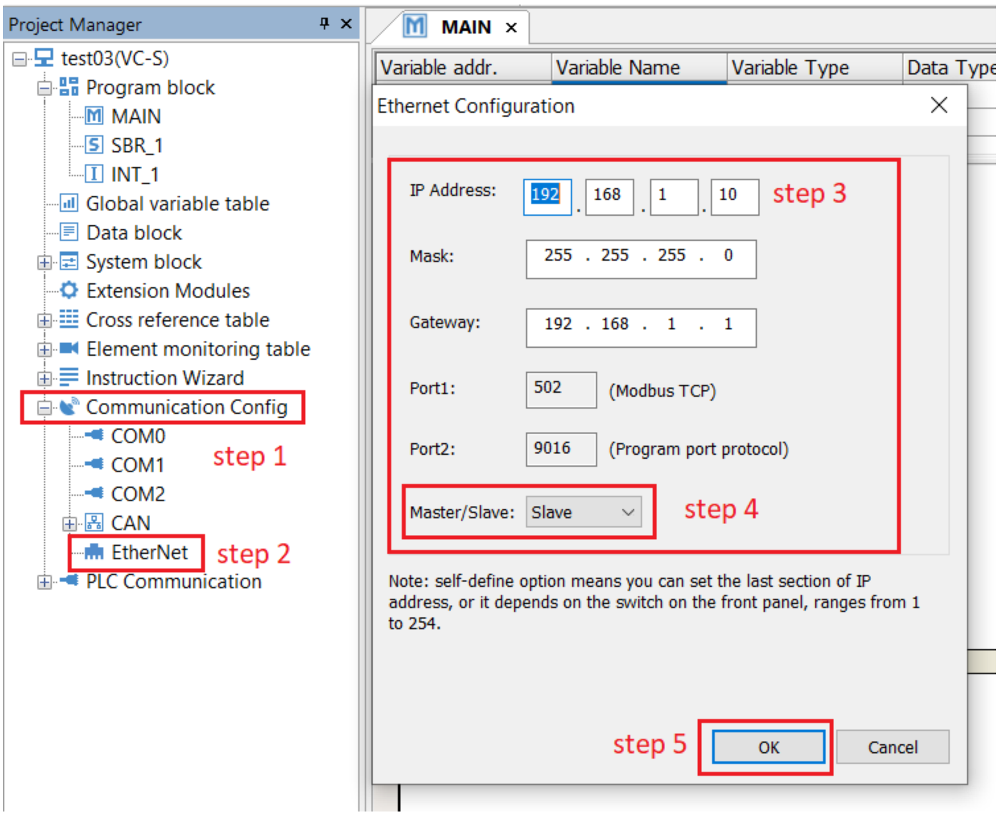


Ethernet Indicator Description

Silkscreen name	Name	Function
Eth	Communication indicator (light green)	Blinking: data transmission Off: no data transmission

10.8.2 Ethernet master/slave configuration

Select the **communication config** in the "Connect", double-click "EtherNet", the pop-up window is as follows:



① The default IP for communication in VC-S is 192.168.1.10, which is the default IP after the PLC is formatted at the factory, which can directly communicate with the host computer and the Modbus TCP client.

② 【IP address】: The identification of the device's identity in network communication, the uniqueness of each device's IP address must be ensured. Otherwise, the device will not be able to access the network.

③ 【Subnet mask】: Address multiple physical networks under the same network address. The mask is used to divide the subnet address and the device address of the host ID. The way to get the subnet address is to reserve the bits in the IP address that correspond to the positions of the mask containing 1's, and replace the other bits with 0's. If no special requirements are required, the subnet mask is 255.255.255.0;

④ 【Gateway Address】: The message can be routed to the device that is not in the current network. If there is no gateway, the gateway address is 0.0.0.0

⑤ 【Port No. 1】: The listening of TCP port 502 is reserved for Modbus TCP communication. Not set.

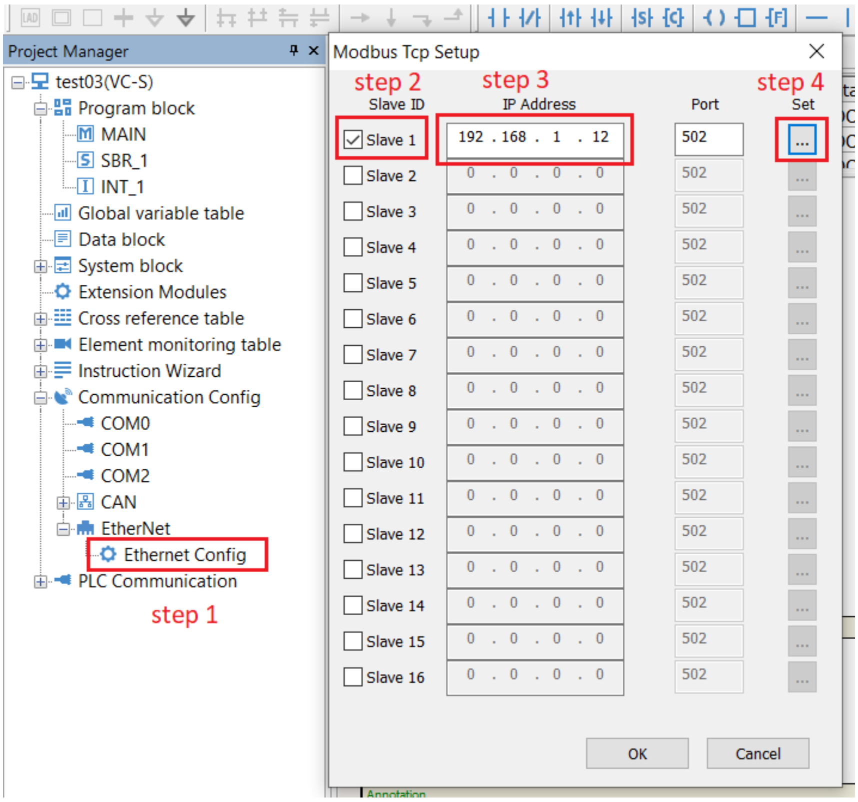
⑥ 【Port No. 2】: Port 9016 is used to communicate with the host computer VEDA PCT. Not settable

⑦ 【Master/Slave】: Set master mode or slave mode;

A multi-master multi-slave network can be constructed using N: N. The meanings of "master" and "slave" here are: "master" is a PLC that can write its own M and D elements and can read M and D elements of other stations; "slave" can only read into other stations PLC of M and D components. Under the set max number of sites (the number of stations is also restricted by the refresh mode), the PLC with the station number smaller than the number of stations can be used as the "master", while the PLC with the station number greater than the number of stations can only be used as the "slave". ". The slave station can only read the relevant M and D elements of the master station. These M and D elements have a corresponding relationship with each master station according to the refresh mode in the master station. You can refer to the N: N shared M and D element table. The slave station There are no corresponding M and D elements in these tables.

10.8.3 Ethernet Modbus TCP protocol

A. When VC-S is used as the master station, in addition to setting the IP address on the body, it is also necessary to configure some information of the device to be accessed: such as IP address, data, length, etc., need to be configured in the interface. Right-click "EtherNet" to add "Ethernet Configuration", and double-click "EtherNet Configuration" to pop up the Modbus TCP configuration table. As shown below

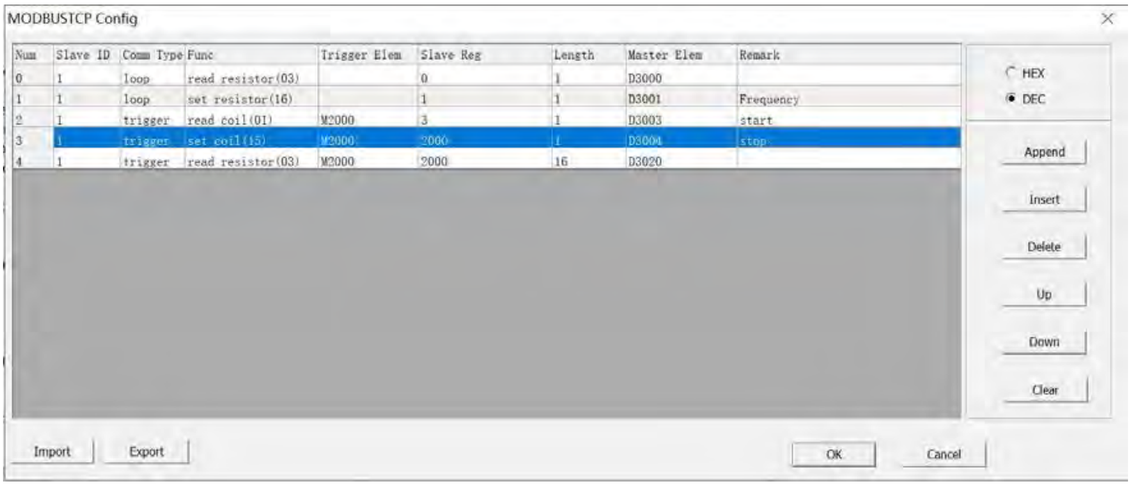


① Right-click "EtherNet" to add "Ethernet Configuration", then double-click "Ethernet Configuration" to pop up the Modbus TCP configuration table;

② Check Slave 1, it means enable, allow to write IP address, if you add multiple slaves, you can check it in turn.

③ Manually enter the IP address of the device. The port number 502 remains the default;

④ Click [**Settings**] to pop up the Modbus TCP configuration table. As shown below;



1) 【Slave station ID】 No need to set; reserved

2) 【Communication type】: cyclic mode and trigger mode; cyclic mode: indicates cyclic access to the slave station; trigger mode: it needs to be used with the trigger element in the trigger condition.

When the device is ON, the slave station is accessed, and it is automatically OFF after the access is completed.

3) 【Function】: read coil, write coil, read register, write register.

4) 【Trigger Condition】: Support M setting element;

5) 【Slave register address】: The address of the coil or register to be accessed. (decimal or hexadecimal)

6) 【Data length】The data length to be accessed. If accessing slave M10-M20, it is 11 components, so fill in 11.

7) 【Master station buffer address】The starting address of the master station buffer. The configuration of number 4 in the figure above means that the values in the 16 elements starting from D500 of the local machine are written into the 16 registers starting from the address of the slave device 2000. The configuration of No. 5 in the figure above means that the local machine reads the 16 register values starting from the address of the slave device 2000 and stores them in the 16 registers starting from D600 of the local machine.

8) 【Remark】 Comment description.

9) Modbus TCP configuration has a maximum of 128 configurations.

Function	Quantity
Read register	123
Write register	121
Read coil	1968
Write coil	1936

B. Modbus TCP function codes supported by VC-S

Function code	Function	Data length
0x01	Read coil	>=1
0x02	Read coil	>=1
0x03	Read register	>=1
0x04	Read register	>=1
0x05	Write a single coil	=1
0x06	Write a single register	=1
0x0f	Write multiple coils	>1
0x10	Write multiple registers	>1

C. Modbus TCP communication address

When VC-S is used as a Modbus TCP communication slave, the Modbus TCP address corresponding to the device is shown in the following table

Element	Type	Physical element	Protocol address	Supported function codes	Notes
Y	Bit element	Y0~y777 (octal code) a total of 512 points	0000～0511	01, 05, 15	The status of the output, the component numbers are y0~Y7, y10~y17
X	Bit element	X0～x777 (octal code) a total of 512 points	1200～01711	01, 05, 15 02	The state of the input, supports two kinds of addresses, the component number is the same as above
M	Bit element	M0～m2047 M2048～m10239	2000～4047 12000-20191	01, 05, 15	
Sm	Bit element	Sm0～sm255 Sm256～sm1023	4400～4655 30000-30767	01, 05, 15	
S	Bit element	S0～s1023 S1024～s4095	6000-7023 31000-34071	01, 05, 15	
T	Bit element	T0～t255 T256～t511	8000～8255 11000-11255	01, 05, 15	The state of the T element
C	Bit element	C0～c255 C256～c511	9200～9455 10000-10255	01, 05, 15	The state of the C element
D	Word element	D0～d7999	0000～7999	03, 06, 16	
Sd	Word element	Sd0～sd255 Sd256～sd1023	8000～8255 12000-12767	03, 06, 16	
Z	Word element	Z0～z15	8500～8515	03, 06, 16	
T	Word element	T0～t255 T256～t511	9000～9255 11000-11255	03, 06, 16	Current value of T element
C	Word element	C0～c199	9500～9699	03, 06, 16	Current value of C element (int)
C	Double word element	C200～c255	9700～9811	03, 16	Current value of C element (dint)
C	Double word element	C256～c263	10000-10101	03, 16	Current value of C element (dint)
R	Word element	R0～r32767	13000-45767	03, 06, 16	

10.8.4 Ethernet connection failure detection

- Is the network connection normal?
  If the network is unstable, it may be caused by interference or poor contact. Please use a shielded network cable and redo the crystal head. It can be tested by the ping command that comes with the computer.
- Whether the IP address setting is correct, check SD470~SD473;
- If a gateway is used, whether the gateway address is set correctly;
- The Modbus TCP configuration table sets whether the address of the slave is correct, pay attention to the selected hexadecimal format: hexadecimal or decimal.
- If the IP addresses of two different network segments (the first three segments of the IP addresses are different) want to communicate, a device with routing function needs to be added to connect.

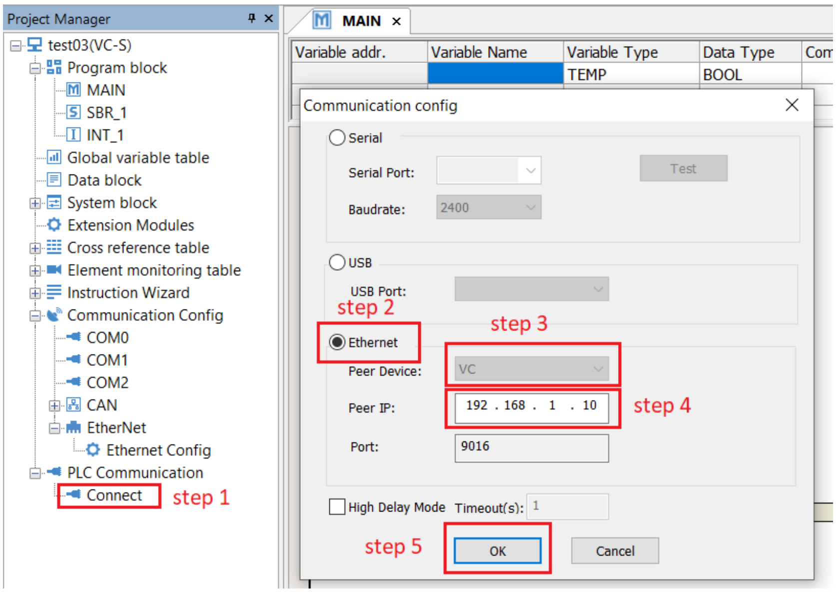
10.8.5 Ethernet Special SD Register

Address	Actions and Functions	R/W			VC-S		
SD470	IP address 0	R			√		

Address	Actions and Functions	R/W			VC-S		
SD471	IP address 1	R			√		
SD472	IP address 2	R			√		
SD473	IP address 3	R			√		
SD474	Ethernet slave listening port	R			√		
SD475	MAC address 0	R			√		
SD476	MAC address 1	R			√		
SD477	MAC address 2	R			√		
SD478	MAC address 3	R			√		
SD479	MAC address 4	R			√		
SD480	MAC address 5	R			√		
SD481	Displays the slave IP3 address number of the communication error	R			√		

10.8.6 Ethernet download and monitoring

A. VC-S program download and monitoring can be set through the network port as shown in the figure below:



① Double-click【Project Manager】-【PLC Communication】under【Connect】to pop up the connection setting interface;

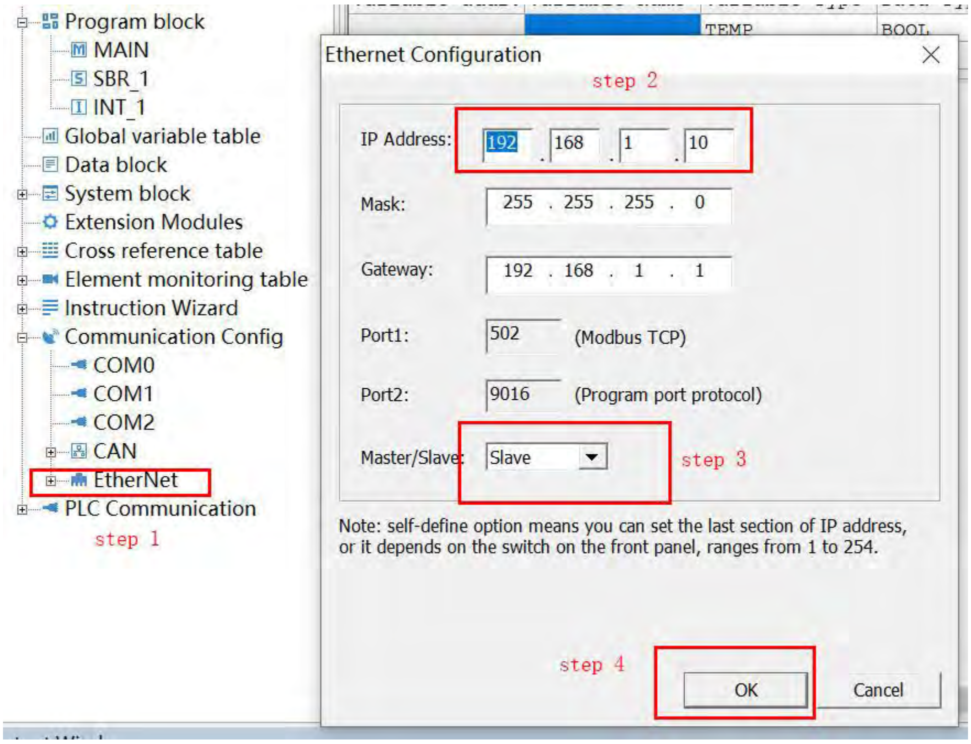
② Select【Ethernet】communication mode;

③【Peer device type】indicates the device to be connected; here select the VC main module;

④【Peer IP address】indicates the IP address of the connected device. VC-S series PLC factory IP address setting value is 192.168.1.10;

⑤【Port number】 9016 port, cannot be changed by default. After completing the above configuration, click [Confirm] to complete, and the interface will pop up a message prompt box for whether the connection is successful.

B. Local IP address setting



① Double-click "EtherNet" to pop up the Ethernet configuration information window;

② Allow to modify the IP address of the machine;

③ Set to slave mode.

④ After completing the configuration, click "Confirm" to compile and download the program to complete the setting of the local IP address.

C. Precautions:

① Before communication, set the first three IP addresses of the PC, which belong to the same network segment as the VC-S local IP address;

② The last segment of the PC IP address and the last segment of the VC-S host IP address need to be set differently.

Chapter 11 Positioning Commands and Interpolation

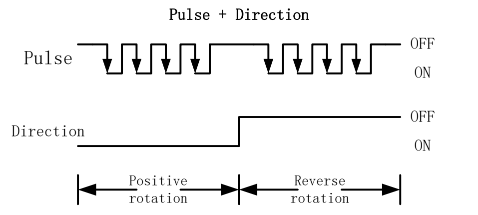
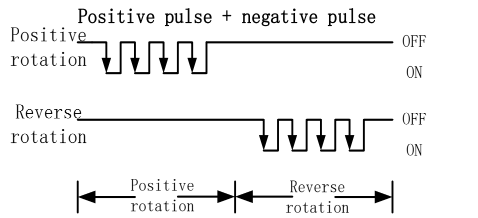
11.1 VEDA VC series PLC Positioning Function Overview

VEDA VC series PLC supports positioning function: including pulse output positioning function, two-axis linear and arc trajectory interpolation, electronic cam, and inter-axis synchronous motion control function, which can be widely used in positioning control system for stepping of various brands and servo drive for control.

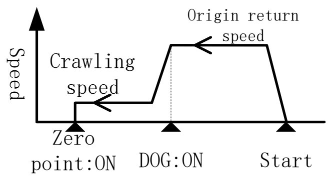
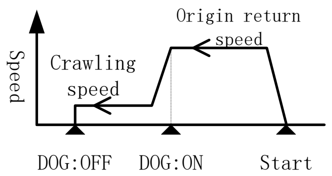
11.1.1 VEDA VC series PLC positioning function introduction

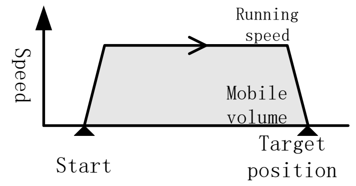
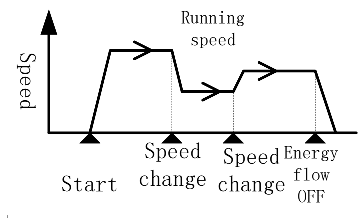
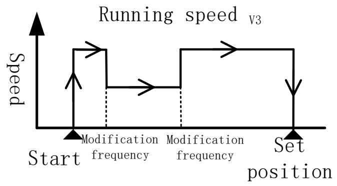
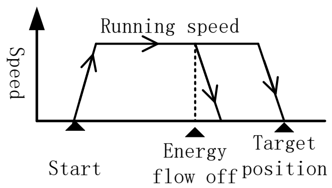
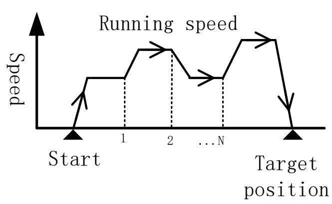
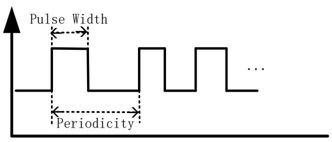
Function	VC-B	VC-S	VC-SM	VC-P
Number of control axes	3 axes (Y0~Y2)	8 axes (Y0~Y7)	8 axes (Y0~Y7)	Reserve
Maximum output frequency	100khz	200khz	200khz	
Pulse output method	Open collector	Open collector	Open collector	
Pulse output form	Pulse + direction	Pulse + direction	Pulse + direction	
Trapezoidal acceleration and deceleration	Support	Support	Support	
S-shaped acceleration and deceleration	Not support	Support	Support	
Electronic cam	Not support	Not support	Support	
Sync function	Not support	Not support	Support	
Two-axis linear interpolation, circular interpolation	Not support	Not support	Support	
4-axis flying shear/flying shear	Not support	Not support	Support	

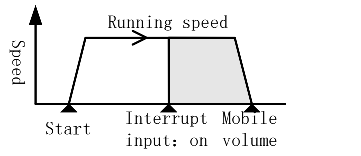
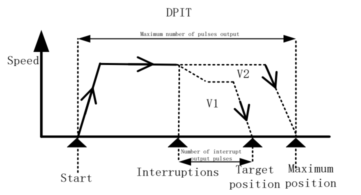
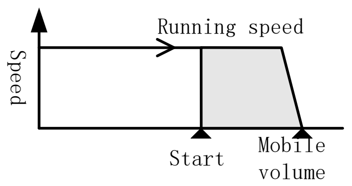
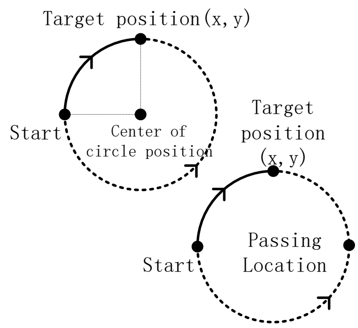
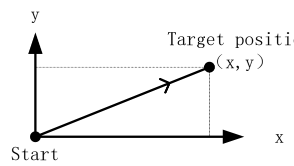
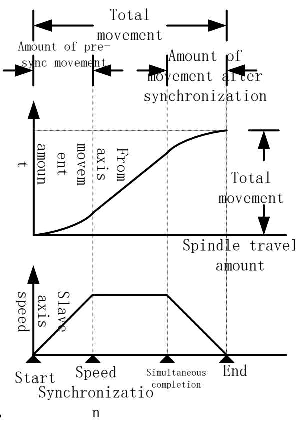
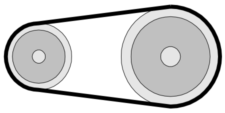
(2) Definition of pulse output form

	Pulse + direction	Positive and Negative Pulse (CW/CCW)
Pulse output method		
*Note: Positive and negative pulse output modes are supported only in interpolation commands, electronic cams, and electronic gears		

(3) VEDA VC series PLC supports positioning instruction table Different VEDA VC series PLC supports different positioning instructions, as shown in the following table;

Command name	Movement track	Function	VC-B	VC-S	VC-SM	VC-P
DSZR		It operates at the specified return-to-origin speed and can automatically search for the DOG signal. When DOG is detected (DOG sensor is ON), it will decelerate to creep speed. When there is a Zero flagal input, it stops and the origin return is completed.	√	√	√	
ZRN		It operates at the specified return-to-origin speed. When DOG is detected (DOG sensor is ON), it will decelerate to creep speed. When the DOG sensor is OFF, it stops and the origin return is completed.	√	√	√	

Command name	Movement track	Function	VC-B	VC-S	VC-SM	VC-P
DRVI		Act according to the set running speed, stop at the target position, and the position adopts relative coordinates.	√	√	√	
DRVA		Move according to the set running speed, stop at the target position, and the position adopts absolute coordinates.	√	√	√	
PLSV		It operates at the set running speed. If the running speed changes, run at the new speed; if the power flow becomes invalid, the pulse output stops. When there is an acceleration/deceleration operation, the acceleration/deceleration is performed when the speed is changed.	√	√	√	
PLSY		According to the set frequency, there is no acceleration/deceleration running speed action. If the speed changes during the running process, it will run at the new speed. If the energy flow is invalid, the pulse will stop immediately, and there will be no acceleration/deceleration function. The instruction has no direction output control and needs to be added by user programming.	√	√	√	
PLSR/ DPLSR		According to the set acceleration and deceleration time, the set frequency, and the set number of pulses, if the power flow is cut off during operation, it will decelerate and stop the operation according to the deceleration time.	√	√	√	
PLS		According to the set multi-stage position and frequency, run different frequencies in different position sections to realize multi-stage speed operation, and it is not allowed to change the speed during operation. Support acceleration and deceleration settings. (N represents the number of segments)	√	√	√	
PWM		Output modulated square wave according to the set pulse width and period	√	√	√	

Command name	Movement track	Function	VC-B	VC-S	VC-SM	VC-P
DVIT		It operates at the set running speed. If the interrupt input is ON, it will run for the specified number of pulses and then decelerate and stop.	√	√	√	
DPTI		It operates at the set running speed. If the interrupt input is ON, it will run for the specified number of pulses and then decelerate and stop. If no interrupt signal is detected, it will output pulses according to the set maximum number of pulses and then stop.	⊠	√	√	
STOPDV		When a positioning operation is being executed, if this command is started, it will decelerate and stop after running the specified number of pulses.	⊠	√	√	
CW		According to the specified linear speed, move clockwise along the arc trajectory to the target position.	⊠	⊠	√	
CCW		According to the specified linear velocity, it moves to the target position along the arc track in the counterclockwise direction.	⊠	⊠	√	
LIN		Move to the target position along a linear trajectory at the specified vector speed.	⊠	⊠	√	
Electronic cam		The slave axis follows the movement of the master axis, keeps synchronization with the speed of the master axis within the specified position range, and supports acceleration and deceleration control during the transition process before and after synchronization.	⊠	⊠	√	
Electronic gear		According to a certain electronic gear ratio, the slave axis is controlled to follow the master axis.	⊠	⊠	√	
Note: Hit "√" to indicate that the series supports, hit "√"⊠" means not supported						

11.1.2 Description of special devices for positioning commands

For the high-speed output axis of the positioning command, it is necessary to set reasonable parameters of the corresponding axis such as the maximum speed, the base speed, and the acceleration and deceleration time before the pulse can be output. Mainly set by SM element and SD element.

High-speed output involves special soft components; VC-B series supports 3 axes (Y0~Y2); VC-S series supports 8 axes (Y0~Y7);

(1) Special SM components are defined as follows:

Axis number								Describe
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Functional properties
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410	Pulse output stop control bit
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411	Monitor bit in pulse output
SM272	SM292	SM312	SM332	SM352	SM372	SM392	SM412	Pwm instruction cycle unit switching is valid
SM273	SM293	SM313	SM333	SM353	SM373	SM393	SM413	Plsy interrupt drive pulse output valid
SM274	SM294	SM314	SM334	SM354	SM374	SM394	SM414	Pls multi-speed command cycle execution is valid
SM275	SM295	SM315	SM335	SM355	SM375	SM395	SM415	The gradual change of pslv command frequency is valid
SM276	SM296	SM316	SM336	SM356	SM376	SM396	SM416	Dszr/zrn instruction clear function is valid
SM277	SM297	SM317	SM337	SM357	SM377	SM397	SM417	Dszr instruction clear signal specified element is valid
SM278	SM298	SM318	SM338	SM358	SM378	SM398	SM418	Dszr instruction origin return direction specification is valid
SM279	SM299	SM319	SM339	SM359	SM379	SM399	SM419	Forward limit
SM280	SM300	SM320	SM340	SM360	SM380	SM400	SM420	Inversion limit
SM281	SM301	SM321	SM341	SM361	SM381	SM401	SM421	Logic inversion of near-point signal is valid
SM282	SM302	SM322	SM342	SM362	SM382	SM402	SM422	The logic inversion of the zero flagal is valid
SM283	SM303	SM323	SM343	SM363	SM383	SM403	SM423	The logic inversion of the interrupt signal is valid
SM284	SM304	SM324	SM344	SM364	SM384	SM404	SM424	Interrupt input function specification is valid
SM285	SM305	SM325	SM345	SM365	SM385	SM405	SM425	User interrupt input command
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426	S-type acceleration and deceleration are valid
SM287	SM307	SM327	SM347	SM367	SM387	SM407	SM427	Dvit interrupt signal masking is valid

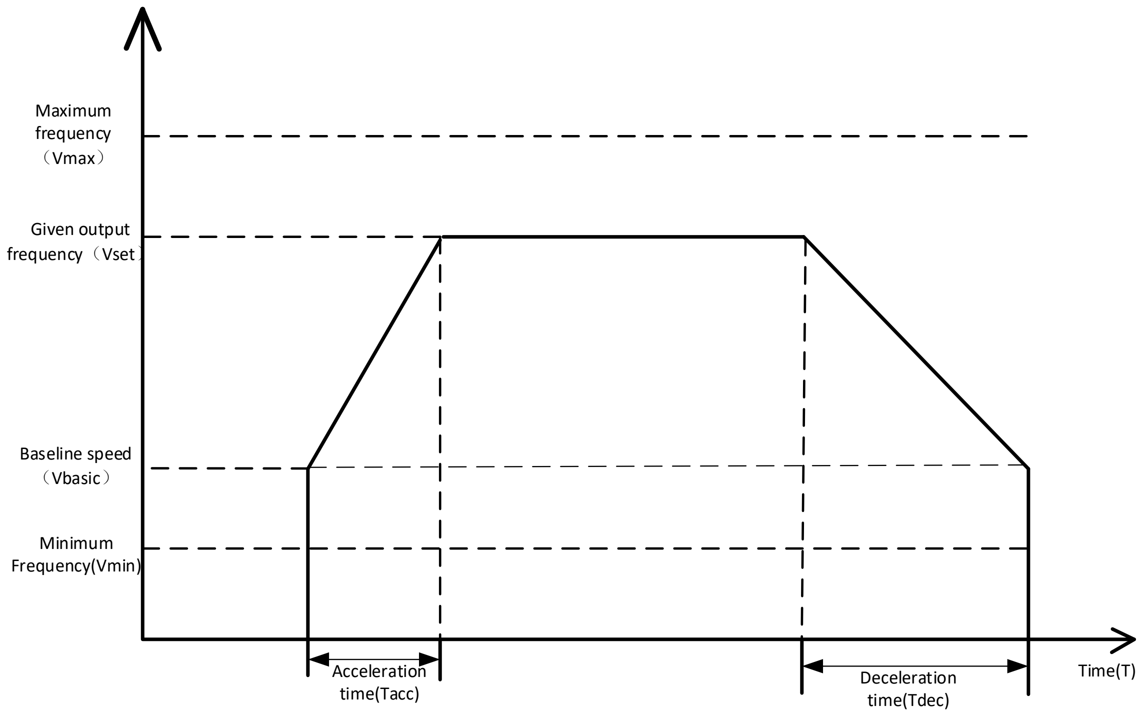
(2) The definition of special SD components is as follows:

Axis number								Function description
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
SD160	SD180	SD200	SD220	SD240	SD260	SD280	SD300	Pulse output cumulative number (32 bits)
SD161	SD181	SD201	SD221	SD241	SD261	SD281	SD301	
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302	Positioning command current position (32 bits)
SD163	SD183	SD203	SD223	SD243	SD263	SD283	SD303	
SD164	SD184	SD204	SD224	SD244	SD264	SD284	SD304	Current frequency of positioning command (32 bits)
SD165	SD185	SD205	SD225	SD245	SD265	SD285	SD305	
SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306	Maximum speed unit Hz; (32 bits)
SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307	Default VC-B: 100kHz; VC-S: 200kHz
SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308	Base speed: unit Hz (maximum speed 1/10)

								Default 800
SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309	Acceleration time unit ms (10~5000ms) default 100ms
SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD310	Deceleration time unit ms (10~5000ms) Default 100ms
SD171	SD191	SD211	SD231	SD251	SD271	SD291	SD311	DSZR instruction creep speed setting unit Hz Default 1000
SD172	SD192	SD212	SD232	SD252	SD272	SD292	SD312	DSZR command origin return speed unit Hz Default 5000 (32-bit)
SD173	SD193	SD213	SD233	SD253	SD273	SD293	SD313	
SD174	SD194	SD214	SD234	SD254	SD274	SD294	SD314	The number of currently executed segments of the PLS instruction
SD175	SD195	SD215	SD235	SD255	SD275	SD295	SD315	DSZR/ZRN instruction clear signal designation
SD176	SD196	SD216	SD236	SD256	SD276	SD296	SD316	DVIT designation interrupt signal device designation

11.1.3 Description of output frequency and acceleration/deceleration time

The output frequency relationship is shown in the following figure



Illustrate:

Vmax: the highest frequency; (Hz) is generally set by the SD special register;

Vset: Pulse output frequency set by the user, set by the command;

Vbias: The base output frequency set by the user, generally set by the SD special register;

Vmin: minimum frequency, calculated

Tacc: acceleration time; the time required to accelerate from the base speed Vbias to the maximum speed Vmax;

Tdec: Deceleration time, default Tdec= TacC, generally set by SD register

In a general case, Vmax≥Vset, Vbias≥Vmin, if the above conditions are not met, the frequency will be adjusted. Vmax and Vmin determine the pulse

The upper and lower limits of the pulse output frequency.

Vmin: The lowest frequency value Vmin that can actually be output

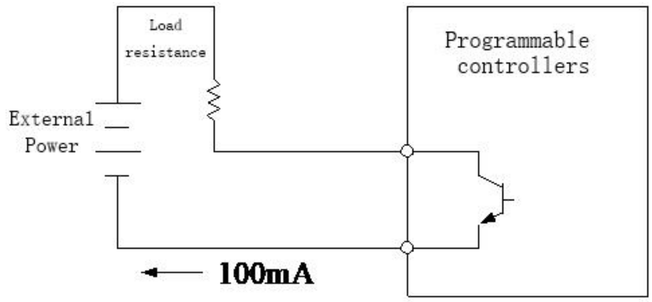
11.1.4 Notes on using positioning instructions

1) When the positioning command or high-speed command is running effectively (including output completion), other operations on the same port are invalid. Only when the high-speed pulse output command is invalid, other commands have correct output.
2) When there are multiple positioning commands or high-speed commands on the same port, the first valid command occupies the output end, and the later valid command does not occupy the output end.

A. Transistor output

1. A VEDA VC series PLC with transistor output must be used.

2. When the PLC performs high-speed pulse output, the load current specified by the PLC output transistor described below must be used.



B.Requirements for positioning instructions in programming

Positioning instructions can be used repeatedly in the program, but need to pay attention:

(1) Other positioning or high-speed pulse output commands using the same high-speed pulse output point cannot be driven at the same time. A high-speed pulse output point can only be driven by one positioning command (or high-speed command) at any time.

(2) When the power flow of a positioning instruction is disconnected, the power flow must be turned on after one or more PLC scan cycles before it can be driven again.

Points for Simultaneous Use of High-speed Commands and Positioning Commands

(3) In terms of functional realization, it is recommended to use positioning instructions to replace these high-speed pulse output instructions (PLSY, PLSR, PLS), which can complete the automatic update of absolute position SD components.

(4) Absolute position SD element can be used to store and update the current absolute position after the positioning instruction is used. The automatic increase or decrease of the SD element value of the absolute position is determined according to the cumulative SD element change value of the output pulse, plus the running direction when the positioning command is called, so the two are in a linkage relationship. Please do not write the pulse accumulation SD element when using the positioning command, otherwise the absolute position SD element data may be confused.

(5) If the positioning command and other high-speed pulse output commands (PLSY, PLSR, PLS) must be used at the same time, the PLC program needs to be written so that the data in the absolute position SD element of the absolute position register can be updated correctly.

C. Limitations of the actual output frequency of the positioning command

When the positioning instruction is executed, the minimum frequency of the actual output pulse is limited by the following formula:

$$F_{min\_acc} = \sqrt{\frac{F_{max} \times 500}{T}}$$

In the above formula, F_{max} Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds.Calculation results F_{min_acc} is the minimum output frequency limit value.

If the output frequency specified in the positioning command is F, the following three cases are the actual output frequency.

① F is less than the base frequency or F is greater than F_{max} The highest frequency, with no actual output.

② F is less than F_{min_acc}, the actual output is F_{min_acc}.

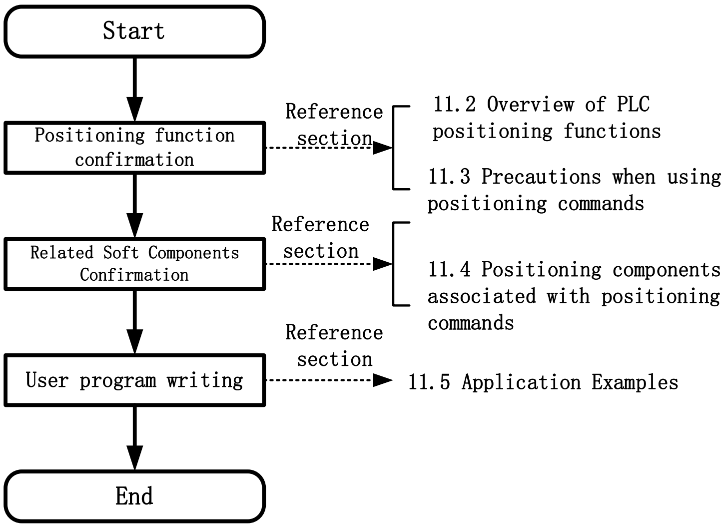
③ F is greater than or equal to F_{min_acc}, and less than or equal to F_{max}, the output is F.

11.2 Positioning Command

(1) Pulse commands are generally divided into speed commands and positioning commands, and different commands need to be used according to different occasions. The pulse command classification table is shown in the following figure:

Instruction Type	Command name	Reference chapter
Positioning command	ZRN origin return command	☒ For details, please refer to Chapter 11 11.2.1
	DSZR with DOG search origin return command	☒ For details, please refer to Chapter 11 11.2.2
	DRVI relative position control command	☒ For details, please refer to Chapter 11 11.2.3
	DRVA absolute position control instruction	☒ For details, please refer to Chapter 11 11.2.4
	PLSR 16-bit variable speed pulse output command with acceleration and deceleration	☒ For details, please refer to Chapter 11 11.2.5
	DPLSR 32-bit variable speed pulse output command with acceleration and deceleration	☒ For details, please refer to Chapter 11 11.2.6
	PLS multi-speed pulse output command	☒ For details, please refer to Chapter 11 11.2.7
	DVIT interrupt fixed length	☒ For details, please refer to Chapter 11 11.2.8
	DPTI maximum fixed-length interrupt positioning instruction	☒ For details, please refer to Chapter 11 11.2.9
	STOPDV pulse output stop command	☒ For details, please refer to Chapter 11 11.2.10
High speed command	PLSY pulse output	☒ For details, please refer to Chapter 11 11.3.1
	PLSV variable speed pulse output	☒ For details, please refer to Chapter 11 11.3.2
	PWM pulse width modulation command	☒ For details, please refer to Chapter 11 11.3.3
	HTOUCH read position capture instruction	☒ For details, please refer to Chapter 11 11.3.4
Interpolation command	LIN linear trajectory interpolation	☒ For details, please refer to Chapter 11 11.4.1
	CW clockwise arc path interpolation	☒ For details, please refer to Chapter 11 11.4.2
	CCW counterclockwise arc trajectory interpolation	☒ For details, please refer to Chapter 11 11.4.3

(2) Steps for using positioning commands and high-speed commands:



11.2.1 ZRN: Origin return command

Ladder Diagram: ZRN *(S1)* *(S2)* *(S3)* *(D)*	**Applicable models** **VC-B VC-S VC-P**
	Affect the flag
Command list: ZRN ***(S1)*** ***(S2)*** ***(S3)*** **(*D*)**	**Step size** **11**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S3	BOOL		X	Y	M	S										
D	BOOL			Y												

- **Operand Description**

S1: Origin return speed. Specify the speed at which the origin return starts.

Range VC-B: 10~100000Hz; VC-S: 10~200000Hz; VC-P: 10~200000Hz;

S2: Creeping speed. Specify a relatively low speed after the near-point signal (DOG) turns ON. Range: 10~32767Hz;

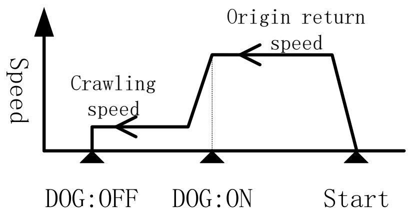
S3: Near-point signal. Specify the near-point signal input X element.

When a device other than the input relay (X) is specified, the offset of the origin position will increase due to the influence of the PLC operation cycle.

D: High-speed pulse output start address. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7;

- **Function Description**

Function description: After the instruction is executed, accelerate to the origin return speed with the set acceleration time, make the actuator move to the origin (DOG), detect the DOG signal, decelerate to the creeping speed, the DOG signal is OFF, and stop the pulse output. As shown below



1) The current pulse position can monitor the special register, see the table below

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

2) The "pulse output stop Sign" can check the pulse output status, the Flag bit is set to ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

3) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
top speed (Default: 100KHz or 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

4) The default is T-shaped acceleration and deceleration. When the SM special auxiliary relay is set to ON, the S-shaped acceleration and deceleration are used for startup. See the table below;

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Remark	Invalid modification during command execution						

5) The ZRN command is a speed control command, so there is no pulse output completion interrupt;

6) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

7) Clear signal output is valid

By setting the SM special element, the output clearing function is valid.

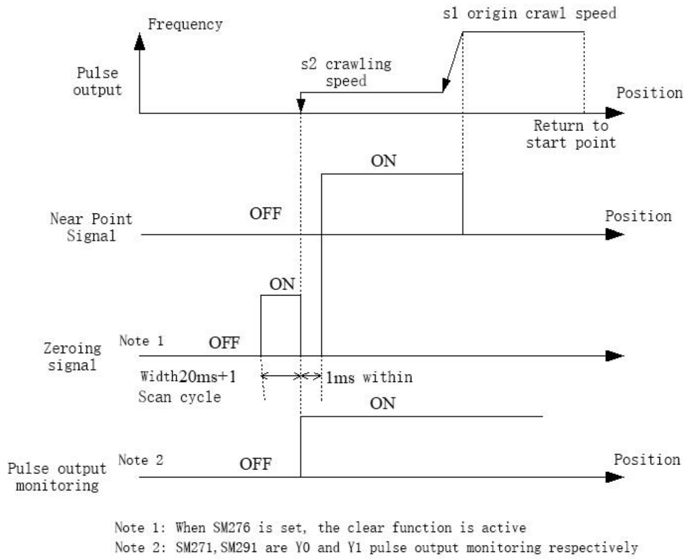
The specified clear signal is valid							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM276	SM296	SM316	SM336	SM356	SM376	SM396	SM416
Default Y10	Default Y11	Default Y12	Default Y13	Default Y14	Default Y15	Default Y16	Default Y17
Take Y0 as an example: when SM276 is ON, Y10 will be output as a clear signal when the ZRN instruction home return is completed.							

7) Specify the output of the clear signal, which can be set by setting the SM special element "clear signal designation is valid", the clear signal can be specified through the SD special register, only the Y port, see the following table

Clear signal is valid							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM277	SM297	SM317	SM337	SM357	SM377	SM397	SM417
SD175	SD195	SD215	SD235	SD255	SD275	SD295	SD315
Take Y0 as an example: when the designation of SM277 clearing signal is valid, Y3 specified by SD175=3 is the clearing signal output port, (the clearing device number is in decimal, such as: 8 means specifying Y10 and so on) When it is valid at the same time as the designated clearing signal Flag bit, the designated clearing signal is valid.							

8) Program demonstration (take Y0 as an example)

Program description: When M0 is ON, Y0 sends pulses at a frequency of 10000Hz to make the actuator run in the direction of the origin. When the DOG signal changes from OFF to ON, the pulse output frequency switches to 2000Hz and runs at a crawling speed. When the DOG signal changes from ON to OFF When Y0 stops pulse output immediately, and clears the current position register SD162/SD613. In addition, if the set clear signal is valid, the clear signal will be output at the same time. The monitoring Flag bit SM271 during pulse output changes from ON to OFF, and the origin return is completed. The timing diagram is as follows;



- **Precautions**

1. Since the return-to-origin command ZRN does not have the function of automatically searching for the near-point signal, the return-to-origin operation must be performed from farther than the front end of the near-point detection device.

2. During the origin return process, the value of the current value register will move in the decreasing direction.

3. The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{min\_acc} = \sqrt{\frac{F_{max} \times 500}{T}}$$

In the above formula, F_{max} Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results F_{min_acc} is the minimum output frequency limit value.

4. For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result.If the maximum speed is lower than the above calculation result, there will be no pulse output.

5. The creep speed should be greater than zero and less than one tenth of the top speed.

11.2.2 DSZR: Origin return command with DOG search

Ladder Diagram:	Applicable models	VC-B VC-S VC-P
├─┤ ├─┤[DSZR *(S1)* *(S2)* *(D1)* *(D2)*]	Affect the flag	
Instruction list: DSZR *(S1)* *(S2)* (D1) (D2)	Step size	9

Operand	Type	Applicable devices													Index
S1	BOOL		X	Y	M	S									
S2	BOOL		X												
D1	BOOL			Y											
D2	BOOL			Y	M	S									

- **Operand Description**

S1: Specify the device number of the input near-point signal (DOG). When the input device is specified, the offset of the origin position will increase due to the influence of the operation cycle of the programmable controller. The timeliness of the specified signal at point X is the best;

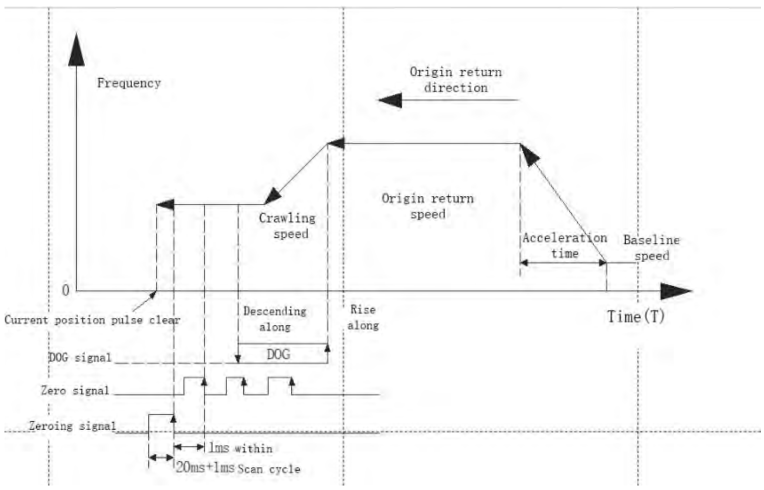
S2: Specify the device number of the input Zero flagal. Range: X0～X7.

D1: Specify the pulse port for outputting pulses. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

D2: Specify the rotation direction signal output port. ON: Forward rotation (current value of pulse output increases); OFF: Reverse rotation (current value of pulse output decreases)

- **Function Description**

Instruction function: After the instruction is executed, Start to output the frequency at the origin regression speed set in the SD special register, so that the moving mechanism moves to the near point (DOG) according to the set action sequence. When the DOG signal is detected, decelerate to the crawl speed, and the zero point signal is detected, stop the output immediately. As shown below



1) The current pulse position can monitor the special register; see the following table:

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

The current position SD register will be cleared after the origin regression is completed.

2) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

The stop Flag bit will be automatically OFF when the zero point return is completed..

3) Support T-type acceleration and deceleration, the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
top speed (Default: 100KHz or 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

4) Origin return speed and creep speed settings Applicable DSZR instruction.

Origin return speed and creep speed settings							
Creep speed (16-bit default: 1000Hz)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD171	SD191	SD211	SD231	SD251	SD271	SD291	SD311
Origin return speed (32-bit default: 5000Hz)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD172	SD192	SD212	SD232	SD252	SD272	SD292	SD312
SD173	SD193	SD213	SD233	SD253	SD272	SD293	SD313

5) Origin return direction designation, applicable to DSZR instruction, When SM Component of each axis is set to ON it means that the direction of origin return is the forward direction; OFF, it means that the direction of origin return is the reverse direction; see the following table:

Return-to-origin direction specification							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM278	SM298	SM318	SM338	SM358	SM378	SM398	SM418

6) Forward limit and reverse limit settings

when SM element is ON, it means reaching the limit of the forward rotation direction; OFF, it means that the forward rotation limit is not reached;

Forward and reverse limit settings							
Forward limit position (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM279	SM299	SM319	SM339	SM359	SM379	SM399	SM419
Reverse limit position (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM280	SM300	SM320	SM340	SM360	SM380	SM400	SM420

7) The origin signal and DOG signal are logically negated.

When SM element is set to ON, process signal according to negative logic, that is, when the input signal is OFF, process it according to ON signal; When the SM component is OFF, process the signal according to the positive logic, that is, when the input signal is ON, process it according to the ON signal..

Origin signal and DOG signal are logically inverted							
Near-point signal logic inversion (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7

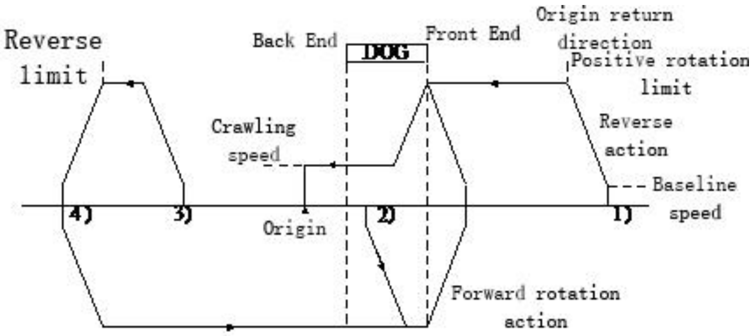
SM281	SM301	SM321	SM341	SM361	SM381	SM401	SM421
Origin signal logic inversion (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM282	SM302	SM322	SM342	SM362	SM382	SM402	SM422

8) After the origin return is completed, the output clearing signal function is valid or the specified output clearing signal function is valid. See the table below:

Output clear signal function setting							
The output clear signal function is valid (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM276	SM296	SM316	SM336	SM356	SM376	SM396	SM416
Default Y10	Default Y11	Default Y12	Default Y13	Default Y14	Default Y15	Default Y16	Default Y17
Take Y0 as an example: when SM276 is ON, after the DSZR instruction origin return is completed, Y10 output maintains the ON signal for 20ms+1 scan period.							
The specified output clear signal function is valid (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM277	SM297	SM317	SM337	SM357	SM377	SM397	SM417
SD175	SD195	SD215	SD235	SD255	SD275	SD295	SD315
Take Y0 as an example: When SM277 is ON, SD175=5, after the DSZR command origin return is completed, Y5 output maintains the ON signal for 20ms+1 scan period. The clear signal specifies that the SD value type of the soft element is decimal, for example, 8 means Y10 and so on.							

9) DOG search function

It is designed with forward limit and reverse limit. The operation of the origin return is different depending on the position of the origin return. As shown in the figure below, 1-4 represent the four cases of returning to the origin



a). When the start position is before passing the DOG:

1) The origin return operation is started by executing the origin return command.

2) Start moving in the return-to-origin direction at the return-to-origin speed.

3) Once the front end of the DOG is detected, it starts to decelerate to the creeping speed.

4) After detecting the rear end of the DOG, it stops when the first Zero flagal is detected.

b). When the start position is within the DOG:

1) The origin return operation is started by executing the origin return command.

2) At the return-to-origin speed, start moving in the direction opposite to the return-to-origin direction.

3) It decelerates and stops after detecting the front end of the DOG. (leaving DOG)

4) At the return-to-origin speed, start moving in the return-to-origin direction. (enter DOG again)

5) Once the front end of the DOG is detected, it starts to decelerate to the creeping speed.

6) After detecting the rear end of the DOG, it stops when the first Zero flagal is detected.

c). When the starting position is at the near-point signal OFF (after passing the DOG):

1) The origin return operation is started by executing the origin return command.

2) Start moving in the return-to-origin direction at the return-to-origin speed.

3) It decelerates and stops when the reverse rotation limit is detected.

4) Start moving in the opposite direction of the origin return at the origin return speed.

5) When the front end of the DOG is detected, it decelerates to a stop (detects (leaves) the DOG).

6) At the return-to-origin speed, start moving in the return-to-origin direction.

7) Once the front end of the DOG is detected, it starts to decelerate to the creeping speed.

8) After detecting the rear end of the DOG, it stops when the first Zero flagal is detected.

d). When the start position is at the limit switch position in the return-to-origin direction (reverse rotation limit is ON):

1) The origin return operation is started by the origin return command.

2) At the return-to-origin speed, start moving in the direction opposite to the return-to-origin direction.

3) It decelerates and stops after detecting the front end of the DOG. (check out (leave) DOG)

4) At the return-to-origin speed, start moving in the return-to-origin direction. (Enter DOG again.)

5) Once the front end of the DOG is detected, it starts to decelerate to the creeping speed.

6) After detecting the rear end of the DOG, it stops when the first Zero flagal is detected.

e). Note: When planning the near-point signal (DOG), it is necessary to consider the time when the signal is ON to decelerate to the creeping speed. Please use the creeping speed as low as possible, because it stops without decelerating, and the creeping speed is too fast, which may cause position deviation .

10) The DSZR instruction is a speed control instruction, and there is no pulse output completion interrupt.

11)The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds.Calculation results $F_{\min\_acc}$ is the minimum output frequency limit value.

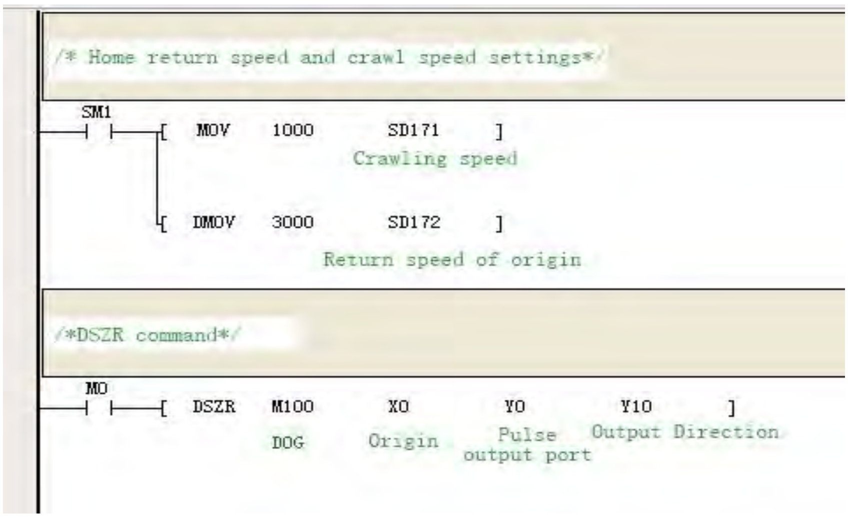
11) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

12) Program demonstration: (take Y0 as an example)

Parameters such as maximum speed, base speed, acceleration/deceleration time, origin return speed, and creeping speed can use default values, or can be reset by assigning soft components.



Program description: When M0 is ON, set the origin return speed to 3000Hz, and execute the origin return command with DOG search. When M100 is changed from OFF to ON, the speed switches to a creeping speed of 1000Hz. When the X0 signal is detected, it stops sending pulses immediately. and clear the current position SD register at the same time.

- **Precautions**

1) Only the PLC with transistor output can use this instruction;

2) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, the command will not be driven again.

3) High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

4). For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output.The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result.If the maximum speed is lower than the above calculation result, there will be no pulse output.

5). The creep speed should be greater than zero and less than one tenth of the top speed.

11.2.3 DRVI: Relative Position Control Instruction

Ladder Diagram: [DRVI *(S1)* *(S2)* *(D1)* *(D2)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction list: DRVI *(S1)* *(S2)* (D1) (D2)	**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D1	BOOL			Y												
D2	BOOL			Y	M	S										

- **Operand Description**

S1: Number of output pulses:

Range -2147483648~2147483647. The negative sign indicates the opposite direction.

S2: Output pulse frequency: 32-bit command,

VC-B range: 10~100000 (Hz)

VC-S range: 10 ~ 200000 (Hz).

D1: High-speed pulse output port. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

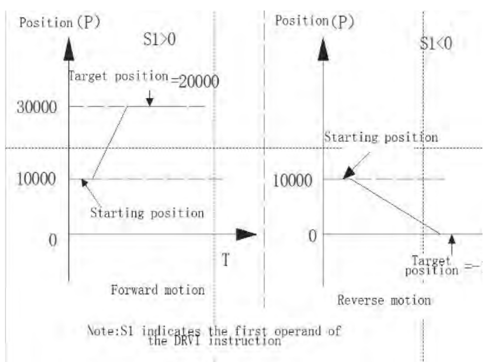
D2: Rotation direction signal output port or variable.

S1Positive: D2 is ON to indicate forward running;

S1Negative: D2 is OFF to indicate reverse operation;

- **Function Description**

Command function: send the set number of pulses with the set output port, the specified pulse frequency and direction. Movement based on relative position. As shown below:



1) The current pulse position can monitor the special register; see the following table:

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

2) The "pulse output stop Sign" can check the pulse output status, the Flag bit is set to ON in the pulse output, and the output is automatically turned OFF;

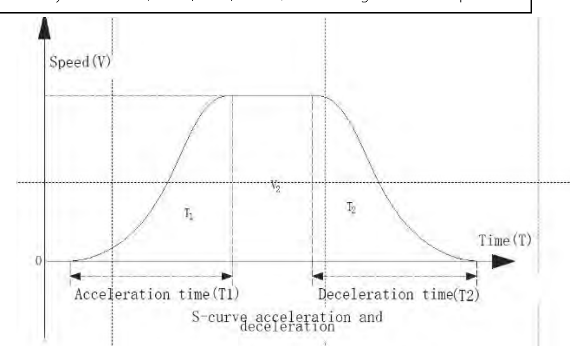
Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

3) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
top speed (Default: 100KHz or 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

4) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below;

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modify invalid DRVI, DRVA, PLSR, DPLSR, PLSV during instruction operation							



5) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

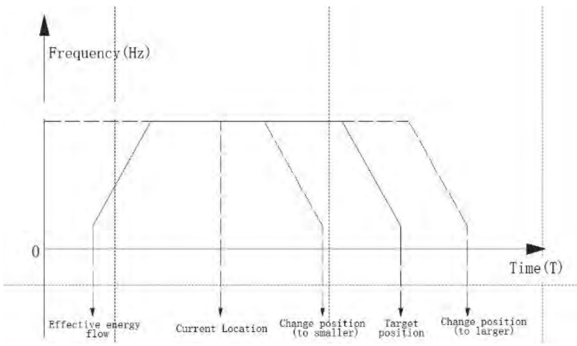
$$F_{min\_acc} = \sqrt{\frac{F_{max} \times 500}{T}}$$

In the above formula, F_{max} Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results F_{min_acc} is the minimum output frequency limit value.

For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. If the maximum speed is lower than the above calculation result, there will be no pulse output.

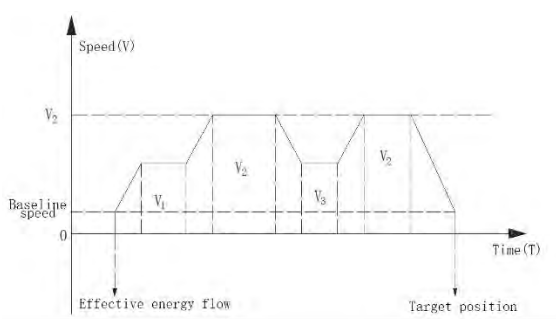
6) During the execution process, it is allowed to change the number of pulse outputs (may be large or small)

There is no need to set SM special auxiliary components. It should be noted that the changed position must be larger than the current pulse position. As shown below:



(Note: During the operation of the instruction, it is not allowed to modify the number of pulses across positive and negative, because the positive and negative pulse values represent the direction bit)

7) During the execution of the instruction, it is allowed to change the pulse running frequency. There is no need to set SM special auxiliary components. Note: It is not allowed to modify the running frequency during the acceleration and deceleration process. If the modified frequency is large but the number of target pulses is not enough, it will automatically decelerate to complete the positioning.



8) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

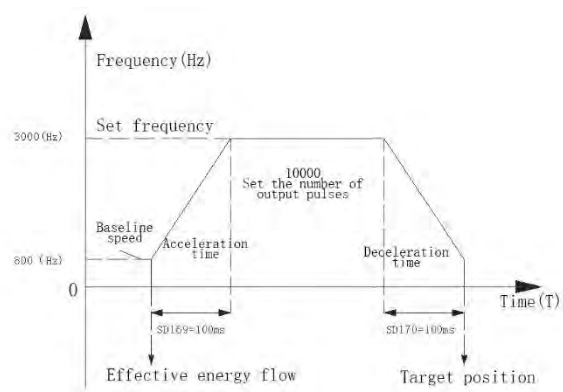
9) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

10) Program demonstration (take Y0 as an example)

Program description: When M0 is ON, the Y0 port outputs 100,000 pulses at a frequency of 3KHz, and Y10 outputs the direction bit to make the external servo or stepping mechanism run. The running sequence diagram is shown in the following figure:



- **Precautions**

1. Only PLC with transistor output can use this command

2. After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, it will not accept the command to drive again.

3. High-speed commands, envelope commands, and positioning commands can use the Y port to output high-speed pulses. Be careful not to

use these instructions for high-speed pulse output on the same high-speed port at the same time.

11.2.4 DRVA: Absolute position control command

Ladder Diagram: ├─┤ ├─[DRVA *(S1)* *(S2)* *(D1)* *(D2)*]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Command list: DRVA *(S1)* *(S2)* (D1) *(D2)*	**Step size**	**11**

Operand	Type	Applicable devices													Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D1	BOOL			Y												
D2	BOOL			Y	M	S										

- **Operand Description**

S1: specify target position (absolute position specification)

The range is -2147483648~2147483647; the negative sign indicates the opposite direction.

S2: Output pulse frequency (Hz)

VC-B range: 10~100000 (Hz);

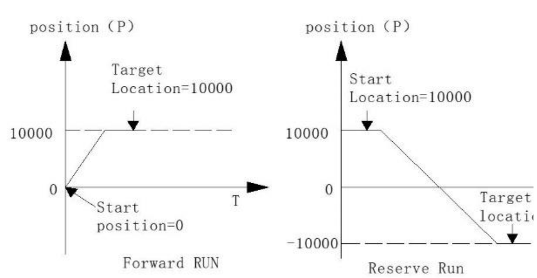
VC-S range: 10～200000(Hz);

D1: High-speed pulse output port designation. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7;

D2: Running direction signal output port or bit variable. According to the difference between S1 and the current position, the output is ON, which means forward running, otherwise it is reverse running.

- **Function Description**

Command function: send the set number of pulses with the set output port, the specified pulse frequency and direction. Movement based on absolute position. As shown below



1） The current pulse position can monitor the special register; see the following table:

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

2) The "pulse output stop Sign" can check the pulse output status, the Flag bit is set to ON in the pulse output, and the output is automatically turned OFF;

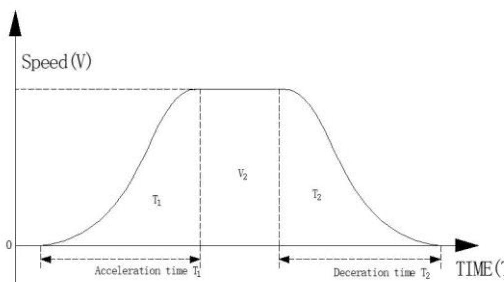
Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

3) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Top speed (Default: 100khz or 200khz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
Basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
Acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
Deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

4) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modification during command operation is invalid; S-type acceleration and deceleration are applicable to DRVI, DRVA, PLSR, DPLSR, PLSV and other commands							



S curve acceleration and deceleration

5) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

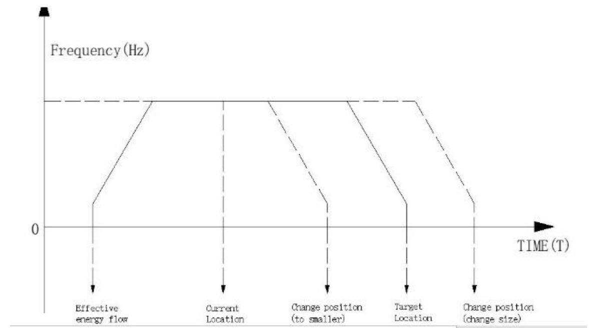
In the above formula, $F_{\max}$ Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results $F_{\min\_acc}$ is the minimum output frequency limit value.

For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. Initial acceleration and final deceleration

Part of the frequency can not be lower than the above calculation results If the maximum speed is lower than the above calculation results, there will be no pulse output.

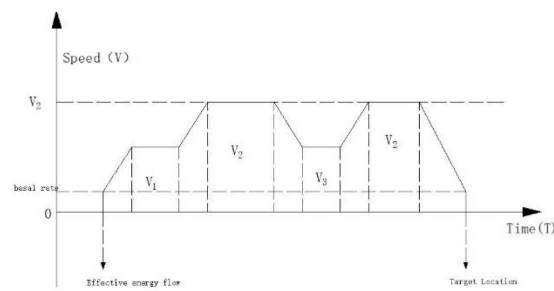
6) During the execution process, it is allowed to change the number of pulse outputs (may be large or small)

There is no need to set SM special auxiliary components. It should be noted that the changed position must be larger than the current pulse position. As shown below



(Note: During the operation of the instruction, it is not allowed to modify the number of pulses across positive and negative, because the positive and negative pulse values represent the direction bit)

7) During the execution of the instruction, it is allowed to change the pulse running frequency. There is no need to set SM special auxiliary components. Note: It is not allowed to modify the running frequency during acceleration and deceleration. If the modified frequency is large, but the number of target pulses is not enough, it will automatically decelerate to complete the positioning.



8) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

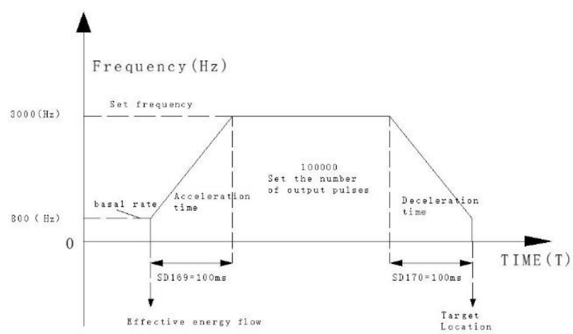
9) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

10) Program demonstration (take Y0 as an example)

Program description: When M0 is ON, the Y0 port outputs 10,000 pulses at a frequency of 3KHz, and Y10 outputs the direction position, so that the external servo or stepping mechanism runs from the designated origin to the target position.



- **Precautions**

1) Only the PLC with transistor output can use this instruction;

2) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, the command will not be driven again.

3) High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

11.2.5 PLSR: 16-bit counting pulse output command with acceleration and deceleration

Ladder Diagram: ├─┤ ├─[PLSR *(S1)* *(S2)* *(S3)* *(D)*	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction List: PLSR *(S1)* *(S2)* *(S3)* **(D)**	**Step size**	**10**

Operand	Type	Applicable devices														Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	BOOL			Y												

- **Operand Description**

S1: Output frequency (unit: Hz). Settable range: 10～32767

S2: Set the number of pulse outputs. The setting range is: 12～2147483647.

S3: Acceleration and deceleration time (unit: ms) Settable range: 10~32767 (ms) The default acceleration time is the same as the deceleration time, please pay attention when setting.

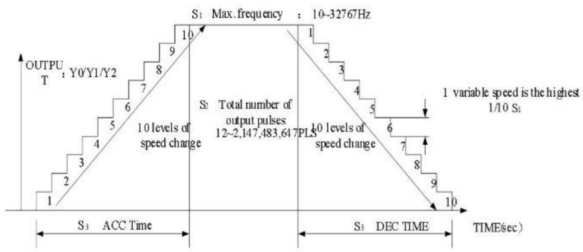
D: High-speed pulse output port designation. VC-B can specify Y0/Y1/Y2;

VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7;

- **Function Description**

With the set acceleration/deceleration time and the specified pulse frequency, the set number of pulses is output. as shown below:

Acceleration and deceleration time setting special register								
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Attributes
SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306	Top speed (default: 100khz or 200khz)
SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307	
SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308	Basal velocity (default: 800hz)
SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309	Acceleration time (default 100ms)
SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300	Deceleration time (default 100ms)
Note: when using the plsr instruction, the effective value of the acceleration and deceleration time is the value set by the instruction s3; the acceleration and deceleration value set by the sd element is invalid.								
Note: 1. Modification during command operation is invalid; 2. VC-B does not support S-type acceleration and deceleration.								



1) The current pulse position can monitor the special register; see the following table:

Pulse output count cumulative SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD160	SD180	SD200	SD220	SD240	SD260	SD280	SD300
SD161	SD181	SD201	SD221	SD241	SD261	SD281	SD301

2) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

3) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

4) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below;

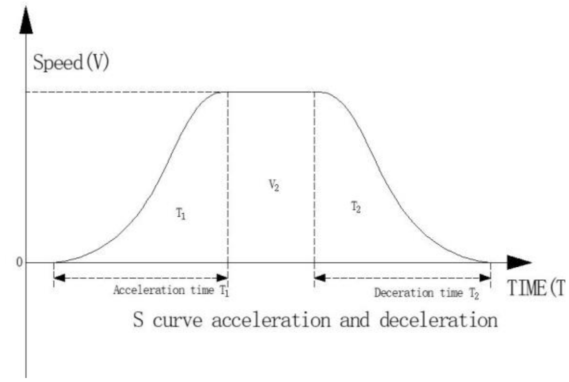
Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: 1. Modification during command operation is invalid; 2. VC-B does not support S-type acceleration and deceleration.							

5) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results $F_{\min\_acc}$ is the minimum output frequency limit value.

For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of



6) Modification of frequency and number of pulses during operation is not supported.

7) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

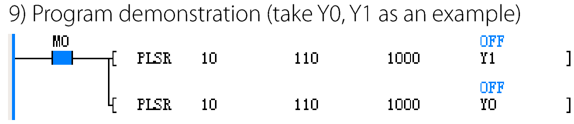
Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

8) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

9) Program demonstration (take Y0, Y1 as an example)



Program description: When M0 is ON, pulses are output from Y0 and Y1 ports according to the set value. After 110 pulses are completed, it will not output. When M0 transitions from OFF to ON, it will be output again next time. When M0 is OFF, the port output is OFF.

the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. If the maximum speed is lower than the above calculation result, there will be no pulse output.

- **Precautions:**

1) When the operand S1×S3<100000, the system will process it according to S3=100000/S1, and the system will prompt the PLSR command parameter error alarm, and the acceleration and deceleration sequence is not necessarily certain. When operand S1×S3>S2×909. The system processes it according to S3=S2×909/S1, and at the same time, the system prompts a PLSR command parameter error alarm, and the acceleration and deceleration sequence is not certain. The number of times of shifting during acceleration and deceleration is handled as a fixed 10 times, and the amount of change each time is S1/10.

2) Only the PLC with transistor output can use this instruction;

3) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, it will not accept the command to drive again.

4) High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

11.2.6 DPLSR: 32-bit counting pulse output command with acceleration and deceleration

Ladder Diagram: —| |—[DPLSR (S1) (S2) (S3) (D)]	Applicable models	VC-B VC-S VC-P
	Affect the flag	
Instruction List: DPLSR *(S1)* *(S2)* *(S3)* (D)	Step size	17

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	BOOL			Y												

- **Operand Description**

S1: Output frequency (unit: Hz). Settable range:

VC-B: Range 10~100000 (Hz)

VC-S: Range: 10～200000(Hz)

S2: Set the number of pulse outputs. The setting range is: 12～2147483647.

Acceleration and deceleration time setting special register								
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Attributes
SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306	Top speed (default: 100khz or 200khz)
SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307	
SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308	Basal velocity (default: 800hz)
SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309	Acceleration time (default 100ms)

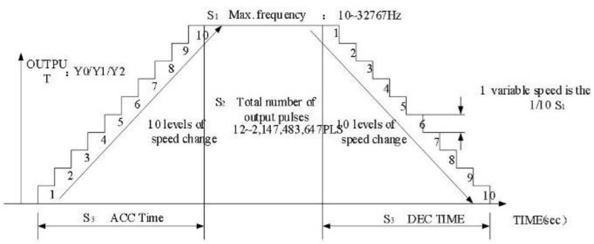
S3: Acceleration and deceleration time (unit: ms) Settable range: 10~32767 (ms) The default acceleration time is the same as the deceleration time, please pay attention when setting.

D: High-speed pulse output port designation. VC-B can specify Y0/Y1/Y2;

VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7;

- **Function Description**

With the set acceleration/deceleration time and the specified pulse frequency, the set number of pulses is output. as shown below:



4) The current pulse position can monitor the special register; see the following table:

Pulse output count cumulative SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD160	SD180	SD200	SD220	SD240	SD260	SD280	SD300
SD161	SD181	SD201	SD221	SD241	SD261	SD281	SD301

5) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

6) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

6) Modification of frequency and number of pulses during operation is not supported.

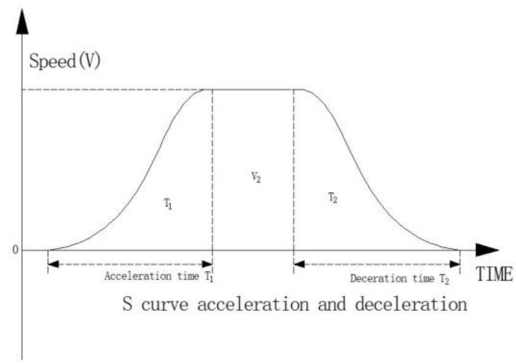
7) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300	Deceleration time (default 100ms)
Note: when using the plsr instruction, the effective value of the acceleration and deceleration time is the value set by the instruction s3; the acceleration and deceleration value set by the sd element is invalid.								

4) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below;

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: 1. Modification during command operation is invalid; 2. VC-B does not support S-type acceleration and deceleration.							



5) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results $F_{\min\_acc}$ is the minimum output frequency limit value.

For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. If the maximum speed is lower than the above calculation result, there will be no pulse output.

Precautions:

1) When the operand S1×S3<100000, the system will process it according to S3=100000/S1, and the system will prompt the PLSR command parameter error alarm, and the acceleration and deceleration sequence is not necessarily certain. When operand S1×S3>S2×909. The system

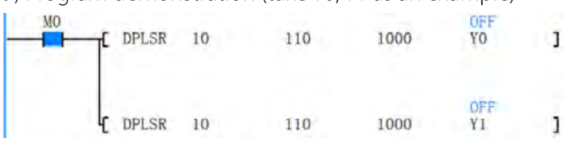
Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

8) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

9) Program demonstration (take Y0, Y1 as an example)



Program description: When M0 is ON, pulses are output from Y0 and Y1 ports according to the set value. After 110 pulses are completed, it will not output. When M0 transitions from OFF to ON, it will be output again next time. When M0 is OFF, the port output is OFF.

processes it according to S3=S2×909/S1, and at the same time, the system prompts a PLSR command parameter error alarm, and the acceleration and deceleration sequence is not certain. The number of times of shifting during acceleration and deceleration is handled as a fixed 10 times, and the amount of change each time is S1/10.

2) Only the PLC with transistor output can use this instruction;

3) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, it will not accept the command to drive again.

4) High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

11.2.7 PLS: Multi-speed pulse output command

Ladder Diagram: PLS *(S1)* *(S2)* *(D1)*															
Applicable models	VC-B VC-S VC-P														
Affect the flag															
Instruction list: PLS *(S1)* *(S2)* (D1)															
Step size	7														
Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D1	BOOL			Y												

- **Operand Description**

S1: The starting address of the D element specified by the parameter; 32-bit instruction

Pulse output frequency; VC-B range: 10~100000 (Hz); VC-S range: 10 ~ 200000 (Hz).

S2: Output segment number 0～255; when the segment number is 0, no pulse is output.

D: High-speed pulse output port. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

- **Function Description:**

Specify the high-speed output port, and output pulses continuously according to the pulse frequency and number of pulses set in each segment. Based on relative

DMOV M-stage step frequency Dn+4M-4

DMOV number of pulses in the M-stage step Dn+4M-2

DMOV maximum speed Dn+4M

MOV minimum speed Dn+4M+2

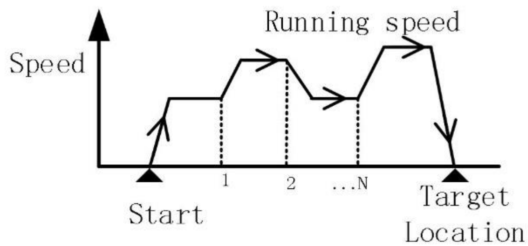
MOV acceleration time Dn+4M+3

MOV deceleration time Dn+4M+4

2) The current pulse position can monitor special registers; see the table below

Pulse output count cumulative SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD160	SD180	SD200	SD220	SD240	SD260	SD280	SD300
SD161	SD181	SD201	SD221	SD241	SD261	SD281	SD301

position movement, there is acceleration and deceleration during operation. As shown below



1) The PLS instruction can quickly create multi-segment speed programming through the "Instruction Wizard". After completing the steps, the program automatically creates two subprograms "PLS_EXE" and "PLS_SET", which can be called in the main program.

Content description of subroutine PLS_SET: (let n be the component number of D, and M be the total number of segments):

LD SM0

DMOV step 1 frequency Dn

DMOV 1st segment step pulse number Dn+2

DMOV step 2 frequency Dn+4

DMOV 2nd segment step pulse number Dn+6

DMOV 3rd stage step frequency Dn+8

DMOV 3rd segment step pulse number Dn+10

…

6) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

7) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

8) Program demonstration (take Y0 as an example, use the instruction wizard)

3) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM element during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

4) The effective selection of PLS instruction cyclic execution; when the SM element is ON, when the power flow before the instruction is maintained, the set multi-speed will be cyclically executed. When the SM element is OFF, the output stops after the execution is completed once, and the power flow needs to be re-supplied when it is restarted.

The PLS instruction loops through the active SM elements							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM274	SM294	SM314	SM334	SM354	SM374	SM394	SM414

5) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results $F_{\min\_acc}$ is the minimum output frequency limit value.

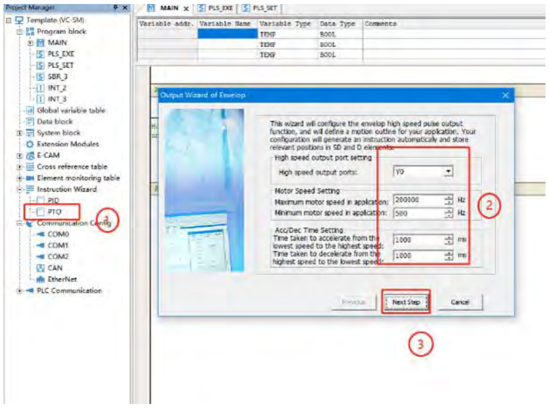
For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. If the maximum speed is lower than the above calculation result, there will be no pulse output.

- **Precautions:**

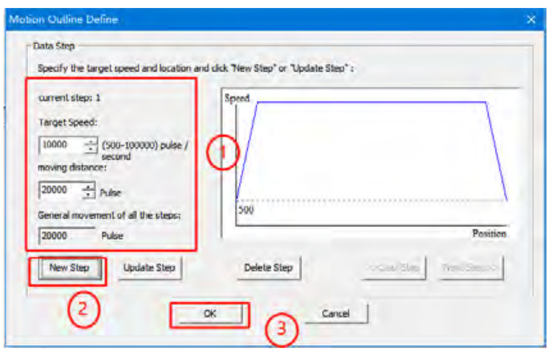
1. It is recommended to use the PLS instruction generated by the PTO wizard. If you directly write the PLS instruction, please note that the number of pulses in each step cannot be too small. Under the set acceleration, the number of pulses in each step must be greater than the minimum number of pulses required for conversion between frequencies.

2. use P Indicates the number of pulses output in a step, F_N represents the frequency of the Nth segment, $F_{\max}$, $F_{\min}$ represents the highest and lowest speed, T_{up} , T_{down} Indicates acceleration time and deceleration time,in milliseconds.

1. In Connect - Instruction Wizard - PTO as shown below



2. Set the position and speed of the first stage (10000, 5000); click ② to increase the position and speed of the second stage (20000, 10000), click ③ to complete, and enter the next D register configuration until it is completed.



3. After completion, the program automatically adds two subprograms; the subprogram is called in the main program to compile and download. (Note: Since the power flow of the PLS instruction is SM0, the program starts to output pulses when the program runs, and the user can change the control mode) as shown in the figure below:

1) When the speed of step N is greater than the speed of step N-1, the number of pulses in step N must meet the following conditions:

$$P \geq \frac{(F_N + F_{N-1}) \times (F_N - F_{N-1}) \times T_{up}}{2000 \times (F_{\max} - F_{\min})}$$

2) When the speed of step N is less than the speed of step N-1, the number of pulses in step N must meet the following conditions:

$$P \geq \frac{(F_N + F_{N-1}) \times (F_N - F_{N-1}) \times T_{down}}{2000 \times (F_{\max} - F_{\min})}$$

3. Special:

1) When N=1, the frequency of step N-1 is taken as $F_{\min}$, into the above formula.

2) When the number of all steps is 1, that is, when there is only one segment, the number of pulses must meet the following conditions:

$$P \geq \frac{(F_1 + F_{\min}) \times (F_1 - F_{\min}) \times (T_{up} + T_{down})}{2000 \times (F_{\max} - F_{\min})}$$

3) The number of pulses in the last step must satisfy the following formula:

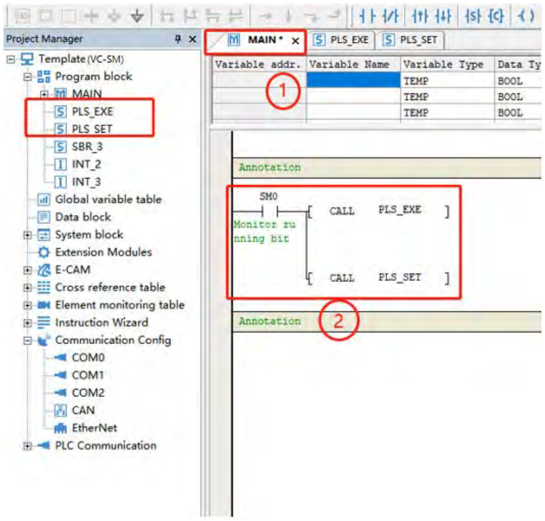
$$P \geq \frac{(F_M + F_{M-1}) \times (F_M - F_{M-1}) \times (T_{up} + T_{down})}{2000 \times (F_{\max} - F_{\min})}$$

4) The frequency specified in each step cannot be greater than the previously set maximum speed, nor lower than the minimum speed

5) Only the PLC with transistor output can use this instruction;

6) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, the command will not be driven again.

7) High-speed commands, envelope commands, and positioning commands can use Y port to output high-speed pulses. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.



11.2.8 DVIT: interrupt positioning command

Ladder Diagram: ├─┤ ├─[DVIT *(S1)* *(S2)* *(D1)* *(D2)*]								**Applicable models**		**VC-B VC-S VC-P**						
								Affect the flag								
Command list: DVIT *(S1)* *(S2)* (D1) (D2)								**Step size**		**11**						
Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D1	BOOL			Y												
D2	BOOL			Y	M	S										

- **Operand Description**

S1:The number of pulses output after the specified interrupt occurs. Range -2147483648~2147483647. The negative sign indicates the opposite direction. Its positive or negative determines the direction of the pulse output.

S2: Specify the output pulse frequency of the speed segment before the interruption occurs.

D1: Specify the output port for outputting pulses. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

D2: Specify the output port or bit variable of the rotation direction signal.

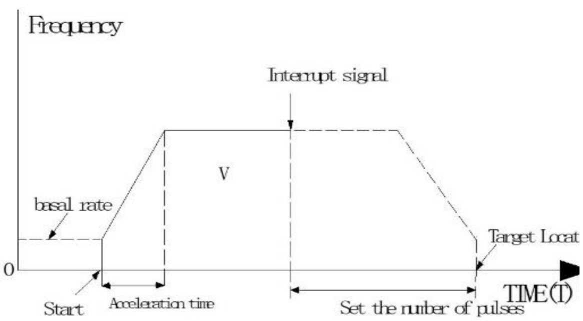
S1 is positive: D2 is ON to indicate forward running;

S1 is negative: D2 is OFF to indicate reverse operation;

Interrupt signal setting							
The interrupt signal is specified by default							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Default X0	Default X1	Default X2	Default X3	Default X4	Default X5	Default X6	Default X7
Take y0 as an example: after the dvit instruction is executed, when the x0 signal is valid, enter the interrupt output to set the number of pulses. (x point does not need to be configured) note: when using this command, it is forbidden to use the interrupt function corresponding to x point.							
Designated interrupt signal valid and designated interrupt signal sd device							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM284	SM304	SM324	SM344	SM364	SM384	SM404	SM424
SD176	SD196	SD216	SD236	SD256	SD276	SD296	SD316
Sd soft element setting value range: 0~8; (1) values 0~7 correspond to x0~x7; means: designate the x point corresponding to the value content as the interrupt input signal. Take y0 as an example: if sd176=1, it means x1 is designated as the interrupt signal, and so on. (2) the value 8 means: designate the sm interrupt device as the input source of the interrupt signal.							
Take y0 as an example: turn sm284 on, sd176=5, execute the dvit instruction, when the x5 signal is valid, enter the interrupt output to set the number of pulses.							

- **Function Description**

Command function: After the command is executed, run to the set speed segment output frequency according to the set acceleration time. When the interrupt input signal is detected, immediately enter the position segment output frequency (consistent with the speed segment frequency), and output the set output frequency. number of pulses. As shown below:



V:Indicates the frequency of operation, and the output frequency of the speed segment before the interrupt signal occurs is allowed to be modified during operation

1) Interrupt signal setting. The interrupt signal can use the default designated X point, or by enabling the SM element, the content of the designated SD soft element can be used as the interrupt signal. See the table below:

Specify the interrupt signal as the sm device							
SM285	SM305	SM325	SM345	SM365	SM385	SM405	SM425
Take y0 as an example: turn sm284 on, sd176=8, execute the dvit instruction, when the sm285 signal is valid, enter the interrupt output to set the number of pulses. (note: the interrupt sm device is affected by the scan cycle, and the timeliness is not as good as specifying the x point signal)							

2) The current pulse position can monitor the special register; see the following table:

3) "Pulse output stop Sign" can check the pulse output status,

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

4) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Top speed (default: 100khz or 200khz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
Basal velocity (default: 800hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
Acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
Deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

5) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below;

7) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{min\_acc} = \sqrt{\frac{F_{max} \times 500}{T}}$$

In the above formula, F_{max} Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results F_{min_acc} is the minimum output frequency limit value.

For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. If the maximum speed is lower than the above calculation result, there will be no pulse output.

8) Pulse output completion interrupt

6) Interrupt signal logic inversion. SM Component is set to

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modification during command operation is invalid; S-type acceleration and deceleration are applicable to DRVI, DRVA, PLSR, DPLSR, PLSV and other commands							

ON When the input is OFF, it is processed according to the negative logic: when the interrupt signal input is OFF, it is processed according to the signal ON; OFF When the signal is ON, it is processed according to the positive logic, that is: when the interrupt signal input is ON, it is processed as if the signal is ON. See the table below

Interrupt signal logic inversion enable							
Logical inversion (default: OFF)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM283	SM303	SM323	SM343	SM363	SM383	SM403	SM423

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

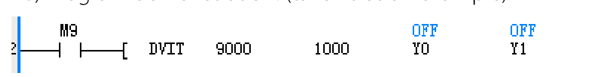
Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

9) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

10) Program demonstration: (take Y0 as an example)



When M9 is ON, it runs at 1000HZ, and interrupts when X0 rises

After that, start at 1000HZ frequency and output after 9000 pulses. Stop sending pulses.

- **Precautions**

1) Only the PLC with transistor output can use this instruction;

2) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, the command will not be driven again.

3) High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

4) For the output pulse frequency number S2, even if a value lower than the calculated result above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. However, if the maximum speed is lower than the above calculation result, there will be no pulse output.

5) When the number of output pulses is less than the number of pulses required for deceleration, it operates at a frequency that can complete deceleration.

11.2.9 DPIT: maximum fixed-length interrupt positioning instruction

Ladder Diagram:	**Applicable models**	**VC-S VC-P**

├─┤ ├─[DPIT *(S1) (S2) (D1) (D2)*]	**Affect the flag**	
Command list: DVIT *(S1) (S2)* **(D1) (D2)**	**Step size**	**11**

Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D1	BOOL			Y												
D2	BOOL			Y	M	S										

Interrupt signal setting

(1) The interrupt signal is specified by default

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
default X0	default X1	default X2	default X3	default X4	default X5	default X6	default X7

Take Y0 as an example: After the DPIT instruction is executed, when the X0 signal is valid, enter the interrupt output to set the number of pulses. (X point does not need to be configured) Note: When using this command, it is forbidden to use the interrupt function corresponding to X point.

(2) The specified interrupt signal is valid and the specified interrupt signal SD device

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM284	SM304	SM324	SM344	SM364	SM384	SM404	SM424
SD175	SD195	SD215	SD235	SD255	SD275	SD295	SD315

SD soft element setting value range: 0~8;

(1) Values 0~7 correspond to X0~X7; means: designate the X point corresponding to the value content as the interrupt input signal. Take Y0 as an example: if SD175=1, it means X1 is designated as the interrupt signal, and so on.

(2) The value 8 means: designate the SM interrupt device as the input source of the interrupt signal.

Take Y0 as an example: turn SM284 ON, SD175=5, execute the DPIT instruction, when the X5 signal is valid, enter the interrupt output to set the number of pulses.

(3) The interrupt signal is designated as the SM device

SM285	SM305	SM325	SM345	SM365	SM385	SM405	SM425

Take Y0 as an example: turn SM284 ON, SD175=8, execute the DPIT instruction, when the SM285 signal is valid, enter the interrupt output to set the number of pulses. (Note: The interrupt SM device is affected by the scan cycle, and the timeliness is not as good as specifying the X point signal)

Pulse output complete interrupt enable

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

Pulse output stop Sign

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

Current position SD register (32 bits)

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

Acceleration and deceleration time setting special register

Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
top speed (Default: 100KHz or 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

Monitor SM during pulse output

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

Type T and Type S selection settings

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426

Note: Modification during command operation is invalid; S-type acceleration and deceleration are applicable to DRVI, DRVA, PLSR, DPLSR, PLSV and other commands

Interrupt signal logic inversion enable

Logical inversion (default: OFF)

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM283	SM303	SM323	SM343	SM363	SM383	SM403	SM423

- **Operand Description**

S1: Set the maximum number of output pulses. 32-bit instructions

Range -2147483648~2147483647. The negative sign indicates the opposite direction. Its positive or negative determines the direction of the pulse output.

S1+2: The number of pulses output after the specified interrupt occurs. 32-bit instructions

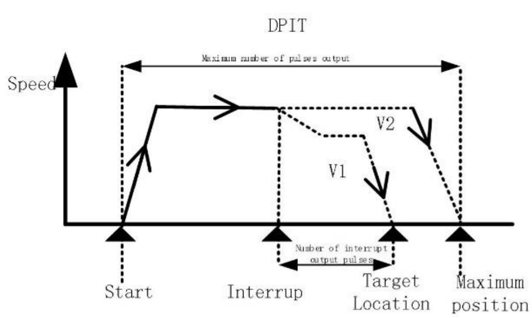
Range -2147483648~2147483647.

S2: Specify the output pulse frequency of the speed segment before the interruption occurs. 32bit

S1 is negative: D2 is OFF to indicate reverse operation;

- **Function Description**

Command function: This command outputs pulses according to the specified port, frequency and running direction. When the interrupt signal is detected, it will continue to output the given number of pulses, so that the servo actuator moves with the offset amount on the basis of the current position; if the interrupt signal is not detected during the operation, it will output the set maximum pulse. As shown below:



1)Interrupt signal setting. There are three ways: (1) The interrupt signal can use the default designated X point; (2) By enabling the SM element, the content of the specified SD device can be used as the source of the interrupt signal; (3) By specifying the SM element as the source of the interrupt signal The following table See:

instruction (10~200KHz)

S2+2: Specify the output pulse frequency of the speed segment after the interruption occurs. 32bit instruction (10~200KHz)

D1: Specify the output port for outputting pulses. VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

D2: Specify the output port or bit variable of the rotation direction signal.

S1 is positive: D2 is ON to indicate forward running;

2)The current pulse position can monitor the special register; see the following table:

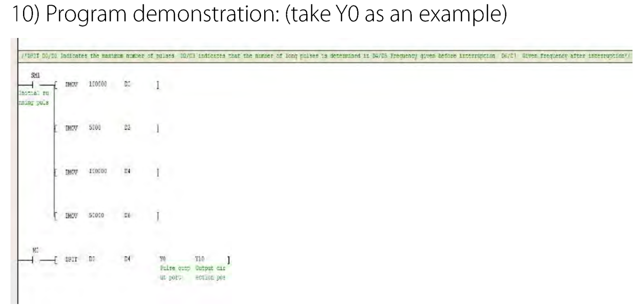
3) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

4) Support T-type and S-type acceleration and deceleration, the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

5) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below;

6) Interrupt signal logic inversion. SM Component is set to ON When the input is OFF, it is processed according to the negative logic: when the interrupt signal input is OFF, it is processed according to the signal ON; OFF When the signal is ON, it is processed according to the positive logic, that is: when the interrupt signal input is ON, it is processed as if the signal is ON. See the table below

10) Program demonstration: (take Y0 as an example)



7) The minimum frequency of the output pulse frequency that can actually be output is determined according to the following formula:

$$F_{min\_acc} = \sqrt{\frac{F_{max} \times 500}{T}}$$

In the above formula, F_{max} Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results F_{min_acc} is the minimum output frequency limit value.

For the number of output pulse frequencies, even if a value lower than that calculated above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. If the maximum speed is lower than the above calculation result, there will be no pulse output.

8) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

9) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Program description: When M0 is ON, the Y0 port runs at a frequency of 10000Hz. When the rising edge of X0 is interrupted, it switches to run at a frequency of 5000Hz, and stops sending pulses after outputting 5000 pulses. If the rising edge of X0 is not detected, continue to run at a frequency of 10000Hz and output the maximum set pulse number of 100000 and stop running.

- **Precautions**

1) Only the PLC with transistor output can use this instruction;

2) After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, it will not accept the command to drive again.

3) High-speed commands, envelope commands, and positioning commands can use Y port to output high-speed pulses. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

4) For the output pulse frequency number S2, even if a value lower than the calculated result above is specified, the frequency of the calculated value will still be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. However, if the maximum speed is lower than the above calculation result, there will be no pulse output.

5) When the number of output pulses is less than the number of pulses required for deceleration, it operates at a frequency that can complete deceleration.

11.2.10 STOPDV: pulse output stop command

Ladder Diagram: ├──┤ ├───[STOPDV *(S1)* *(S2)* *(S3)* *(D)*]														**Applicable models**	**VC-S VC-P**
														Affect the flag	
Command list: STOPDV *(S1) (S2) (S3)* (*D*)														**Step size**	**12**
Operand	Type	Applicable devices													Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		√
S3	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		√
D	BOOL			Y											

- **Operand Description**

S1: The number of output pulses after the instruction is executed.

Range: 0~2147483647;

S2: Base speed during deceleration.

Range: 100~200000Hz;

S3: The time from the original output frequency to the base speed.

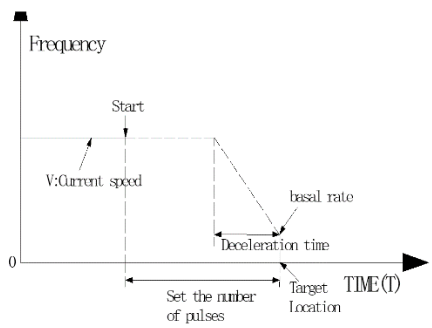
Range: 10~32767ms;

D: Specify the output port corresponding to the high-speed pulse.

VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

- **Function Description**

Instruction function description: During the execution of high-speed instruction (PLSY), envelope curve instruction (PLS), and positioning instruction (DRVI, DRVA), the currently executed action can be stopped, and the deceleration time set by the STOPDV instruction and the set pulse output the number.



1)The current pulse position can monitor the special register; see the following table:

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

2) The "pulse output stop Sign" can check the pulse output status, the Flag bit is set to ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

3)Support T-type acceleration and deceleration, the time can be set separately, the acceleration and deceleration time range:10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
top speed (Default: 100KHz or 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300
Note: The base speed deceleration time of the STOPDV instruction uses the operand of the instruction itself, and has nothing to do with the SD setting parameters.								

4)The actual minimum frequency that can be output is determined according to the following formula:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

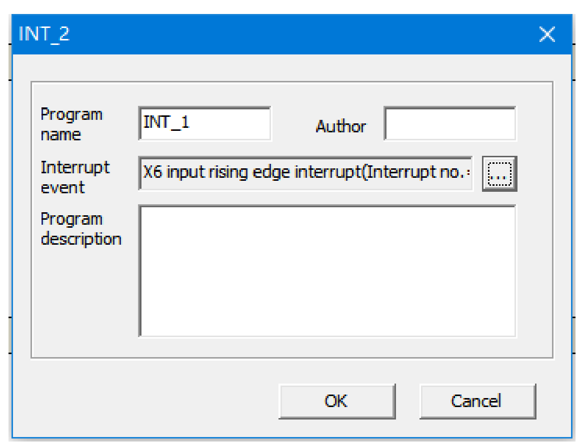
In the above formula, $F_{\max}$ Indicates the maximum speed; T Indicates the acceleration and deceleration time, in milliseconds. Calculation results $F_{\min\_acc}$ is the minimum output frequency limit value.

5) Pulse output completion interrupt

Use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit), see the following table

Pulse output complete interrupt enable

(2) Set the interrupt source (total interrupt sum) of the interrupt subroutine, for example:



Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

6) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

7) Instruction execution process description;

1.When the command drive power flow is ON, it will run for the specified number of pulses and then stop. When the specified number of pulses is 0, the output action will be stopped immediately; when the specified number of pulses is greater than 0, the original output action will be continued, and then decelerated to the base speed, and the output action will be stopped when the base speed is reached.

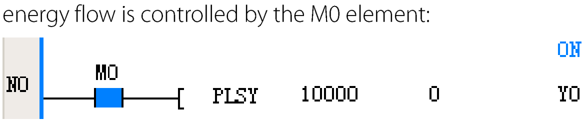
2. The base speed and acceleration/deceleration time are also set in the special data register of the output shaft, and the execution of the instruction will not change the setting of the special data register; the base speed and acceleration/deceleration time during the execution of the instruction are executed according to the setting of the instruction operand. , do not use the configuration in the special data registers.

3. The direction signal of the output shaft does not need to be specified. It automatically recognizes the direction signal specified in the original high-speed command, envelope command, and positioning command, and does not change the ON/OFF status of the direction signal during command execution.

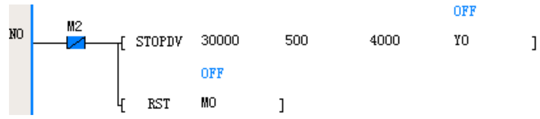
4.The STOPDV instruction can be executed in the interrupt program or in the main program. Due to the influence of the scan cycle, the execution in the interrupt program is better in real time.

8) Program demonstration (take Y0 as an example)

(1) In the main program, use the PLSY instruction to drive Y0. The energy flow is controlled by the M0 element:



(3) Add the following statement to the interrupt subroutine:



When M0 is set to ON, Y0 starts to send pulses at 10000Hz frequency. When the rising edge of X6 is valid, it enters the interrupt program and executes the STOPDV instruction, and stops the output according to the set deceleration time and the number of output 30000 pulses. From the rising edge of X6 to the complete stop of Y0, 30000 pulses will be output, which is not affected by the scan period. (Note that when calling the STOPDV instruction, it is necessary to cut off the relevant energy flow of the high-speed instruction operating on Y0 in the main function at the same time, so as to prevent the high-speed output from being restarted after the instruction is scanned in the main function after Y0 stops).

- **Precautions**

1. Only the PLC with transistor output can use this instruction;

2. For the output pulse frequency number S2, even if a value lower than that calculated above is specified, the frequency of the calculated value will be output. The frequency of the initial part of acceleration and the final part of deceleration cannot be lower than the above calculation result. However, if the maximum speed is lower than the above calculation result, there will be no pulse output.

3. When the number of output pulses is less than the number of pulses required for deceleration, it operates at a frequency that can complete deceleration.

5.When executing the command, if the output shaft is already in the stopped state, nothing will be done; if the output shaft is executing LIN, CW, CCW commands, nothing will be done.

11.3 High Speed Command

11.3.1 PLSY: High-speed pulse output command

Ladder Diagram: ⊢⊣ ⊢[PLSY *(S1)* *(S2)* *(D)*]	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	

Instruction list: PLSY *(S1)* *(S2)* *(D)*												Step size		9		
Operand	Type	Applicable devices														Index
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	BOOL			Y												

- **Operand Description**

S1: output frequency (Hz). 32-bit instructions

Range VC-B: 10~100000Hz: VC-S: 10~200000Hz;

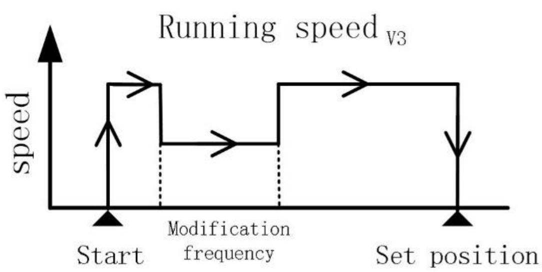
S2: set output number of pulses. 32-bit instructions

Range -2147483648~2147483647. When S2 is equal to zero, an uninterrupted infinite number of pulses are sent.

D:High-speed pulse output port. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

- **Function Description**

At the specified pulse frequency, the set number of pulses is output. As shown below:



1) Current pulse position, special register can be monitored; see the table below

Pulse output count cumulative SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD160	SD180	SD200	SD220	SD240	SD260	SD280	SD300
SD161	SD181	SD201	SD221	SD241	SD261	SD281	SD301

2) "Pulse output stop Sign" can check the pulse output status, the Sign is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

3) Does not support acceleration and deceleration function;

4) During the execution of the instruction, it is allowed to modify the output frequency. (Can be large or small) No need to set special SM components.

5) When the pulse output number setting bit is 0, the PLSY instruction is in the speed mode, which is to send uninterrupted infinite pulses.

6) When the number of pulses is changed during the instruction running, the operand will only take effect when the next drive is valid.

7) Pulse output completion interrupt

To use the pulse out to complete the interrupt, you need to set the SM special element (interrupt enable Flag bit) as shown in the following table:

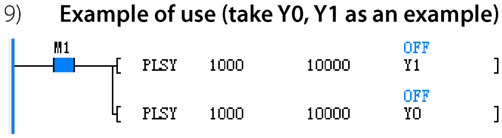
Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

8) Control pulse output stop

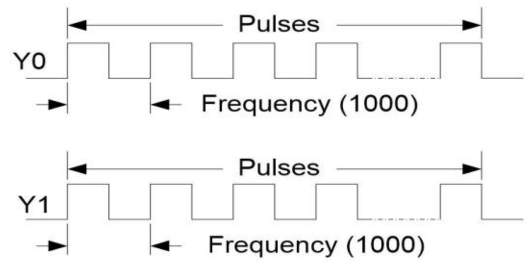
By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

9) **Example of use (take Y0, Y1 as an example)**



Program Description: When M1 is ON, 10000 pulses with a frequency of 1000Hz are output from Y0 and Y1 ports, and no longer output after 10000 pulses are completed. When M1 transitions from OFF to ON, it will be output again next time. When M1 is OFF, the port output is OFF. As shown below:



- **Precautions:**

1. Only the PLC with transistor output can use this command;

2. After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, the command will not be driven again.

3. High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

4. When S1 is not within the set range, the system reports that the instruction operand is illegal, and no pulse is output at this time.

5. When the set operand is not within this range, the system will report that the instruction operand is illegal, the pulse will not be output, and system resources will not be occupied.

11.3.2 PLSV: Variable speed pulse output command

Ladder Diagram: PLSV (S) (D1) (D2)		**Applicable models**	**VC-B VC-S VC-P**
		Affect the flag	
Instruction list: PLSV (S) (D1) (D2)		**Step size**	**8**

Operand	Type	Applicable devices													Index
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V	R	√
D1	BOOL			Y											
D2	BOOL			Y	M	S									

- **Operand Description**

S: Specify output pulse frequency (Hz); range VC-B: 10～100000 (Hz)

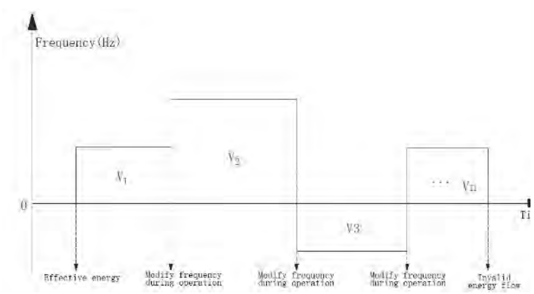
, -10 to -100000 (Hz); VC-S: 10 to 200000 (Hz), -10 to -200000 (Hz). The negative sign indicates the command signal for reverse operation.

D1: High-speed pulse output port. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

D2: Rotation direction signal output start address. Corresponding to the positive and negative conditions of S, the actions are as follows: S is positive: D2 is ON. S is negative: D2 is OFF.

- **Function Description**

Command function: output pulses with the specified output port, set pulse frequency and direction bit, and can set whether to have acceleration and deceleration functions (the default is no acceleration and deceleration). If the power flow is invalid, the output pulse will be stopped immediately.



1) The current pulse position can monitor the SD special register; see the following table

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

2) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF:

Monitoring in pulse output SM							
Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0
SM271	SM271	SM271	SM271	SM271	SM271	SM271	SM271

3) Support T-type and S-type acceleration and deceleration (VC-B VC-S VC-P only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Top speed (default: 100khz or 200khz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
Basal velocity (default: 800hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
Acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
Deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

4) The default is T-type acceleration and deceleration. When the SM special auxiliary relay is ON, S-type acceleration and deceleration is enabled. See the table below;

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modification during command operation is invalid;							

5) The default PLSV instruction does not have the function of acceleration and deceleration, but the function of acceleration and deceleration can be enabled in the process of frequency conversion by setting the SM "progressive frequency conversion Sign". See the table below:

Progressive frequency conversion logo							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM275	SM295	SM315	SM335	SM355	SM375	SM395	SM415
Take Y0 as an example: the progressive Flag bit SM275 is ON, and the instruction performs acceleration, deceleration and frequency conversion according to the acceleration and deceleration time set by SD169 and SD170. Note: It is invalid to modify this Flag bit during the execution of the instruction.							

6)During the command operation, the output frequency can be freely modified without setting the SM special Sign. The plus or minus of the frequency indicates the direction bit.

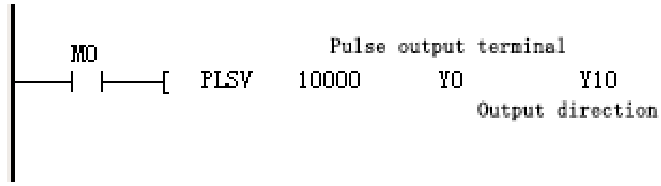
7) The PLSV instruction is a speed control instruction, so there is no pulse output completion interruption.

8) Control the pulse output to stop;

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

9) Program example (Y0 as an example)



When M0 is ON, the Y0 port sends pulses at a frequency of 10000Hz, and Y10 is used to control the running direction. When Y10 is ON, it means forward running.

- **Precautions**:

1. Only PLC with transistor output can use this command.

2. High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

11.3.3 PWM: Pulse output command

Ladder Diagram: PWM *(S1)* *(S2)* *(D)*	**Applicable models**	**VC-B VC-S VC-P**
	Affect the flag	
Instruction List: PWM ***(S1) (S2) (D)***	**Step size**	**7**

Operand	Type	Applicable devices													Index
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	BOOL			Y												

- **Operand Description**

S1: Specify the pulse width (unit: ms or μs)

The setting range is: 0～32767 (ms). When S1 is greater than 32767, the system will report that the instruction operand is illegal. Must be S1≤S2

S2: Specify the pulse period (unit: ms)

Settable range: 1～32767, if the set operand is not within this range, the system will report that the instruction operand is

3) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

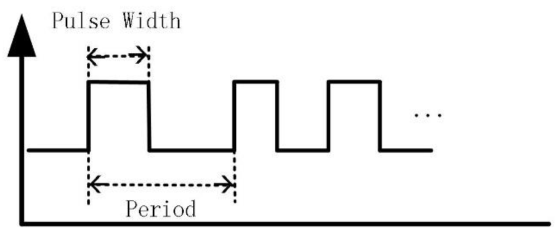
4) Example of use (take Y0 as an example)

illegal, and the pulse will not be output at the same time. Must be S1≤S2

D: High-speed pulse output port. VC-B can specify Y0/Y1/Y2; VC-S can specify Y0/Y1/Y2/Y3/Y4/Y5/Y6/Y7.

- **Function Description**

Specify the high-speed pulse output port, and output the modulated square wave according to the set pulse width and pulse period. As shown below:

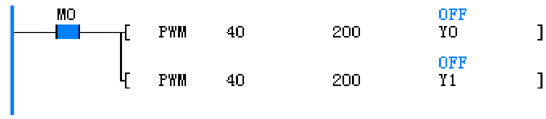


1) Pulse width unit ms or μs selection

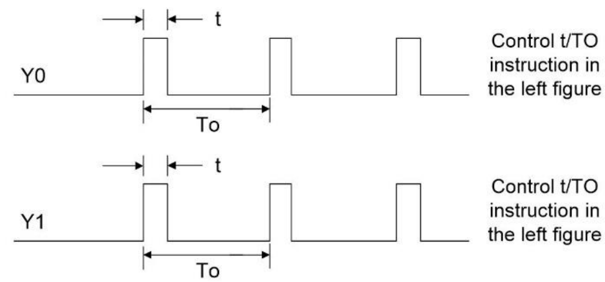
When the SM element is OFF, it means that "ms" is selected as the pulse width unit, and when the SM element is ON, it means that "μs" is selected as shown in the following table:

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM272	SM292	SM312	SM332	SM352	SM372	SM392	SM412
Take Y0 as an example: when SM272 is OFF, S1 is in ms; when SM272 is ON, S1 is in μs.							

2) During the operation of the command, it is allowed to modify S1 (pulse width) and S2 (pulse period), and the output pulse will also change accordingly.



Program Description: When M0 is ON, the Y0 and Y1 ports output PWM pulses with a width of 40ms and a period of 200ms. When M0 is OFF, the output is OFF. The output state is not affected by the scan period. As shown in the figure below: (t is the pulse width, T0 is the pulse period)



●Precautions:

1. Only the PLC with transistor output can use this command;

2. After the command drive power flow is turned OFF, when the high-speed pulse output Sign is ON, the command will not be driven again.

3. High-speed commands, envelope commands, and positioning commands can output high-speed pulses using the Y port. Be careful not to use these instructions for high-speed pulse output on the same high-speed port at the same time.

11.3.4 HTOUCH:Read position capture instruction

Ladder Diagram: HTOUCH S1 S2 S3 D1 D2		**Applicable models**	**VC-S VC-P**
		Affect the flag	
Instruction list: HTOUCH ***(S1) (S2) (S3)*** **(D1) (D2)**		**Step size**	**7**

Operand	Type	Applicable devices														Index
S1	INT	Constant						D						R		√
S2	BOOL	Constant	X													√
S3	BOOL	Constant			M			S								
D1	BOOL	Constant			M			S								

D2	REAL	Constant							D						R	

- **Operand Description**

S1: Latched operand; value range 0~15;

S2: trigger input address; selection range X0~X7;

S3: Trigger edge selection; 0 is rising edge, 1 is falling edge

D1: Position capture Sign; when it is ON, it indicates that the instruction capture is completed.

(Note: the corresponding high-speed input interrupt must be turned on to have the completion Sign, otherwise it will always be OFF, and only the first read after the trigger probe is completed is valid)

D2: store the capture position value;

- **Function Description**

Using the HTOUCH instruction, the counter or pulse axis position value can be latched when the external input trigger condition is valid.

(1) S1 latch selection description:

S1 value	Select latching high-speed counter number or pulse axis
0	C236/C244/C246/C248/C252/C254
1	C237/C256/C260/C262
2	C238 / C249
3	C239 / C257
4	C240 / C245 /C247 /C250 /C253 /C255
5	C241/C258/C261/C263
6	C242 / C251
7	C243 / C259
8	Y0 output pulse timer
9	Y1 output pulse timer
10	Y2 output pulse timer
11	Y3 output pulse timer
12	Y4 output pulse timer
13	Y5 output pulse timer
14	Y6 output pulse timer
15	Y7 output pulse timer
Note: The selection of high-speed counter number depends on the user's use of the counter	

- **Precautions**

1. The HTOUCH instruction allows users to use up to 8

- **Example of use**

For example, X0 counts at high speed, and the counter is C236. When M1=ON, use the HTOUCH instruction, input the trigger condition externally, and when the X1 falling edge trigger is valid, the value of the latch counter C236 is put into the D12 register

11.4 Interpolation Command

11.4.1 LIN: Linear path interpolation

Ladder Diagram: ├─┤ ├──[LIN (S) (D1) (D2)]	Applicable models	VC-SM
	Affect the flag	
Command List: LIN(S) (D1) (D2)	Step size	12

Operand	Type	Applicable devices													Index
S	INT								D						
D1	BOOL			Y											
D2	BOOL			Y											

- **Operand Description**

  S: The starting address of the parameter table memory area.

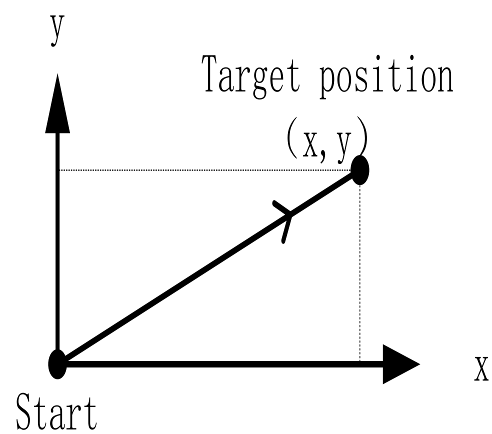
  D1: Output point number corresponding to X-axis pulse signal (or positive pulse signal).can only be specifiedY0/Y4.

  D1+1: Output point number corresponding to X-axis direction signal (or negative pulse signal). can only refer toCertainlyY1/Y5.

  D2: Output point number corresponding to Y-axis pulse signal (or positive pulse signal).can only be specifiedY2/Y6.

  D2+1: Output point number corresponding to Y-axis direction signal (or negative pulse signal).Only Y3/Y7

  (Support 2-axis linear interpolation: X axis: Y0/Y4;Y axis: Y2/Y6)



- **Function Description**

  1. Move to the target position along a straight line at the specified vector speed. As the picture on the right

  2. Parameter table definition

D element	Content
S	Reserve

D element	Content		
S+1	Action mode and output logic relationship (configuration code is in decimal)		
	Configure	Model	Output logic
	00	Incremental	pulse + direction (Forward ON/Reverse OFF)
	01	Incremental	Forward pulse + reverse pulse
	10	absolute value	pulse + direction (Forward ON/Reverse OFF)
	11	absolute value	Forward pulse + reverse pulse
S+2	Composite speed initial speed Fmin (Hz) (high)		
S+3	Composite speed initial speed Fmin (Hz) (low)		
S+4	Synthesis speed Maximum speed Fmax (Hz) (high)		
S+5	Combined speed maximum speed Fmax (Hz) (low)		
S+6	Acceleration / deceleration time T (ms) (high)		
S+7	Acceleration/deceleration time T (ms) (low order)		
S+8	X-axis target position (movement distance) (high position)		
S+9	X-axis target position (movement distance) (low position)		
S+10	Y-axis target position (moving distance) (high position)		
S+11	Y-axis target position (moving distance) (low position)		

In:

A. Incremental mode: The trajectory target adopts relative address, which refers to the moving distance of X and Y axes during the movement from the current position to the target.

B. Absolute value mode: The track target adopts the absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes.

1) When using the LIN command, the parameter settings such as acceleration and deceleration time are subject to the X-axis Y0;

2) The current pulse position can monitor the special register; see the following table:

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

3) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

4) Support T-type and S-type acceleration and deceleration, the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Top speed (default: 200khz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
Basal velocity (default: 800hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
Acceleration time (default: 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
Deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

5) The default is T-type acceleration and deceleration, and S-type acceleration and deceleration is enabled when the SM special auxiliary relay is ON. See the table below

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modification during command operation is invalid; S-type acceleration and deceleration are applicable to DRVI, DRVA, PLSR, DPLSR, PLSV and other commands							

6) Pulse output completion interrupt

Use the pulse output to complete the interrupt, you need to set the SM special component (interrupt enable Flag bit) as shown in the following table: (interpolation of two axes Y0/Y2 will only generate one output completion interrupt)

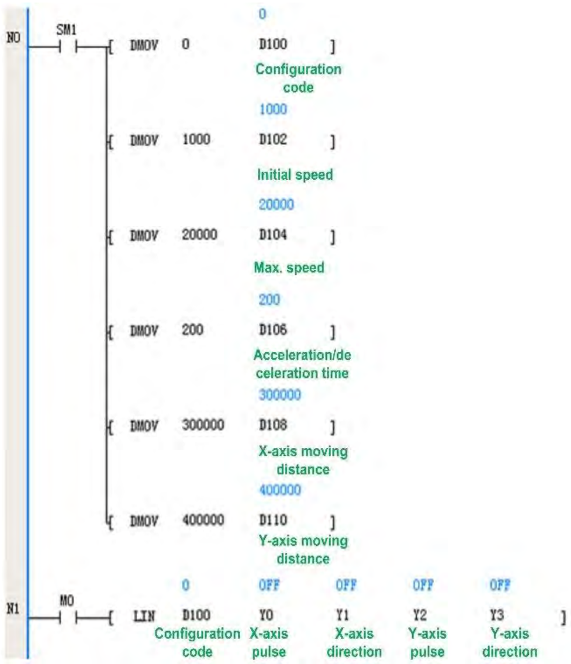
Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

7) Control pulse output stop

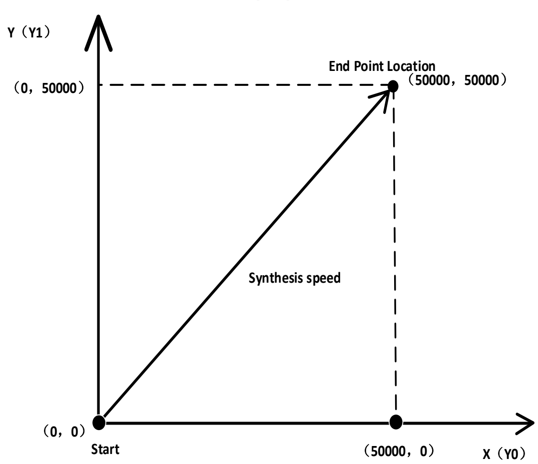
By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

Example of use: (take Y0 Y2 as an example)



Program description: The current position is 0 before starting. When M0 is ON, the interpolation command will be executed, and the coordinates of the target position (50000, 50000) will be run as shown in the following figure:



- **Precautions**

① The output point numbers D1 and D1+1 corresponding to the two output shaft pulse signals (or forward pulse signals) in the command must be used in groups. When only Y0 and Y2 can be specified in the output group, Y1 and Y3 are used as Y0 and Y2 respectively. Used in conjunction with the direction signal or negative pulse output signal.

② The output group (Y0, Y2) can be specified as "pulse + direction" mode or "positive pulse + negative pulse" mode, the maximum speed of single axis is 200k; the maximum speed of synthesis is 200K.

③ The moving distance setting range of each axis is -2147483648~2147483647 pulses.

④ Be careful not to use multiple high-speed commands, envelope commands, or positioning commands for the same high-speed port at the same time.

⑤ The acceleration and deceleration time range is 5-5000ms.

⑥ Completion interrupt only supports one (Y0Y1) interrupt.

11.4.2 CW: Clockwise arc path interpolation

Ladder Diagram: ⊣⊢ ⊣[CW (S) (D1) (D2)]	**Applicable models**	**VC-SM**
	Affect the flag	
Command List: CW(S) (*D1*) (*D2*)	**step size**	**12**

operand	type	Applicable devices													index
S	INT								D						
D1	BOOL			Y											

D2	BOOL			Y											

- **Operand Description**

S:The starting address of the parameter table memory area.

D1: Output point number corresponding to X-axis pulse signal (or positive pulse signal).Only Y0 can be specified/Y4.

D1+1: Output point number corresponding to X-axis direction signal (or negative pulse signal).Y1 is specified by default/Y3.

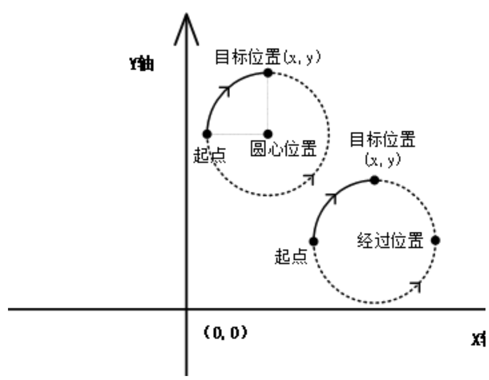
D2: Output point number corresponding to Y-axis pulse signal (or positive pulse signal).Only Y2 can be specified/Y6.

D2+1: Output point number corresponding to Y-axis direction signal (or negative pulse signal).Y3 is specified by default/Y7.

(Support 2-axis linear interpolation: X axis: Y0/Y4;Y axis: Y2/Y6)

- **Function Description**

1. According to the specified linear speed, move clockwise along the arc trajectory to the target position. As shown below:



2. Parameter table definition

<table>
<tr><th>D element</th><th colspan="2">content</th></tr>
<tr><td rowspan="4">S</td><td colspan="2">Arc forming method</td></tr>
<tr><td>configure</td><td>model</td></tr>
<tr><td>0</td><td>Center position specification</td></tr>
<tr><td>1</td><td>specified by location</td></tr>
</table>

<table>
<tr><td rowspan="6">S+1</td><td colspan="3">Action mode and output logic relationship (configuration code is in decimal)</td></tr>
<tr><td>configure</td><td>model</td><td>output logic</td></tr>
<tr><td>00</td><td>Incremental</td><td>pulse + direction (Forward ON/Reverse OFF)</td></tr>
<tr><td>01</td><td>Incremental</td><td>Forward pulse + reverse pulse</td></tr>
<tr><td>10</td><td>absolute value</td><td>pulse + direction (Forward ON/Reverse OFF)</td></tr>
<tr><td>11</td><td>absolute value</td><td>Forward pulse + reverse pulse</td></tr>
<tr><td>S+2</td><td colspan="3">Composite speed initial speed (high)</td></tr>
<tr><td>S+3</td><td colspan="3">Composite speed initial speed (low)</td></tr>
<tr><td>S+4</td><td colspan="3">Synthesis Speed (High)</td></tr>
<tr><td>S+5</td><td colspan="3">Synthesis speed (low order)</td></tr>
<tr><td>S+6</td><td colspan="3">Acceleration / deceleration time T (ms) (high)</td></tr>
<tr><td>S+7</td><td colspan="3">Acceleration/deceleration time T (ms) (low order)</td></tr>
<tr><td>S+8</td><td colspan="3">X-axis moving distance (target position) (high position)</td></tr>
<tr><td>S+9</td><td colspan="3">X-axis moving distance (target position) (low position)</td></tr>
<tr><td>S+10</td><td colspan="3">Y-axis moving distance (target position) (high position)</td></tr>
<tr><td>S+11</td><td colspan="3">Y-axis moving distance (target position) (low position)</td></tr>
<tr><td>S+12</td><td colspan="3">X-axis center position/passing point X-coordinate (high position)</td></tr>
<tr><td>S+13</td><td colspan="3">X-axis center position/passing point X-coordinate (low position)</td></tr>
<tr><td>S+14</td><td colspan="3">Y-axis center position / passing point Y-coordinate (high position)</td></tr>
<tr><td>S+15</td><td colspan="3">Y-axis center position/passing point Y-coordinate (low position)</td></tr>
</table>

in:

(1) In the incremental mode, the trajectory target uses a relative address, which refers to the moving distance of the X and Y axes during the movement from the current position to the target.

(2) In the absolute value mode, the track target adopts the absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes.

1) When using the CW command, the parameter settings such as acceleration and deceleration time are based on the X axis and Y0;

2) The current pulse position can monitor the special register; see the following table:

3)

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

4) "Pulse output stop flag" can check the pulse output status, the flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

5) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register								
Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
top speed (Default: 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

6) The default is T-shaped acceleration and deceleration. When the SM element is ON, S-shaped acceleration and deceleration are enabled. See the table below

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modification during command operation is invalid; S-type acceleration and deceleration are applicable to DRVI, DRVA, PLSR, DPLSR, PLSV and other commands							

7) Pulse output completion interrupt

Use the pulse output to complete the interrupt, you need to set the SM special component (interrupt enable flag bit) as shown in the following table (interpolation of two axes Y0/Y2 will only generate one output completion interrupt)

Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

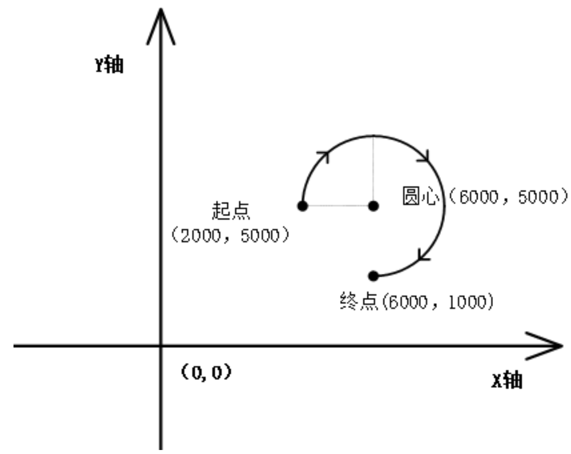
8) Control pulse output stop

By setting the SM "pulse output stop flag", the running pulse command will immediately decelerate and stop the output pulse. see table below

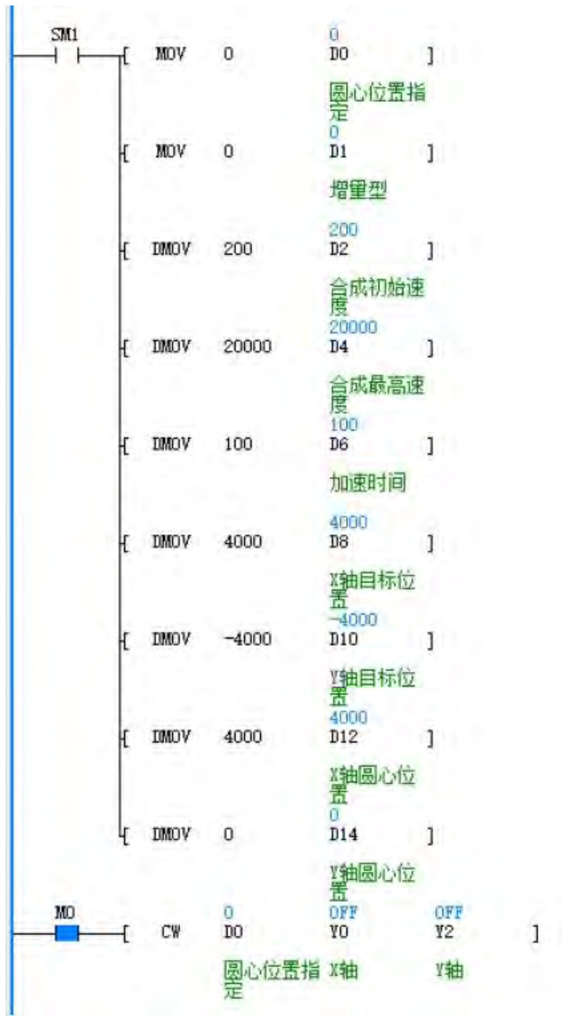
Pulse output stop flag							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

Example of use: (take Y0 Y2 as an example)

Assuming that the SD component value of the current position is (2000, 5000), I hope to draw the arc shown in the following figure:



Program programming: using incremental mode, the displacement of the end point relative to the starting point is (4000, -4000), and the displacement of the circle center relative to the starting point is (4000, 0)



- **Precautions**

1) Two output shaft pulse signals (or forward pulse signals) in the command

The corresponding output point numbers D1 and D1+1 must be used in groups.

When only Y0 and Y2 can be specified in the group, Y1 and Y3 are used as and

Direction signal or negative pulse output signal used in conjunction with Y0 and Y2.

2) Output groups (Y0, Y2) can be specified as "pulse+direction" mode or

"Positive pulse + negative pulse" mode, the maximum speed of single axis is 200k; the maximum speed of composite is 200K.

3) Be careful not to use multiple high-speed commands, envelope commands, or positioning commands for the same high-speed port at the same time.

4) The acceleration and deceleration time range is 5-5000ms.

Completion interrupt only supports one (Y0Y1) interrupt.

11.4.3 CCW: Counterclockwise circular path interpolation

Ladder Diagram: CCW *(S)* *(D1)* *(D2)*	**Applicable models**	**VC-SM**
	Affect the flag	
Instruction List: CCW (S) *(D1)* *(D2)*	**Step size**	**12**

Operand	Type	Applicable devices													Index
S	INT								D						
D1	BOOL			Y											
D2	BOOL			Y											

- **Operand Description**

S: The starting address of the parameter table memory area.

D1: Output point number corresponding to X-axis pulse signal (or positive pulse signal).Only Y0 can be specified/Y4.

D1+1: Output point number corresponding to X-axis direction signal (or negative pulse signal).Only Y1 can be specified/Y5.

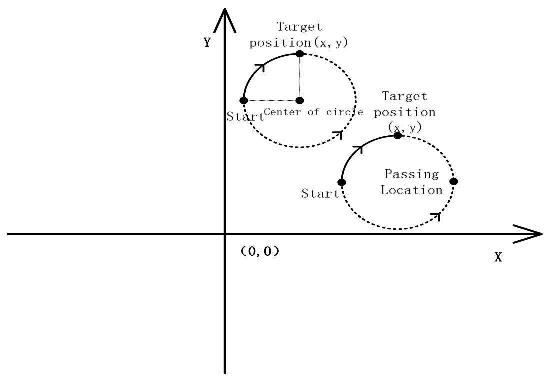
D2: Output point number corresponding to Y-axis pulse signal (or positive pulse signal).Only Y can be specified2/Y6.

D2+1: Output point number corresponding to Y-axis direction signal (or negative pulse signal).Only Y3 can be specified/Y5.

(Support 2-axis linear interpolation: X-axis: Y0/Y4; Y-axis: Y2/Y6)

- **Function Description**

1. According to the specified linear velocity, it moves to the target position along the arc track in the counterclockwise direction. As shown below



2. Parameter table definition

<table>
<tr><th>D element</th><th>Content</th></tr>
<tr><td>S</td><td>Arc forming method<table><tr><th>Configure</th><th>Model</th></tr><tr><td>0</td><td>Center position specification</td></tr><tr><td>1</td><td>specified by location</td></tr></table></td></tr>
<tr><td>S+1</td><td>Action mode and output logic relationship (configuration code is in decimal)<table><tr><th>Configure</th><th>Model</th><th>Output logic</th></tr><tr><td>00</td><td>Incremental</td><td>pulse + direction (Forward ON/Reverse OFF)</td></tr><tr><td>01</td><td>Incremental</td><td>Forward pulse + reverse pulse</td></tr><tr><td>10</td><td>absolute value</td><td>pulse + direction (Forward ON/Reverse OFF)</td></tr><tr><td>11</td><td>absolute value</td><td>Forward pulse + reverse pulse</td></tr></table></td></tr>
<tr><td>S+2</td><td>Composite speed initial speed (high)</td></tr>
<tr><td>S+3</td><td>Composite speed initial speed (low)</td></tr>
<tr><td>S+4</td><td>Synthesis Speed (High)</td></tr>
<tr><td>S+5</td><td>Synthesis speed (low order)</td></tr>
<tr><td>S+6</td><td>Acceleration / deceleration time T (ms) (high)</td></tr>
<tr><td>S+7</td><td>Acceleration/deceleration time T (ms) (low order)</td></tr>
<tr><td>S+8</td><td>X-axis moving distance (target position) (high position)</td></tr>
<tr><td>S+9</td><td>X-axis moving distance (target position) (low position)</td></tr>
<tr><td>S+10</td><td>Y-axis moving distance (target position) (high position)</td></tr>
<tr><td>S+11</td><td>Y-axis moving distance (target position) (low position)</td></tr>
<tr><td>S+12</td><td>X-axis center position/passing point X-coordinate (high position)</td></tr>
<tr><td>S+13</td><td>X-axis center position/passing point X-coordinate (low position)</td></tr>
<tr><td>S+14</td><td>Y-axis center position / passing point Y-coordinate (high position)</td></tr>
<tr><td>S+15</td><td>Y-axis center position/passing point Y-coordinate (low position)</td></tr>
</table>

in:

(1) In the incremental mode, the trajectory target uses a relative address, which refers to the moving distance of the X and Y axes during the movement from the current position to the target.

(2) In the absolute value mode, the track target adopts the absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes.

1) When using the CW command, the parameter settings such as acceleration and deceleration time are based on the X axis and Y0;

2) The current pulse position can monitor the special register; see the following table:

Current position SD register (32 bits)							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SD162	SD182	SD202	SD222	SD242	SD262	SD282	SD302
SD163	SD183	SD203	SD223	SD243	SD262	SD283	SD303

3) "Pulse output stop Sign" can check the pulse output status, the Flag bit is turned ON in the pulse output, and the output is automatically turned OFF;

Monitor SM during pulse output							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM271	SM291	SM311	SM331	SM351	SM371	SM391	SM411

4) Support T-type and S-type acceleration and deceleration (VC-B only supports T-type), the time can be set separately, the acceleration and deceleration time range: 10~32767ms;

Acceleration and deceleration time setting special register

Attributes	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Top speed (Default: 200KHz)	SD166	SD186	SD206	SD226	SD246	SD266	SD286	SD306
	SD167	SD187	SD207	SD227	SD247	SD267	SD287	SD307
Basal velocity (default: 800Hz)	SD168	SD188	SD208	SD228	SD248	SD268	SD288	SD308
Acceleration time (default 100ms)	SD169	SD189	SD209	SD229	SD249	SD269	SD289	SD309
Deceleration time (default 100ms)	SD170	SD190	SD210	SD230	SD250	SD270	SD290	SD300

5) The default is T-shaped acceleration and deceleration. When the SM element is ON, S-shaped acceleration and deceleration are enabled. See the table below

Type T and Type S selection settings							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM286	SM306	SM326	SM346	SM366	SM386	SM406	SM426
Note: Modification during command operation is invalid; S-type acceleration and deceleration are applicable to DRVI, DRVA, PLSR, DPLSR, PLSV and other commands							

6) Pulse output completion interrupt

Use the pulse output to complete the interrupt, you need to set the SM special component (interrupt enable Flag bit) as shown in the following table (interpolation of two axes Y0/Y2 will only generate one output completion interrupt)

Pulse output complete interrupt enable							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM50	SM51	SM52	SM53	SM54	SM55	SM56	SM57

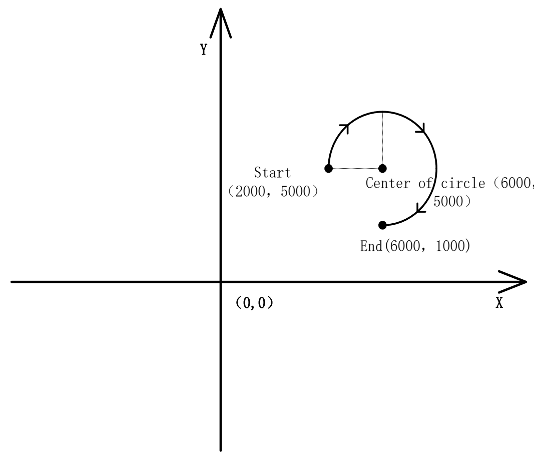
7) Control pulse output stop

By setting the SM "pulse output stop Sign", the running pulse command will immediately decelerate and stop the output pulse. see table below

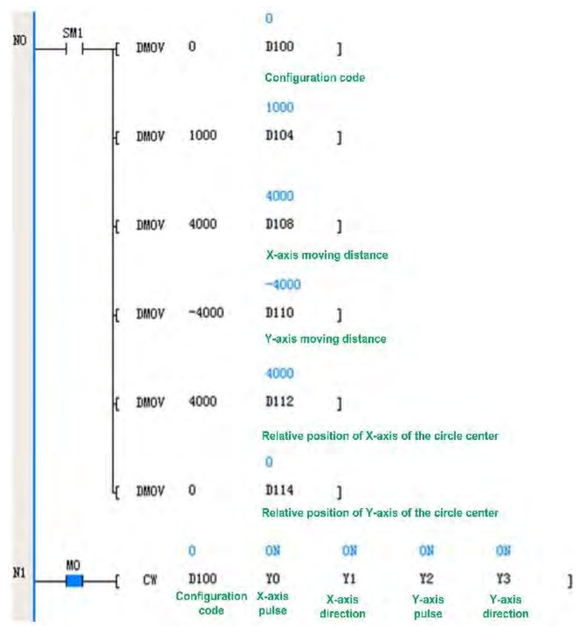
Pulse output stop Sign							
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SM270	SM290	SM310	SM330	SM350	SM370	SM390	SM410

Example of use: (take Y0 Y2 as an example)

Assuming that the SD component value of the current position is (6000, 1000), I hope to draw the arc shown in the following figure:



Program programming: using incremental mode, the displacement of the end point relative to the starting point is (-4000, 4000), and the displacement of the center of the circle relative to the starting point is (0, 4000)



- **Precautions**

1) Two output shaft pulse signals (or forward pulse signals) in the command

The corresponding output point numbers D1 and D1+1 must be used in groups

When only Y0 and Y2 can be specified for group output, Y1 and Y3 are respectively used as

Direction signal or negative pulse output signal used with Y0 and Y2.

2) Output groups (Y0, Y2) can be specified as "pulse+direction" mode or

"Positive pulse+negative pulse" mode, single axis maximum speed 200k; The maximum synthetic speed is 200K.

3) Be careful not to use multiple high-speed commands, envelope commands or positioning commands for the same high-speed port at the same time.

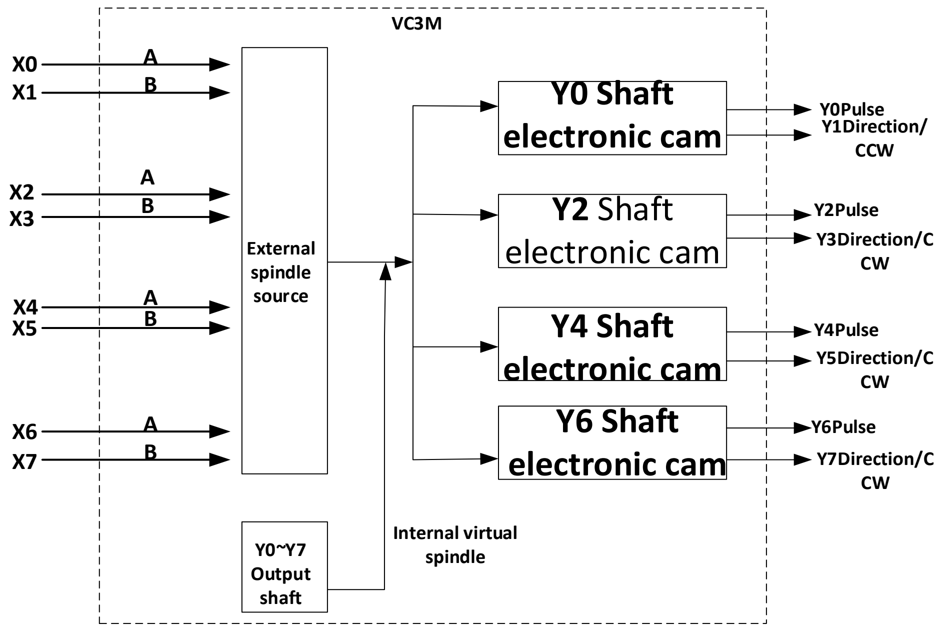
4) Acceleration and deceleration time range: 5-5000ms.

5) The completion interrupt only supports one (Y0Y1) interrupt.

Chapter 12 Electronic Cams

12.1 Electronic Cam Overview

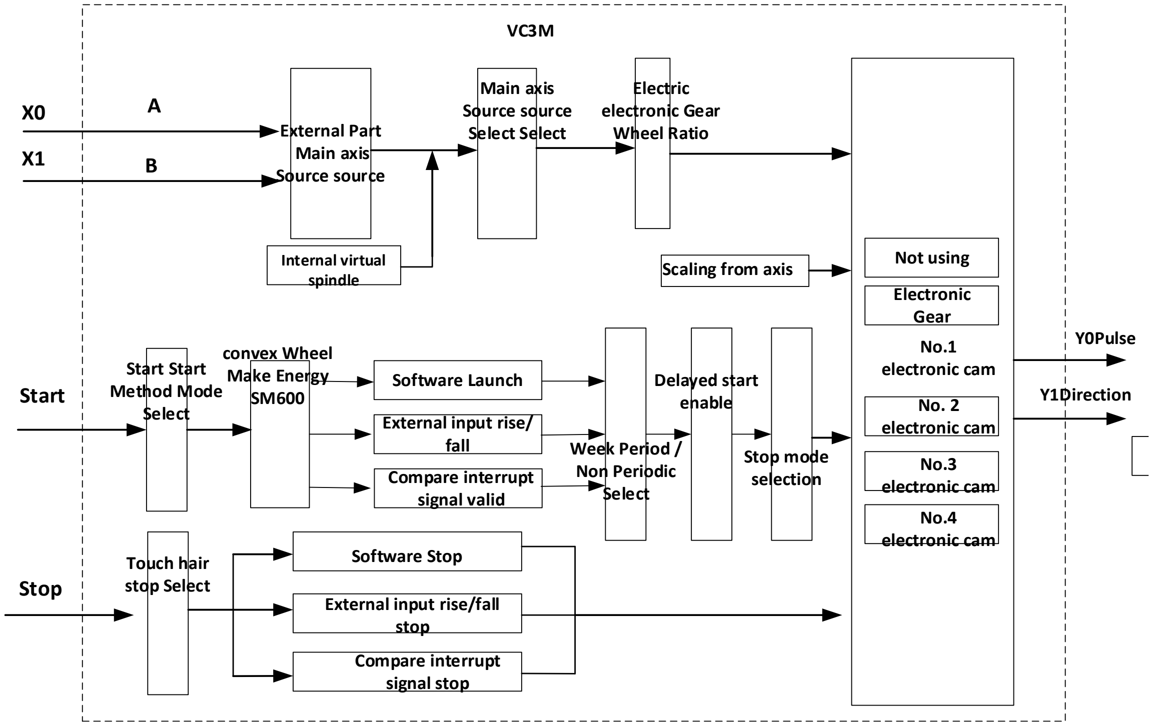
(1) The VC-S series includes the VC-S general-purpose model and the VC-SM motion control model, of which the VC-SM motion control model has 4-axis electronic cam or electronic gear control, and the 4-axis electronic cam frame module is shown in the following figure.



12.1.1 Electronic cam basic architecture

(1) The VC-SM main module integrates 4-axis electronic cams, which can realize the function of following the Primary or electronic gear synchronization by any electronic cam motion trajectory set by the electronic cam table; the Primary input can choose external input, and the external input can choose single-phase or AB-phase high-speed input as the electronic cam Primary input source. The internal virtual axis (Y0~Y7) can also be selected as the source of the electronic camshaft Primary input. When internal virtual connection is selected for Primary input, pulse commands such as positioning of the specified virtual connection axis need to be called as the Primary source signal.

(2) The basic function of the 4-axis electronic cam is the same. Taking the Y0-axis electronic camshaft as an example, the basic framework of the electronic cam for a single axis is shown in the following figure.



12.1.2 Hardware port configuration

(1) Input port

The VC-SM supports single-phase or AB-phase inputs as Primary signal sources. As shown in the table below

Hardware terminals	Single-phase counter	AB phase counting	Maximum input frequency
X0	C236	C256	200kHz
X1	C237		
X2	C238	C257	
X3	C239		
X4	C240	C258	
X5	C241		
X6	C242	C259	
X7	C243		

(2) Output port

The VC-SM supports eight 200kHz high-speed pulse outputs, of which the 4-axis electronic cam pulse output ports and directional ports are assigned as shown below.

Port	Shaft number	Pulse output mode selection			Output Frequency
		Single-phase pulse	Pulse + Direction	Positive pulse (CW) + negative pulse (CCW)	
Y0	Y0	Pulse	Pulse	Forward pulse (CW)	
Y1		/	Direction	Negative pulse (CCW)	
Y2	Y2	Pulse	Pulse	Forward pulse (CW)	

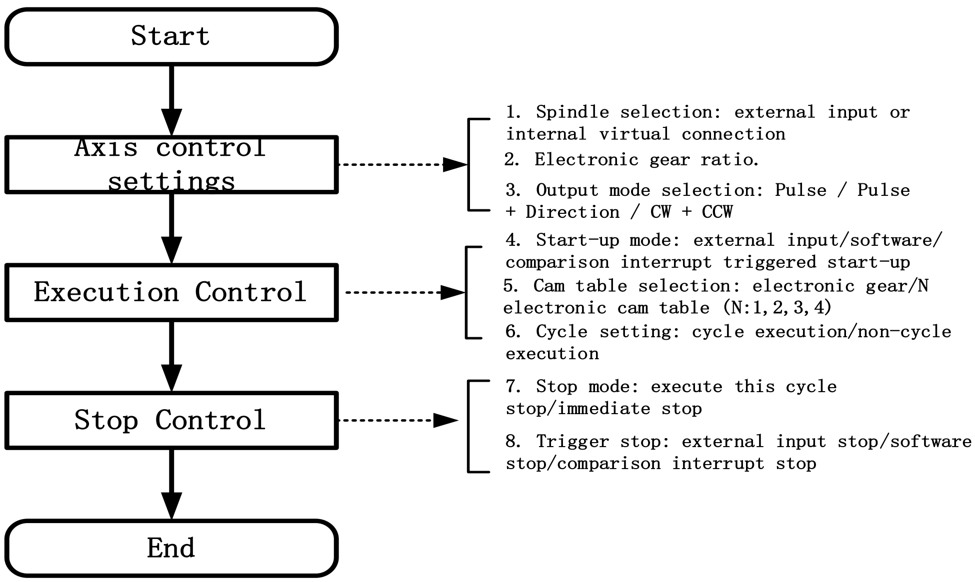
Y3		/	Direction	Negative pulse (CCW)	200kHz
Y4	Y4	Pulse	Pulse	Forward pulse (CW)	
Y5		/	Direction	Negative pulse (CCW)	
Y6	Y6	Pulse	Pulse	Forward pulse (CW)	
Y7		/	Direction	Negative pulse (CCW)	

Notes.

(1) The output port of the electronic cam is set by system default and cannot be modified.

(2) When the electronic cam output mode selects bit single-phase pulse (no direction), the released direction bit port can run pulse output control such as positioning command

12.1.3 Steps for using electronic cams

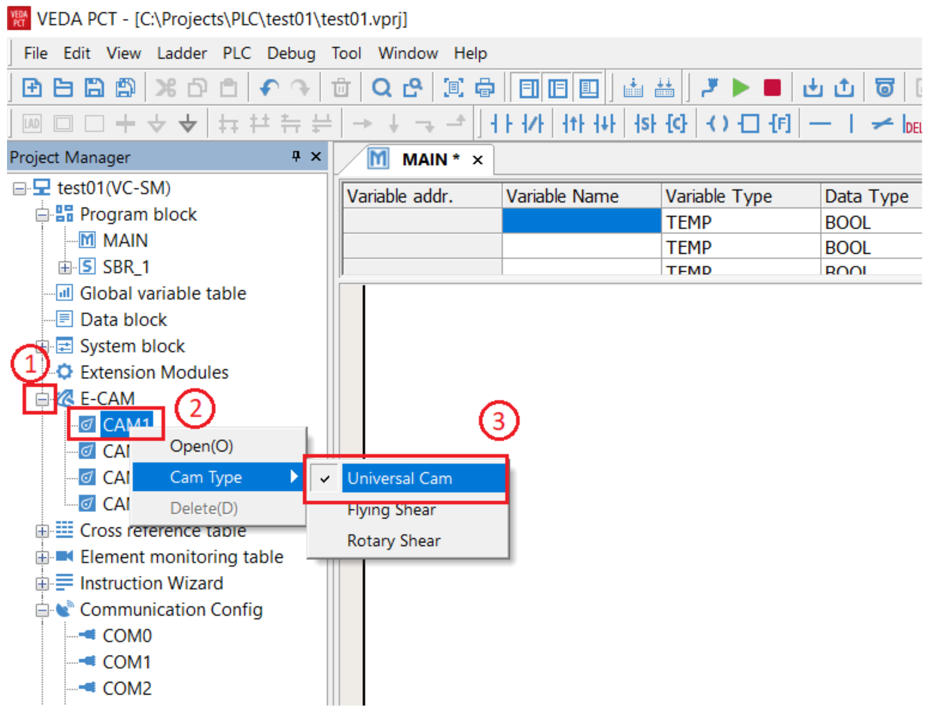


12.2 Create Cam Table

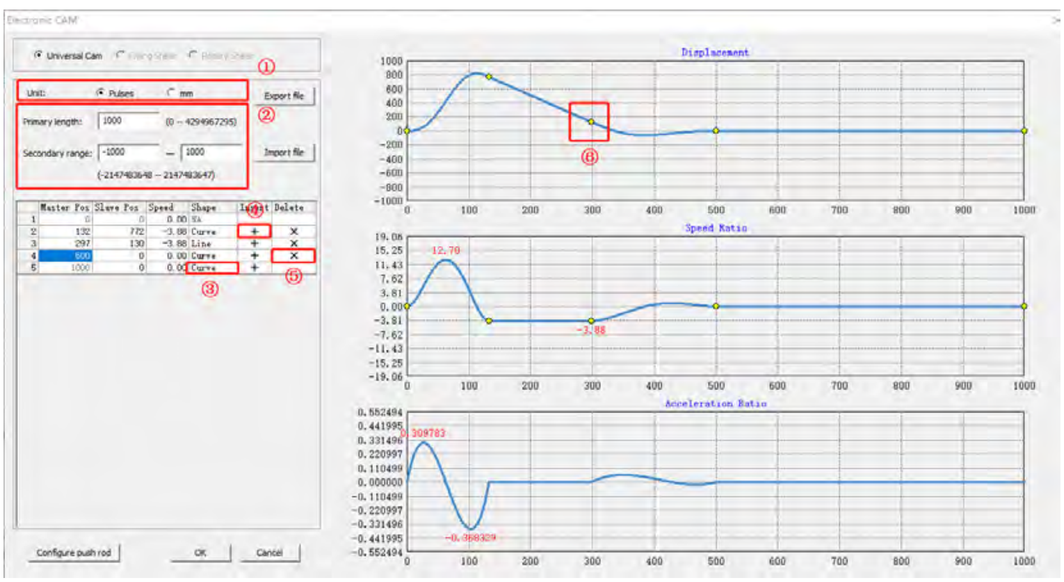
The essence of electronic cams is that the slave axis follows the motion of the main shaft. The relationship between the motion of the main shaft and the slave axis can be expressed in cam table data or electronic gear ratios. With electronic cam table data, a maximum of 360 key points of data can be created. With electronic gear ratios, only a fixed proportional relationship between the master and slave axes is sufficient.

12.2.1 Cam table type setting

A. Open VEDA PCT software, in the Project Manager, click on the electronic cam "+" The system automatically creates four electronic cam tables by default CAM1, CAM2, CAM3, CAM4 Users can right click on the cam table and select the type of cam table according to their needs, the default is a general purpose cam. (The types of cams are divided into: universal cam, flying shear, chasing shear) as shown below



B. Double-click the cam table CAM1, and edit the key point data of the cam table in the pop-up dialog box. As shown below:



① 【Unit】mm: The setting range of the length of the main axis is 0~10000, and the range of the slave axis is ±100000;

Pulse: The setting range of the length of the master axis: 0-4294967296, the range of the slave axis: -2147483648~2147483647;

② 【Primary length】: Indicates the Primary distance corresponding to the cycle of an electronic cam;

【Secondary range】: The setting is to display the graph for easy editing, and the setting can display the master position corresponding to the current position of the slave axis.

③ 【Shape】: You can set the fifth power curve or straight line interpolation fitting.

④ 【Insert】 Click "+" to add key points;

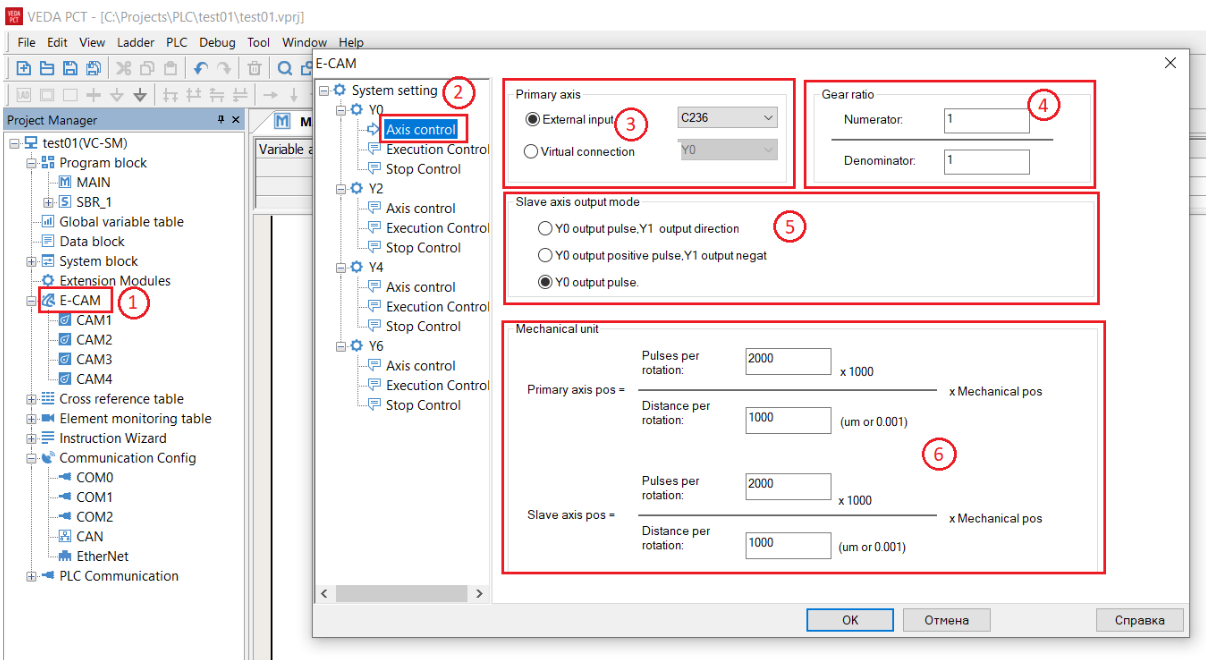
⑤ 【Delete】 Click "x" to delete the key point;

⑥ The position of the master axis and the slave axis can be modified by dragging the key points;

⑦ After completing the key settings, click "OK" to complete the cam table settings.

12.3 Primary Setting Selection

A. Double-click "electronic cam" to enter the electronic cam system configuration: as shown below:



① Double-click 【Electronic Cam】 to enter the electronic cam system configuration;

② Double-click 【Axis Control Settings】 to enter the parameter settings such as Primary selection, gear ratio setting, slave axis output mode, and mechanical unit;

③ 【Primary selection】 Primary signal source can choose external input or internal virtual connection;

④ 【Gear ratio】 According to the input signal, the output pulse is multiplied and down-frequency output;

⑤ 【Slave axis output mode】 Set pulse output, pulse + direction, positive and negative pulse output;

⑥ 【Mechanical unit】 When the electronic cam table CAM unit selects [mm], use this parameter to calculate the pulse position of the master axis and the slave axis.

The special soft components for Primary selection are as follows:

Primary first select SD component			
Y0 axis	Y2 axis	Y4 axis	Y6 axis

SD607	SD657	SD707	SD757
External input selection: 236~259 correspond to C236~C259; Internal virtual connection: 0~7 corresponds to Y0~Y7;			

The gear ratio special devices are as follows:

Gear ratio SD element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD604/SD605	SD654/SD655	SD704/SD705	SD754/SD755
Note: the denominator cannot be set to 0;			

The following table shows the special devices of slave axis output mode:

Slave output mode SD element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD609	SD659	SD709	SD759
SD values indicate the following meanings: 0: Y0 output pulse Y1 output direction signal, 1: Y0 outputs positive pulse, Y1 outputs negative pulse, 2: Only Y0 outputs pulses,			

The special devices of mechanical units are shown in the following table:

	Mechanical unit SD element			
Function	Y0 axis	Y2 axis	Y4 axis	Y6 axis
The number of pulses in one revolution of the Primary motor	SD616/SD617	SD666/SD667	SD716/SD717	SD766/SD767
Primary motor rotates one circle movement distance	SD618/SD619	SD668/SD669	SD718/SD719	SD768/SD769
The number of pulses per revolution of the slave motor	SD600/SD601	SD660/SD661	SD700/SD701	SD750/SD751
The movement distance of one rotation of the slave axis motor	SD602/SD603	SD662/SD663	SD702/SD703	SD752/SD753
Note: This parameter only participates in the calculation when the cam table unit is selected as the millimeter unit.				

12.4 Periodic/Aperiodic Selection

The electronic cam can be executed periodically or aperiodically, which is set by special SM and SD elements. Periodic /Aperiodic selection uses special components as follows:

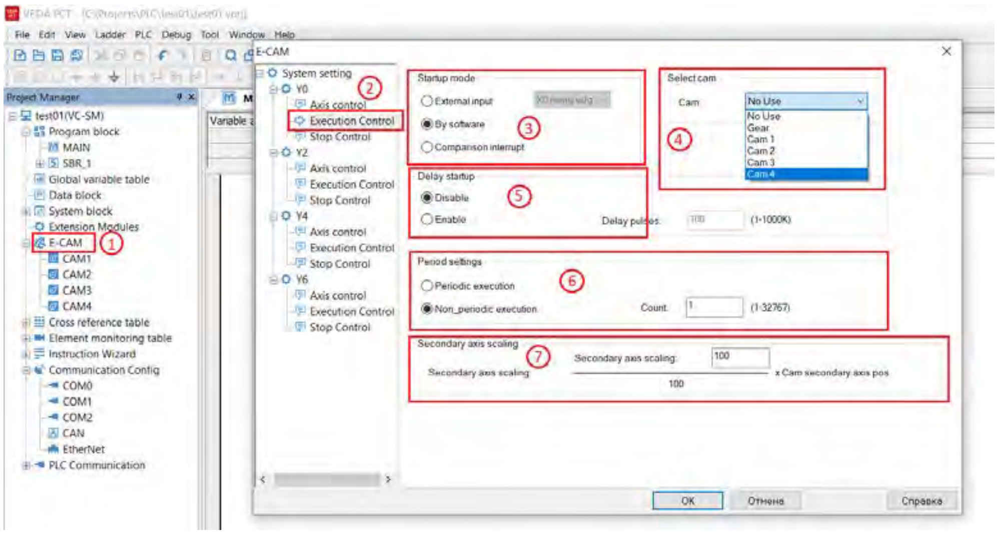
Periodic/Aperiodic setting SD			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD608	SD658	SD708	SD758
SD values indicate the following meanings: The default 0 means that the electronic cam cycle is executed Setting non-0 means the electronic cam aperiodic electronic cam execution times, the maximum can be set to 255 cycles			

Periodic execution: After the electronic cam is started, the relationship set by the electronic cam table is executed continuously and periodically until a stop command is received;

Aperiodic execution: After the electronic cam is started, it will automatically stop after executing the set period. The number of periods of aperiodic execution is determined by SDelement(SD608,SD658,SD708,SD758)setting, the maximum can be set255 cycle.

12.5 Startup Mode Settings

Double-click "Electronic Cam" and select "Execution Control" to set parameters such as start mode, cam table, delayed start, cycle setting, and slave axis scaling as shown in the figure below:



① Double-click 【Electronic Cam】 to enter the electronic cam system parameter configuration;

② Select 【Execution Control】 to set parameters such as start mode, cam table, delayed start, cycle setting, and slave axis scaling;

③ 【Startup mode】 The starting mode of electronic gear and electronic cam can be selected as external edge trigger, software start and comparison interrupt trigger start;

④ 【Select Cam】 Use this function to select the output mode; electronic gear, 1~4 cam table, the default selection does not use electronic cam and gear.

⑤ 【Delay startup】 Start the electronic cam or gear after delaying the number of pulses set by the Primary;

⑥ 【Period setting】 Periodic execution means: cyclic execution of the cam table, non-periodic execution means: stop the output after the execution is completed according to the set number of cycles.

⑦ 【Secondary axis scaling】 The electronic cam can realize the scaling of the cam table through the scaling function.

12.5.1 Boot mode settings

Electronic cam/electronic gear start is divided into three start modes;

① External edge trigger rising edge/falling edge: When the cam enable SM is ON, when the external input X point signal has a rising or falling edge, the electronic cam or electronic gear is activated.

② Software start: when the cam enable SM is ON, start the electronic cam or electronic gear.

③ Comparison interrupt trigger start: when the cam enable SM is ON, when the number of Primary pulses is equal to the value set by the DHSCS instruction, the electronic cam or electronic gear is started.

④

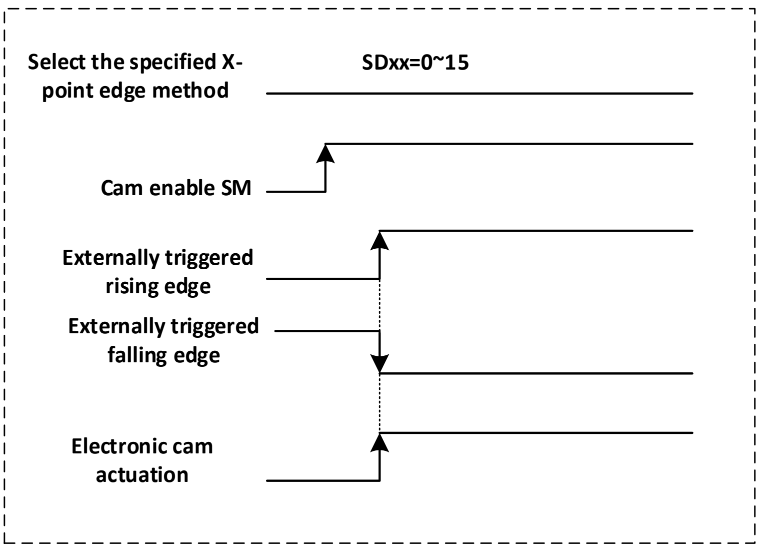
The cam enable SM components are as follows

Cam enable SM element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SM600	SM620	SM640	SM660

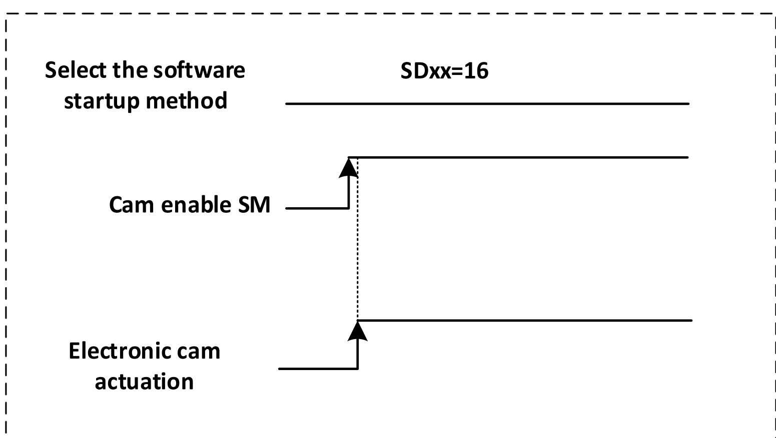
Boot Mode Select SD Register

Startup mode selection SD element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD628	SD678	SD728	SD778
External trigger start	0~7 means to use the rising edge of X0~X7 as the start signal; 8~15 means use the falling edge of X0~X7 as the start signal;		
Software start	16 means using the software startup method;		
Compare interrupt start	17 means using the comparison interrupt start mode;		

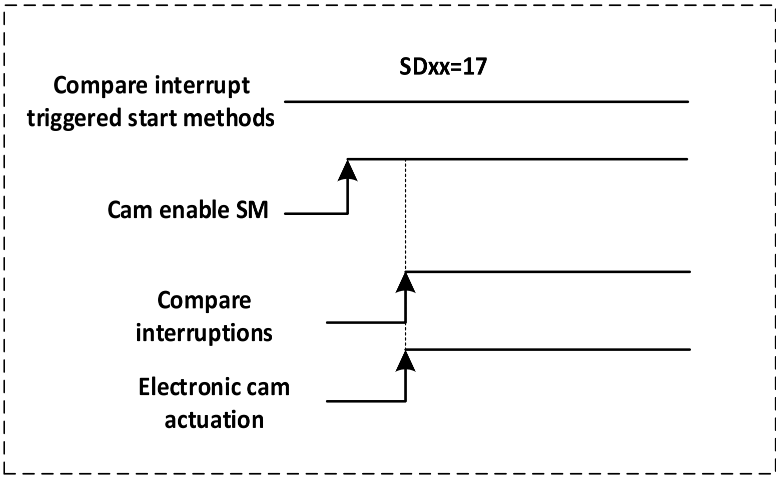
External edge trigger start timing:



Software startup sequence:



Compare interrupt startup:

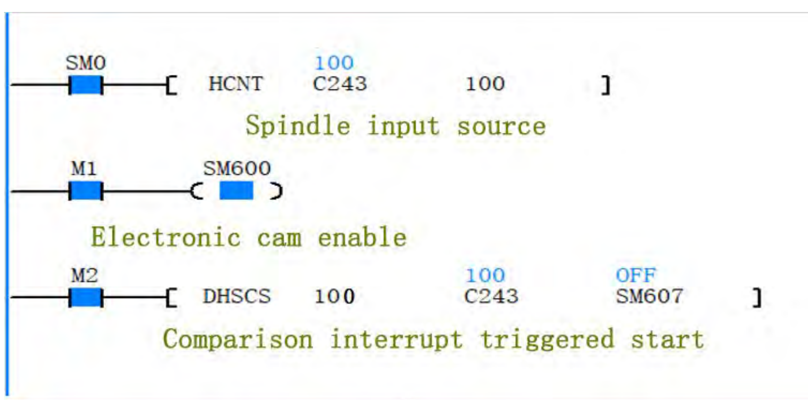


When the electronic cam/electronic gear is selected to start when the comparison interrupt is triggered, in conjunction with the use of the comparison command, the electronic cam/electronic gear is triggered and started by the comparison interrupt.

The comparison interrupt special SM components are as follows

Compare interrupt trigger SM element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SM607	SM627	SM647	SM667
Function: Set SM element through high-speed counting comparison interrupt, start electronic cam or electronic gear (1) It needs to be used in conjunction with HCNT and DHSCS instructions; (2) After the DHSCS count comparison arrives, the SM element will be set, and the software will automatically turn off when the electronic cam or electronic gear is activated;			

Example of use (take Y0 as an example)



Program description: M1, M2 are set, when the Primary input pulse is 100, DHSCS will set SM607, start the electronic cam or electronic gear.

12.5.2 Cam table/electronic gear selection

By setting different cam table to select SD element value, different cam table or electronic gear execution can be selected.

The special components used for cam table selection are as follows:

Cam table selection SD element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD606	SD656	SD706	SD756
Do not use	SD value of 0: Indicates that the electronic cam or electronic gear function is not used		
electronic gear	SD value of 10: indicates the use of the electronic gear function		
Cam 1	SD value of 11: means using CAM1 cam table		
Cam 2	SD value of 12: means using CAM2 cam table		
Cam 3	SD value is 13: means using CAM3 cam table		
Cam 4	SD value of 14: means using CAM4 cam table		

12.5.3 Delay start setting

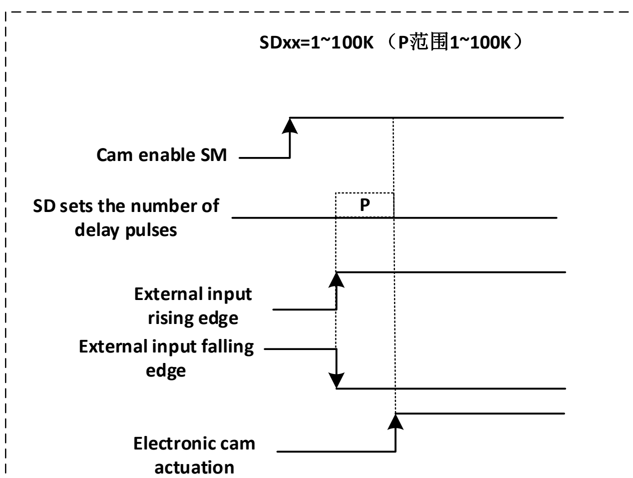
The electronic cam/electronic gear can realize the delay start function according to the delay start setting. Delay start function is triggered by external input and software start.

After that, the electronic cam will be executed after delaying the set number of Primary pulses.

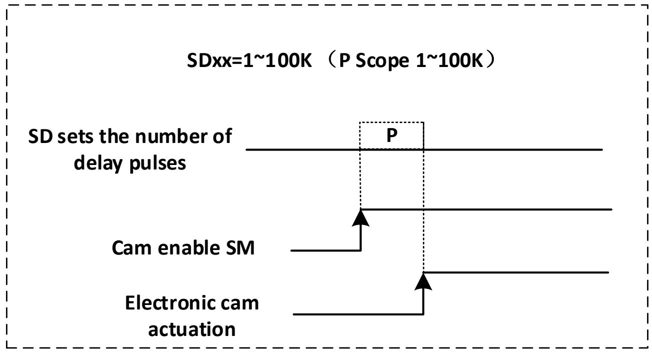
Delay start setting special components table:

Delay start SD element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD610	SD660	SD710	SD760
SD register value: 0 means that startup is prohibited; SD register value: 1~100K means that the delay function is enabled according to the SD register setting value;			

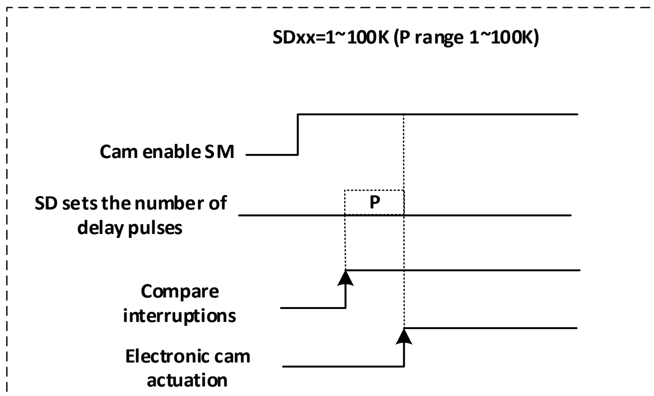
External input trigger delay start sequence:



Software delay start sequence:

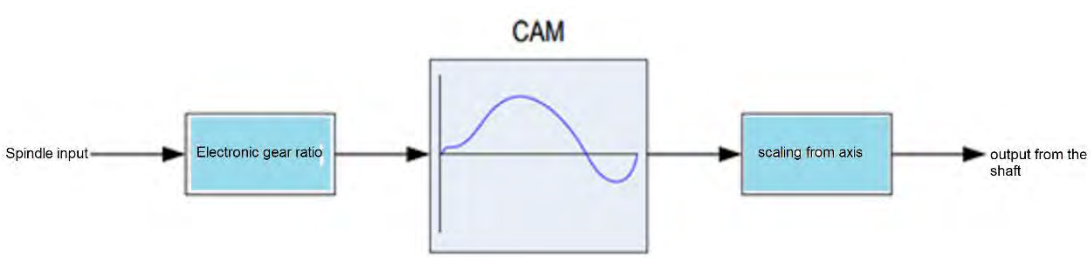


Compare interrupt delay start sequence:

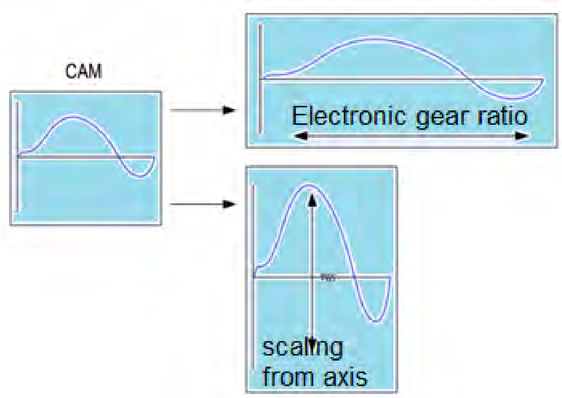


12.6 Scaling

The electronic cam can realize the scaling of the cam table through the scaling function.



The scaling of the master axis or the slave axis is realized by setting the electronic gear ratio and the special SD element of the slave axis scaling ratio.



The special components used by the electronic gear ratio and the slave axis scaling ratio are as follows:

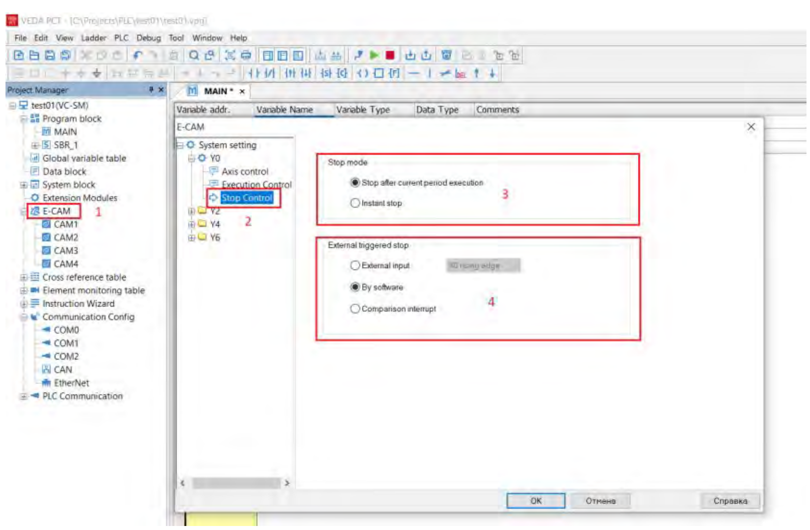
Electronic gear ratio/Primary scaling				Scaling from axis			
Y0 axis	Y2 axis	Y4 axis	Y6 axis	Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD604/SD605	SD654/SD655	SD704/SD705	SD754/SD755	SD620/100	SD670/100	SD720/100	SD770/100
Primary scaling				The scaling ratio of the slave axis, the default value is 100 when the SD component system is started, that is, the scale value is 1			

After the electronic gear ratio and the scaling ratio of the slave axis are changed, they will take effect the next time the cam is started by default. If it needs to take effect in the currently running cam, it is necessary to set the cam table data modification special SMor Scomponent, it can take effect in the next cam cycle of the current operation, after the cam table data modification special SM orS Components reset automatically.

	Y0axis	Y2axis	Y4axis	Y6axis	Illustrate
Cam table data modification	SM603	SM623	SM643	SM663	Automatically reset to OFF after data modification takes effect

12.7 Stop Mode Setting

Stop mode setting: as shown below:



① Double-click 【Electronic Cam】 to enter the electronic cam or electronic gear parameter configuration;

② Double-click 【Stop Control】 to enter stop mode and trigger stop mode settings;

③ 【Stop mode】Stop after the current cycle is executed: When the electronic cam is executed, when the cam enable is turned OFF or the stop signal is valid, the electronic cam will stop after executing the currently executing cycle. Immediate stop: When the electronic cam is executed, when the cam enable becomes OFF or the stop signal is valid, the electronic cam stops immediately.

④ 【External trigger stop】The electronic cam/electronic gear can be stopped in three ways during the execution process: external input trigger stop, software stop and comparison interrupt trigger stop.

Note: When stopping by any of the above methods, if the electronic gear function is being executed, the electronic gear function will be stopped immediately; if the electronic cam function is being executed

Yes, according to the setting of the stop mode, the execution of the current cycle is stopped or the execution of the electronic cam is stopped immediately

12.7.1 Stop mode setting

The electronic cam can set two stop modes through the special SM element, choose to stop after the current cycle or stop immediately.

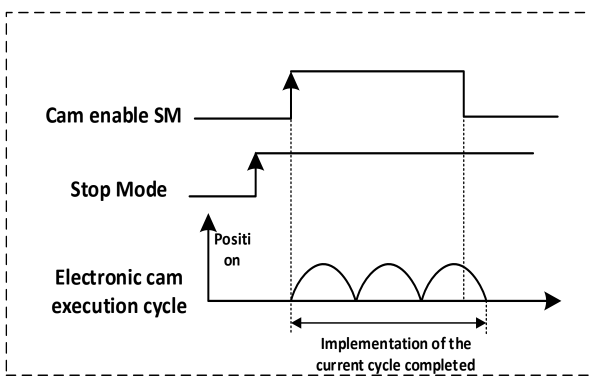
The stop mode setting uses special components as shown in the table below Stop mode SM element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SM605	SM625	SM645	SM665
ON: stop after executing this cycle; OFF: stop immediately;			

(1) Stop this cycle after the execution of the electronic cam: when the electronic cam is executed, the cam enable is turned OFF or the stop signal is valid, the electronic cam is currently executing after the execution

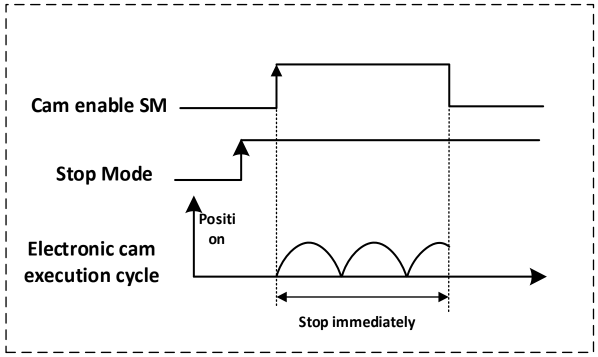
stop after the cycle.

(2) Immediate stop: When the electronic cam is executed, when the cam enable becomes OFF or the stop signal is valid, the electronic cam stops immediately.

Execute this cycle stop sequence:



Immediately stop the timing sequence:



12.7.2 Trigger stop setting

When the electronic cam/electronic gear is being executed, there are three stop modes for trigger stop setting: external input trigger stop：software stop and comparison interrupt trigger stop.

(1) External input trigger stop：use external input to trigger stop;

(2) Stop the software; set the cam enable element SM to OFF;

(3) Comparison interrupt trigger stop: use the comparison interrupt DHSCS instruction to trigger stop;

Note: When stopping by any of the above methods, if the electronic gear function is being executed, the electronic gear function will be stopped immediately; if the electronic cam function is being executed

Yes, according to the setting of the stop mode, the execution of the current cycle is stopped or the execution of the electronic cam is stopped immediately.

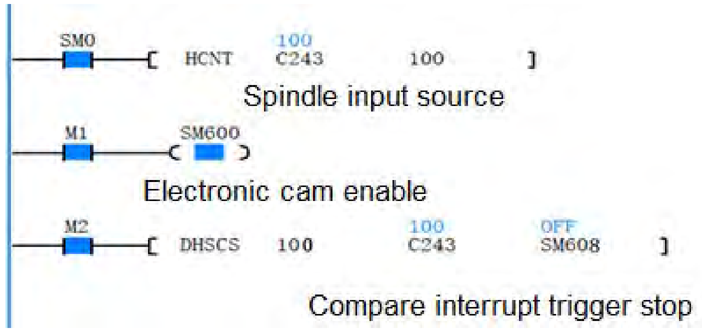
Trigger stop mode is selected by special SD as shown in the table below:

Trigger stop mode SD element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SD629	SD679	SD729	SD779
External trigger stop	0~7 means to use the rising edge of X0~X7 as the stop signal; 8~15 means use the falling edge of X0~X7 as the stop signal;		
Software stop	16 means using the software stop method;		
Compare interrupt stop	17 means using the comparison interrupt stop method;		

Compare interrupt triggers stop SM element

Compare interrupt trigger SM element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SM608	SM628	SM648	SM668
Function: Set SM element through high-speed counting comparison interrupt, stop electronic cam or electronic gear (1) It needs to be used in conjunction with HCNT and DHSCS instructions; (2) After the DHSCS count comparison arrives, the SM element will be set, and the software will automatically turn off when the electronic cam or electronic gear stops;			

Example of use (take Y0 as an example) Trigger stop using compare interrupt



Program description: M1 turns ON to enable the electronic cam, and when the C243 high-speed counter counts to 100, the electronic cam is triggered to stop.

12.7.3 Cycle complete Flag

Every time the electronic cam completes a cycle, the system automatically sets the cycle completion Sign special SM element to ON. After the cycle complete Sign is set, it remains

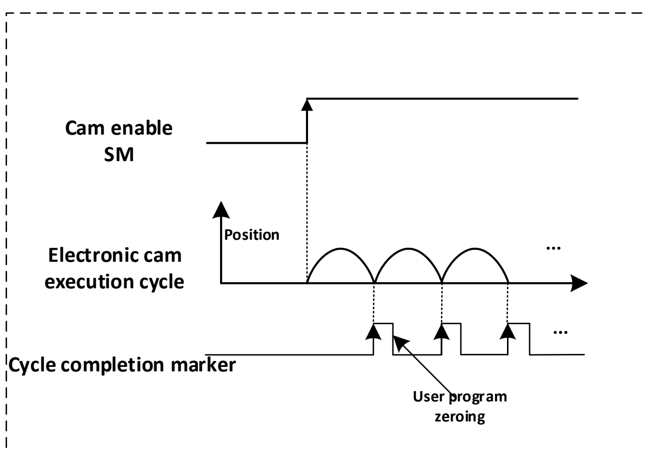
In the ON state, if the completion of the next cycle needs to be detected, the user program needs to clear the cycle completion Sign to OFF and complete the next cycle.

, the system sets the cycle completion Sign to ON again.

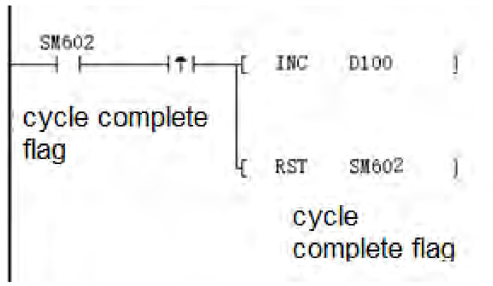
The electronic cam cycle completion Sign SM element is shown in the following table:

Cam cycle complete SM element			
Y0 axis	Y2 axis	Y4 axis	Y6 axis
SM602	SM622	SM642	SM662
ON: The SM element will be turned ON when each cam cycle is completed;			

Cycle Completion Sign Timing:

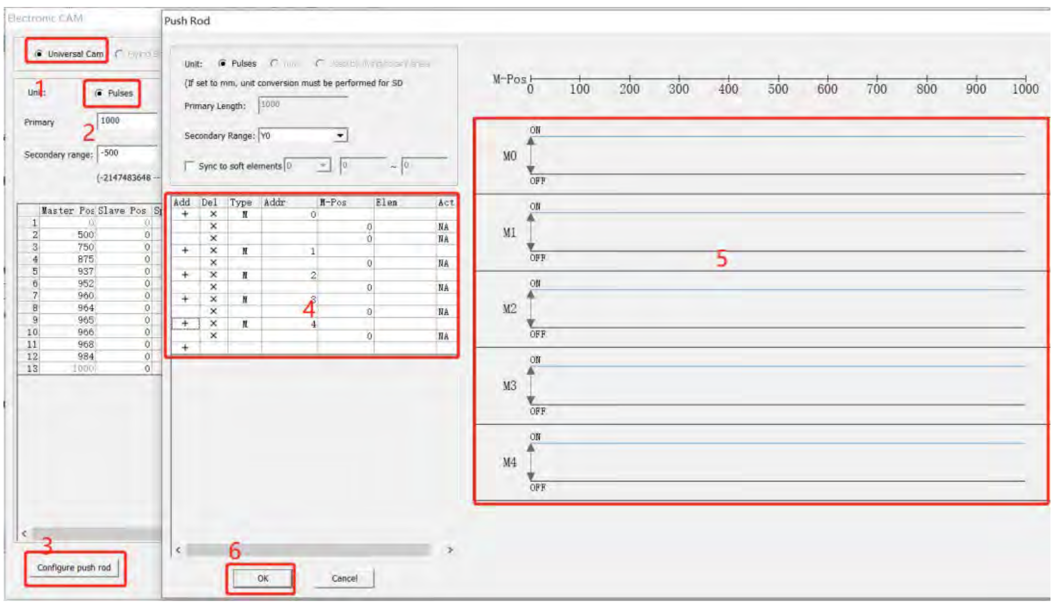


Example of use (take Y0 as an example)



12.8 Ejector Settings

Definition of ejector bar: The ejector bar function can realize the cooperation between the bit element (M, Y) and the position of the electronic cam Primary, and control the ON/OFF change of the bit element with the change of the Primary position. The top rods are set up as follows:



① Double-click the cam table CAM1 to enter the cam table parameter setting;

② The unit can choose pulse or millimeter;

③ Double-click 【Configure push rod】 to enter ejector interface configuration;

④ Ejector data table setting parameters:

【Add】 Click "+" to add ejector;

【Del】 Click "×" to delete the ejector rod;

【Type】 Set the bit element type, support M element and Y element;

【Addr】 Set the M or Y component address label;

【M-Pos】 Set the ejector trigger condition;

[Elem] When "Synchronize to software" is checked, the initial D component label will be displayed. If it is not checked, it will not be displayed.

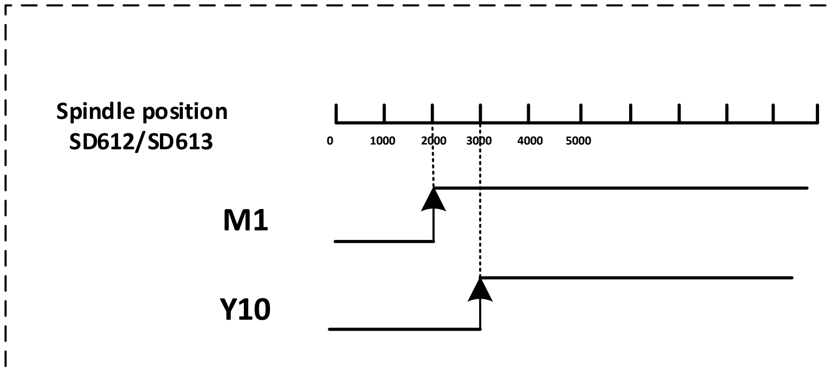
【Action】 The action of the component when the Primary position is equal to the M-Pos setting value; NA means no action; ON means ON; OFF means OFF; INV means inversion;

【Slave axis attribute】 indicates the axis attribute, that is, when the electronic cam slave axis and the axis attribute are the same, the ejector data is valid. Y0/Y2/Y4/Y6 can be selected;

⑤ After the ejector bar is set, the ejector bar graph is displayed;

⑥ After the configuration is complete, click OK to complete the ejector configuration;

The above table sets the ejector data image as shown below:



12.9 Electronic Cam Key Point Modification

Among the 4 CAM tables created by the program, the user can use the electronic cam command to read or modify the data in the CAM table in the program. The command format is as follows

Electronic cam command	Camwr write electronic cam data	Pulse unit to modify key point data
	Camrd read electronic cam data	Pulse unit to modify key point data
	Ecamwr write electronic cam floating point data	Mechanical units modify keypoint data
	Ecamrd read electronic cam floating point data	Mechanical units modify keypoint data

12.9.1 CAMWR writes electronic cam data

CAMWR: Write the electronic cam data (modify the electronic cam table of the pulse unit)

Ladder Diagram:	├─┤├─[CAMWR S1 S2 D1 D2]					**Applicable models**	**VC-S VC-P**
						Affect the flag	
Instruction list: CAMWR S1 S2 D1 D2						**Step size**	**13**
Operand	**type**	**Applicable devices**					**Index**
S1	**DINT**	**D**		**Constant**	**R**		
S2	**DINT**	**D**		**Constant**	**R**		
D1	**DINT**	**D**	**V**	**Constant**			**√**
D2	**DINT**	**D**		**Constant**	**R**		

- **Operand Description**

[S1] Specifies the cam table to be modified. S1=0~3 means: CAM1, CAM2, CAM3, CAM4;

[S2] Set the starting point of the cam table data to be modified. The range of key points is 2 to 360.

[D1] The data storage address to be modified occupies multiple consecutive address units starting from D1. Each key point occupies 2 32-bit registers to respectively mark the position of the master axis and the slave axis, that is, each key point needs to occupy 4 address units.

[D2] The number of key point data to be modified. S2+D2-1 must be less than or equal to the number of downloaded key points.

- **Precautions**

1. The CAMWR instruction can only be executed one at a time. If more than two CAMWR instructions are required in the program, the next instruction can only be started after the previous instruction stops for one scan cycle.

2. CAMWR is a multi-cycle execution instruction.

3. After the CAMWR modification is completed, it only changes the value of the memory. To take effect in the next cycle, there is no need to set the special SM element.

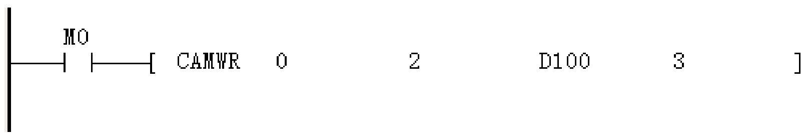
4. The first point of the electronic cam table is the starting point data and cannot be modified, so S2 must be greater than 1; the command parameter S2+D2-1 must be less than or equal to the number of downloaded key points.

5. When modifying the cam table data, the Primary position data must be greater than the Primary position of the previous point and smaller than the Primary position data of the latter point, otherwise an error will occur.

6. The CAMWR instruction can only modify the pulse unit cam table.

7. The CAMWR execution condition needs to be modified before half of the cam running in the current cycle. If the cam table is modified at the end of the current cycle, it is possible that the modification will take effect in the next cycle.

- **Example of use**



Program Description:M10 is set to ON, PLC executes the CAMWR instruction to start modifying the cam table data, the master and slave axes of each key point occupy 32bit registers, that is, one key point occupies 4 D elements. If the data of 3 points is modified, starting from the 2nd point, (D101, D100) represent the position of the main axis of the 2nd point, (D103, D102) represent the position of the slave axis of the 2nd point, and so on, occupying a total of 12 D elements.

12.9.2 ECAMWR writes electronic cam floating point data

ECAMWR writes electronic cam floating point data (modifies millimeter unit electronic cam table)

Ladder Diagram:	ECAMWR S1 S2 D1 D2					Applicable models	VC-S VC-P
						Affect the flag	
Instruction list: ECAMWR S1 S2 D1 D2						Step size	13
Operand	type	Applicable devices					Index
S1	DINT	D		Constant	R		
S2	DINT	D		Constant	R		
D1	DINT	D	V	Constant			√
D2	DINT	D		Constant	R		

- **Operand Description**

[S1]: Specify the cam table to be modified. S1=0~3 means: CAM1, CAM2, CAM3, CAM4;

[S2]: Set the starting point of the cam table data to be modified. The range of key points is 2 to 360.

[D1]: The storage address of the floating-point number to be modified occupies multiple consecutive address units starting from D1. Each key point occupies 2 32-bit registers to respectively mark the position of the master axis and the slave axis, that is, each key point needs to occupy 4 address units.

[D2]: The number of key point data to be modified. S2+D2-1 must be less than or equal to the number of downloaded key points.

- **Precautions**

(1) ECAMWR is a floating point type electronic cam data modification, corresponding to the mechanical unit electronic cam table data modification.

(2) The ECAMWR instruction is the same as the CAMWR instruction except that the operation data is interpreted as a floating point number.

12.9.3 CAMRD reads electronic cam integer data

CAMRD reads electronic cam integer data (reads electronic cam table in pulse unit)

Ladder Diagram:	CAMRD S1 S2 D1 D2					Applicable models	VC-S VC-P
						Affect the flag	
Instruction list: ECAMWR S1 S2 D1 D2						Step size	13
Operand	type	Applicable devices					Index
S1	DINT	D		Constant	R		
S2	DINT	D		Constant	R		
D1	DINT	D	V	Constant			√
D2	DINT	D		Constant	R		

- **Operand Description**

[S1] Specifies the cam table to be read. S1=0~3 means: CAM1, CAM2, CAM3, CAM4;

[S2] Set the starting point to read the cam table data. The range of key points is 2 to 360.

[D1] stores the read cam table data address, occupying multiple consecutive address units starting from D1. Each key point occupies 2 32-bit registers to respectively mark the position of the master axis and the slave axis, that is, each key point needs to occupy 4 address units.

[D2] The number of key point data to be read. S2+D2-1 must be less than or equal to the number of downloaded key points.

- **Precautions**

(1) CAMRD reads the pulse unit electronic cam table data, and the cam table specified to be read must exist in the PLC system, that is, the cam table has been passed through.

Download it to the PLC system through VEDA PCT.

(2) The command parameter S2+D2-1 must be less than or equal to the number of downloaded key points.

- **Example of use**

```
     M0
|----| |----[ CAMRD   0       2       D500     3       ]
```

Program Description:M10 is turned ON, PLC executes the CAMRD instruction to start reading the cam table data, and the read data is stored in the D element starting from D500. The master and slave axes of each key point occupy 32bit registers, that is, one key point occupies 4 D element. If the data of 3 points are read, starting from the 2nd point, (D501, D500) indicate the position of the master axis of the second point, (D503, D502) indicate the position of the slave axis of the second point, and so on, a total of Occupies 12 D elements.

12.9.4 ECAMRD reads electronic cam floating point data

ECAMRD Read electronic cam floating point data (read electronic cam table in millimeters)

Ladder Diagram:	`|--| |--[ ECAMRD S1 S2 D1 D2 ]`					**Applicable models**	**VC-S VC-P**
						Affect the flag	
Instruction list: ECAMRD S1 S2 D1 D2						**Step size**	**13**
Operand	**type**	**Applicable devices**					**Index**
S1	**DINT**	**D**		**Constant**	**R**		
S2	**DINT**	**D**		**Constant**	**R**		
D1	**DINT**	**D**	**V**	**Constant**			**√**
D2	**DINT**	**D**		**Constant**	**R**		

- **Operand Description**

[S1] Specifies the cam table to be read. S1=0~3 means: CAM1, CAM2, CAM3, CAM4;

[S2] Set the starting point to read the cam table data. The range of key points is 2 to 360.

[D1] Store the read cam table data address, occupying multiple consecutive address units starting from D1. Each key point occupies 2 32-bit registers to respectively mark the position of the master axis and the slave axis, that is, each key point needs to occupy 4 address units.

[D2] The number of key point data to be read. S2+D2-1 must be less than or equal to the number of downloaded key points.

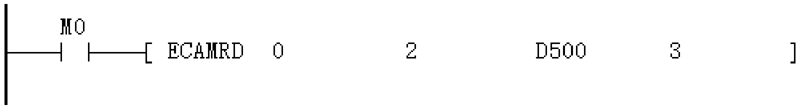
- **Precautions**

(1) ECAMRD reads the electronic cam table data in millimeter units, and the cam table specified to be read must exist in the PLC system, that is, the cam table has been passed through.

Download it to the PLC system through VEDA PCT.

(2) The command parameter S2+D2-1 must be less than or equal to the number of downloaded key points.

- **Example of use**



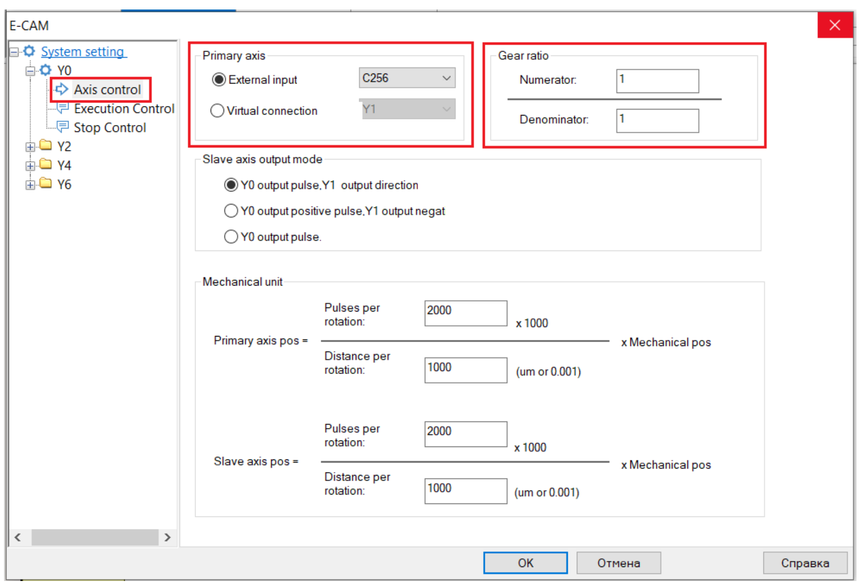
Program Description:M10 turns ON, PLC executes the ECAMRD instruction to start reading the cam table data, and the read data is stored in the D element starting from D500. The master and slave axes of each key point occupy 32bit registers, that is, one key point occupies 4 D registers. element. If the data of 3 points are read, starting from the 2nd point, (D501, D500) indicate the position of the master axis of the second point, (D503, D502) indicate the position of the slave axis of the second point, and so on, a total of Occupies 12 D elements.

12.10 Application Examples

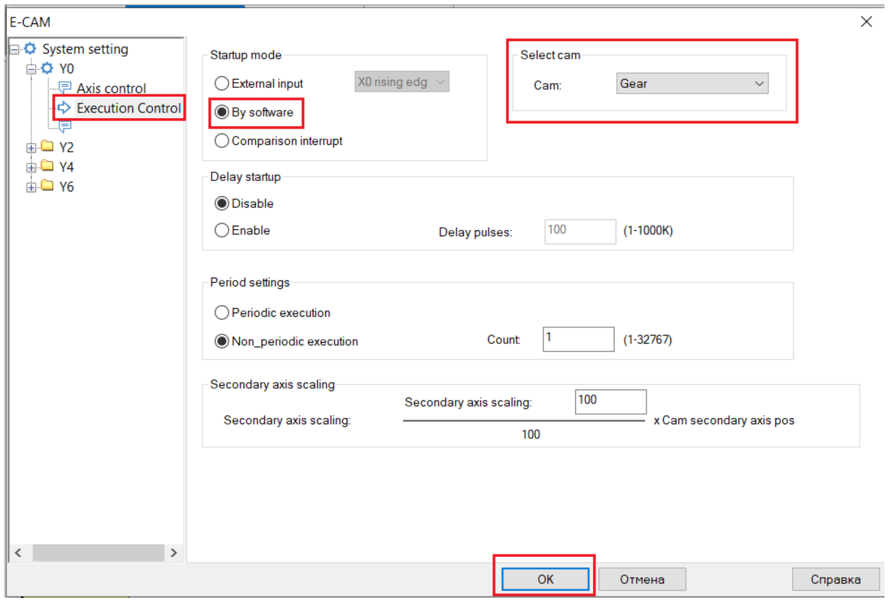
12.10.1 Example of electronic gear:

Control requirements: control the Y0 axis (servo axis) to follow the Primary encoder 1:1 to run synchronously.

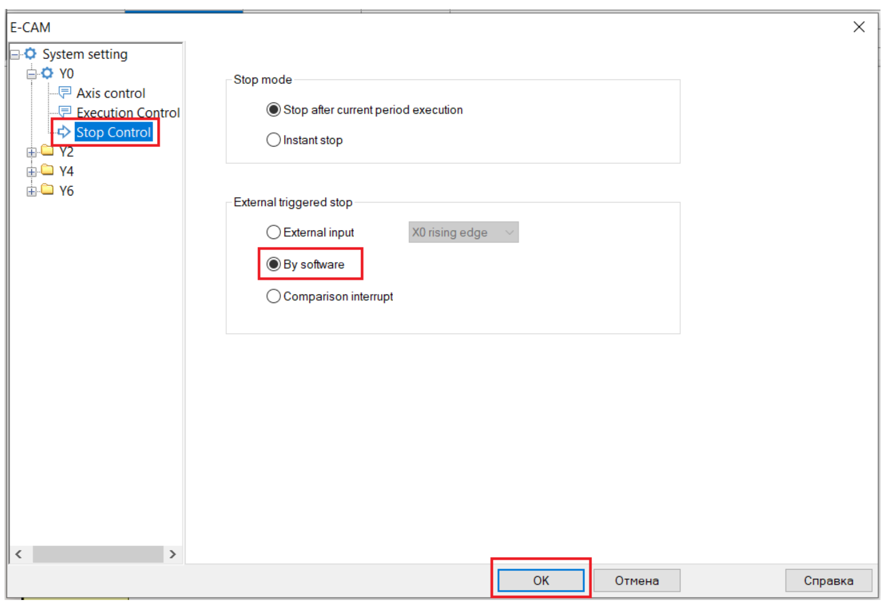
1) Set the external input of the Primary source C256 (X0/X1), and set the electronic gear ratio to 1:1; as shown in the figure below



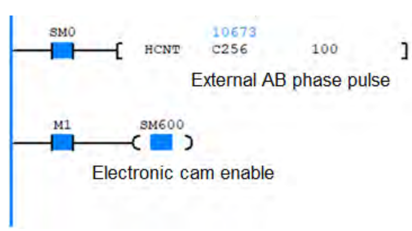
2) The execution control startup mode is software【software startup】, the cam table selects【electronic gear】, and the others remain default.



3) Select the stop method to stop the software, click 【OK】 to complete the setting of the electronic gear



4) Program programming as follows

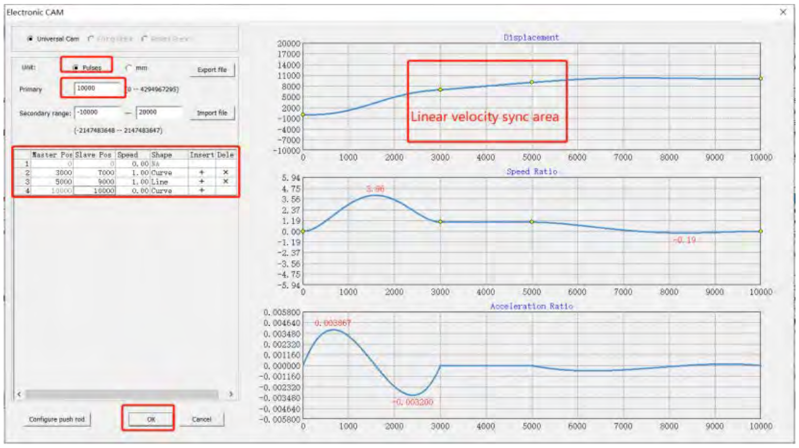


5) Program description: Turn M1 ON, enable the Y0 axis electronic gear, and the slave axis starts to mesh with the master axis. When the master axis pulses input pulses, the slave axis follows the master axis according to the electronic gear ratio of 1:1.

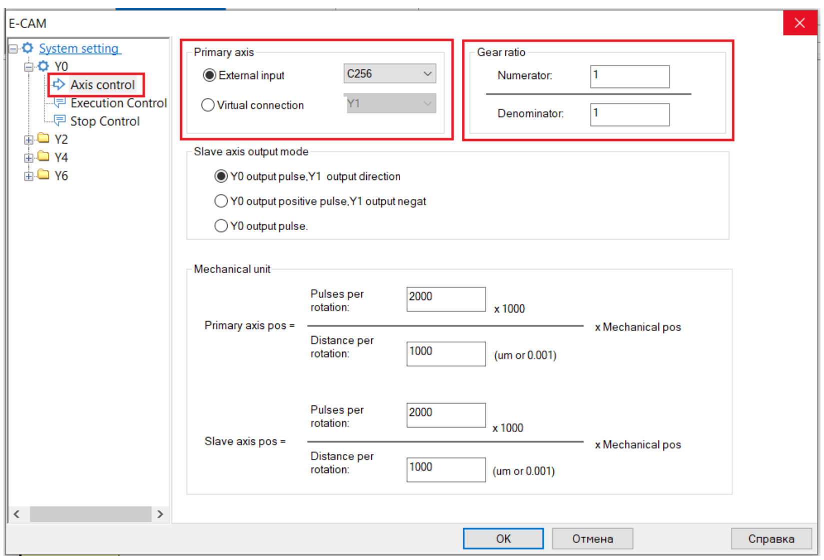
12.10.2 Electronic cam example

Control requirements: use CAM1 cam table, the unit of cam table is pulse, the length of the main axis is 10000, and the length of the slave axis is 10000; control the Y0 axis (servo axis), when the main axis position is in the range of 3000-5000, the line speed synchronization function is realized, and other positions are set according to the setting Fixed curve operation.

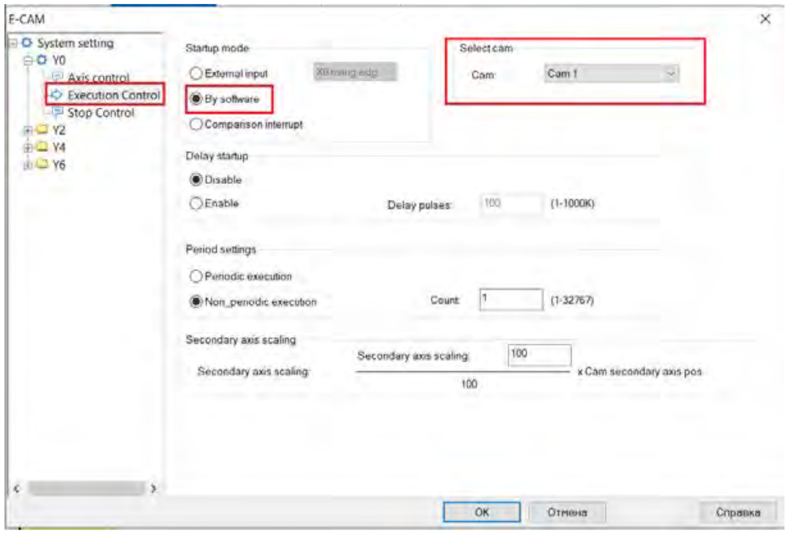
1) The CAM1 cam table settings are shown below:



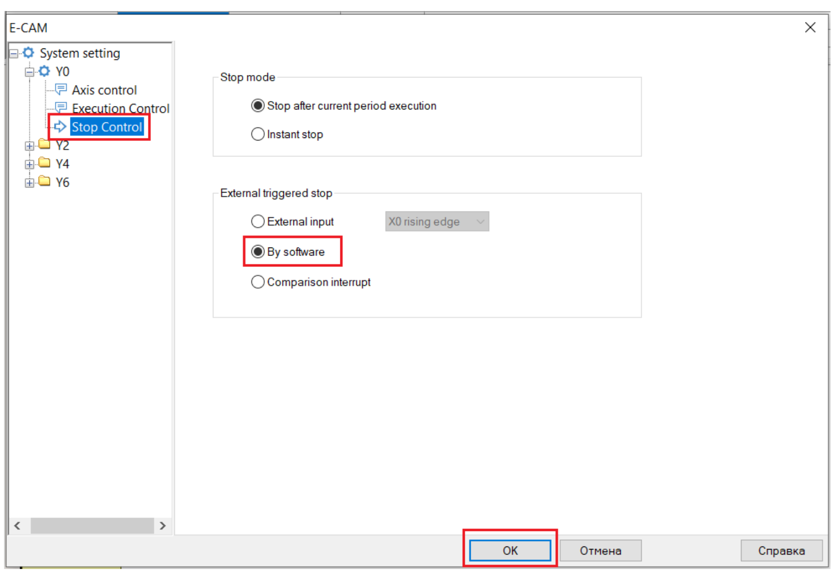
2) Set the external input of the Primary source C256 (X0/X1), and set the electronic gear ratio to 1:1; as shown in the figure below



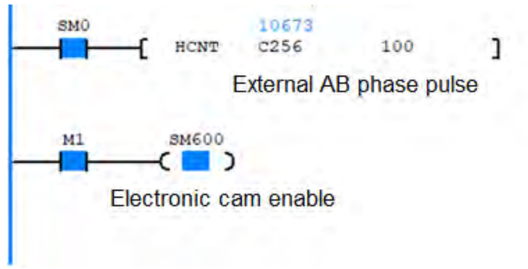
3) The execution control startup mode is 【By software】, the cam table selects【No. 1 electronic cam】, and the others keep the default.



4) Select the stop method to stop the software, click 【OK】 to complete the electronic cam setting



5) Program programming: as follows



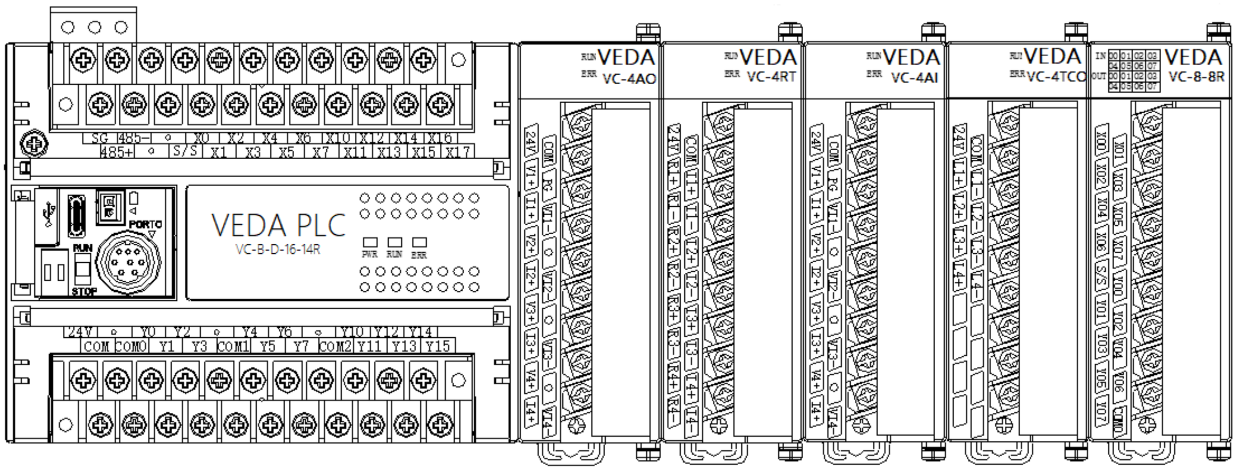
6) Program description: Turn M1 ON, enable the Y0 axis electronic gear, and the slave axis starts to mesh with the master axis. When the master axis pulses input pulses, the slave axis runs according to the track cycle set by the cam table CAM1. When M1 is turned OFF, the electronic cam Stop running after this cycle is executed.

Chapter 13 Expansion Module

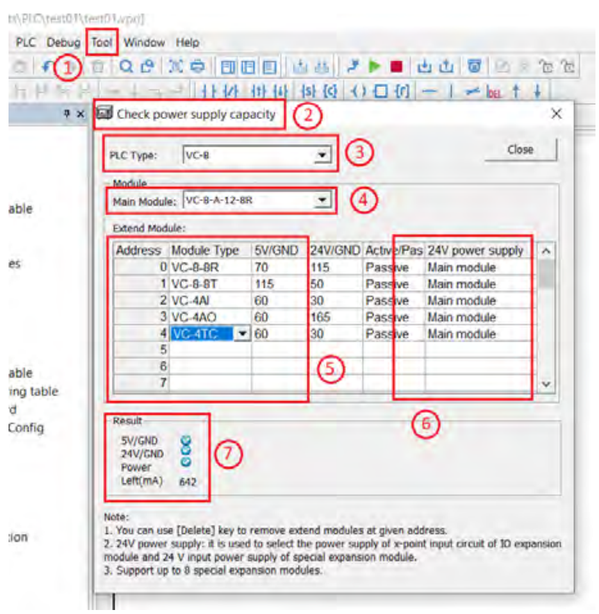
13.1 Overview of Expansion Modules

The VEDA VC series main module PLC can be configured through the expansion module to realize the control of the expansion module.

(1) VEDA VC series module expansion configuration example



A. VEDA VC series PLC can be extended with up to 15 modules, of which only 8 special modules can be configured at most. The number of configurable extensions for different main modules is different, which can be confirmed by calculating the power supply capacity in the VEDA PCT host computer software. As shown below



① Open VEDA PCT software, click [Tools] under the menu bar;

② Select [Power Capacity Calculation] to pop up the power capacity calculation box;

③ Select PLC series type;

④ Select the corresponding PLC main module model;

⑤ Click the blank space under "Module Type" to select the module type to be added; (click [Delete] to delete the configured module)

⑥ 24V power supply: [External] means that the 24V of the expansion module is supplied by the external 24V power supply; [Main module] means that the 24V of the expansion module is supplied by the PLC body with 24V power supply; due to the limited load capacity of the 24V provided by the PLC body, when selecting the main module body When the power supply is 24V, the number of connected expansion modules will be reduced. It is recommended to use an external 24V power supply for power supply.

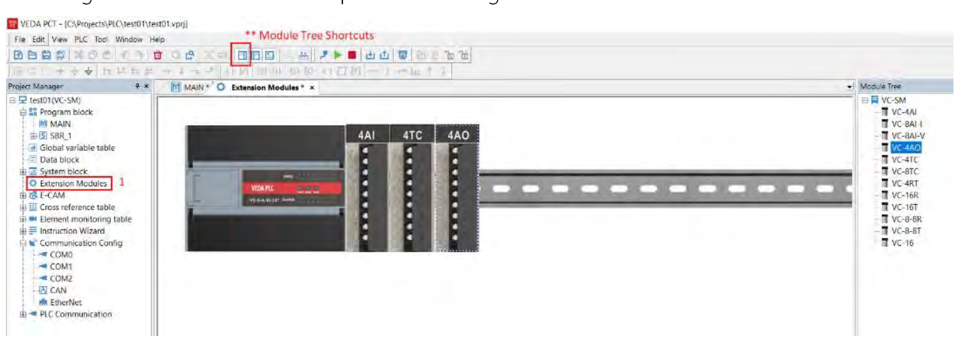
⑦ The remaining current is displayed, and a "cross" means that the number of expansion modules exceeds the capacity of the main module specification.

B. The module types supported by VEDA VC series PLC are shown in the following table:

Expansion Module Type	Expansion Module Model
VC-8-8R	8 inputs and 8 relay outputs
VC-8-8T	8 inputs and 8 transistor outputs
VC-16	16 inputs
VC-16R	16 relay outputs
VC-16T	16 transistor outputs
Special module type	
VC-4AI	4 analog inputs
VC-4AO	4 analog outputs
VC-4TC	4-way thermocouple temperature detection module
VC-4PT	4-way thermal resistance temperature detection module

13.2 Expansion Module Configuration

A. The expansion module adopts the hardware configuration method, and the control of the expansion module can be realized through the VEDA PCT host computer. The configuration is as follows:



① Double-click [Extension Module Configuration], the extension module configuration interface will pop up;

② Double-click the extension module to be added from the "module tree" on the right, and it can be automatically added to the rail, or the module in the left mouse button can be dragged to the rail.

③ The order of the positions of the modules can be adjusted by dragging and dropping. (The "Module Tree" can be opened or closed through the shortcut above)

Precautions:

(1) The digital input module and digital output module can also not be arranged on the guide rail, and the PLC host automatically recognizes the model of the digital quantity module.

(2) The configuration sequence of the special modules configured by the host computer needs to be consistent with the actual connection sequence, otherwise, it will prompt "special module configuration error" and the error light will flash. Digital modules are not affected by this.

3. How to use the module

1) Digital Input Module

After the input expansion module is connected to the main module, the number of the input X port on the expansion module is immediately followed by the number of the X port on the main module, and numbered in sequence. For example, the main module is a VC-B-A-16-14T general model. After connecting to the expansion module VC-16, the last X port number on the main module is X16, then the access numbers of the 16 input X ports on the expansion module during programming are X20~X27 and X30～X37, and so on for the subsequent digital input expansion modules.

2) Digital output module

After the output expansion module is connected to the main module, the output Y port number on the expansion module follows the number of the Y port on the main module.

numbered backwards. For example, the main module is VC-S-A-16-16T general model, after connecting the expansion module VC-16T, the last Y port number on the main module

If it is Y15, the access numbers of the 16 output Y ports on the expansion module during programming are Y20~Y27 and Y30-Y37, and so on for subsequent digital outputs.

3) Special module

When fixing cables, do not bundle cables with AC cables, main circuit cables, high-voltage cables, etc., which may increase the influence of noise, surge and induction;

Do single-point grounding for the shielding of shielded wires and solder-sealed cables; crimp terminals with sleeves without solder joints cannot be used for terminal blocks. It is recommended to use marking tubes or insulating tubes to cover the cable joints of the crimp terminals.

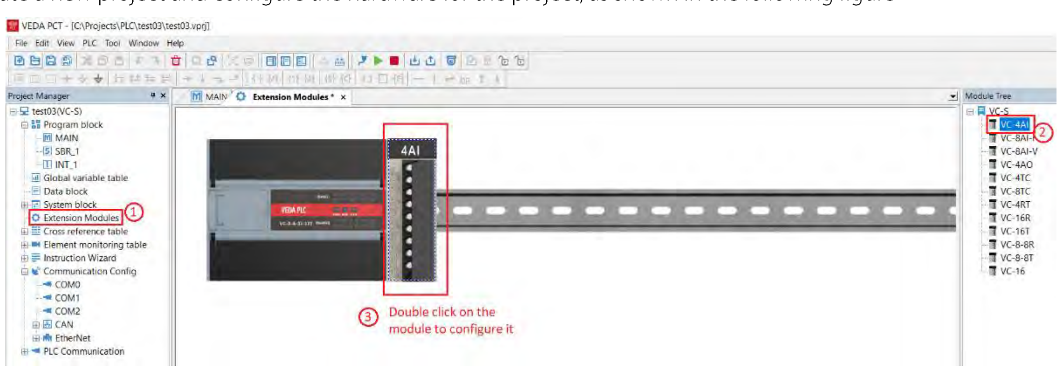
13.2.1 IO module configuration

The IO module does not need to be configured, the module can be reliably connected to the expansion interface on the right side of the main module, and the software will automatically identify it.

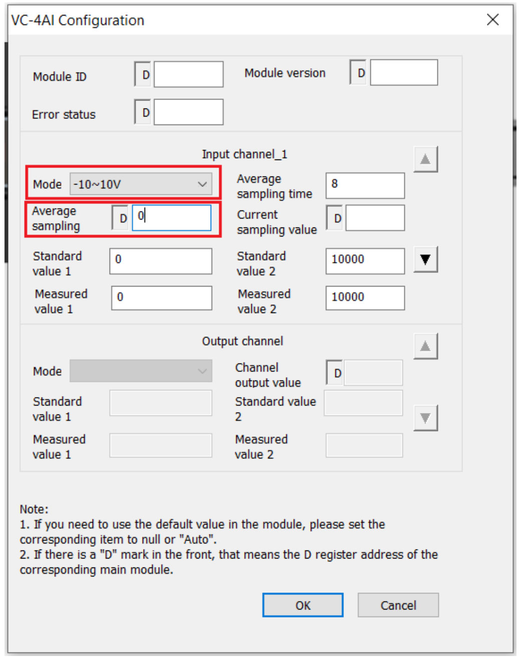
13.2.2 VC-4AI module programming example

Example: VC-4AI module address is 1, use its first channel input voltage signal (–10V～+10V), the second channel input current signal (–20mA～+20mA), turn off the third channel, set the average value to 8, and use the data registers D0, D2 to receive the average value conversion result.

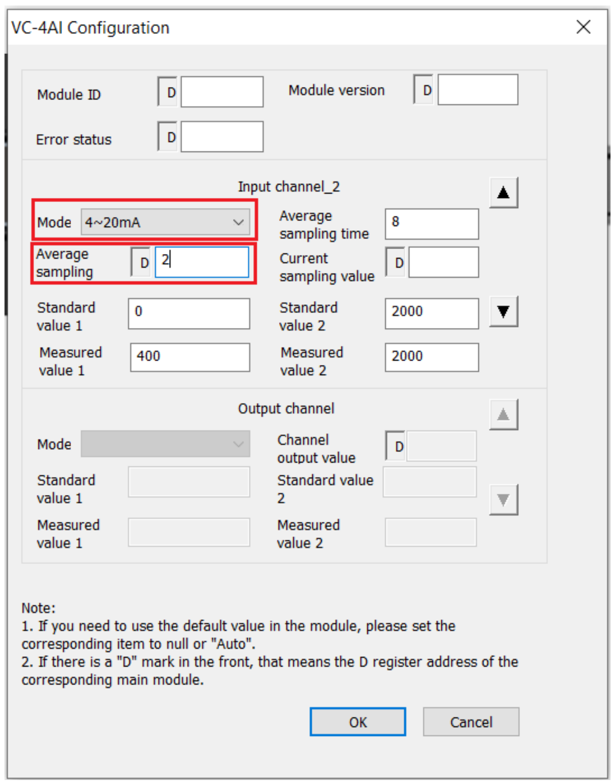
1) Create a new project and configure the hardware for the project, as shown in the following figure



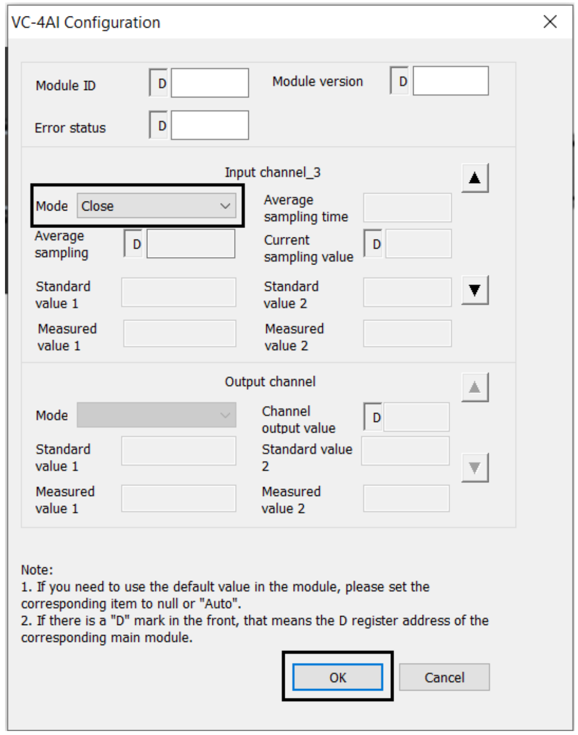
2) Double-click the "VC-4AI" module on the guide rail to enter the 4AD configuration parameters;



3) Click "▼" to select "±20mA" for the second channel mode configuration mode;



4) Click "▼" to configure the third channel mode, click "Confirm" after completion;

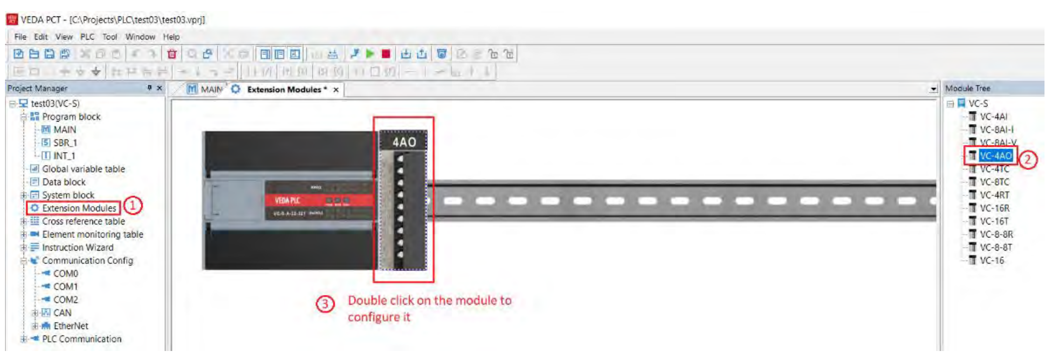


5) After compiling, download and run. The converted values of D0 and D2 can be monitored through the monitoring table. When using, take the value through the MOV instruction or use the register directly.

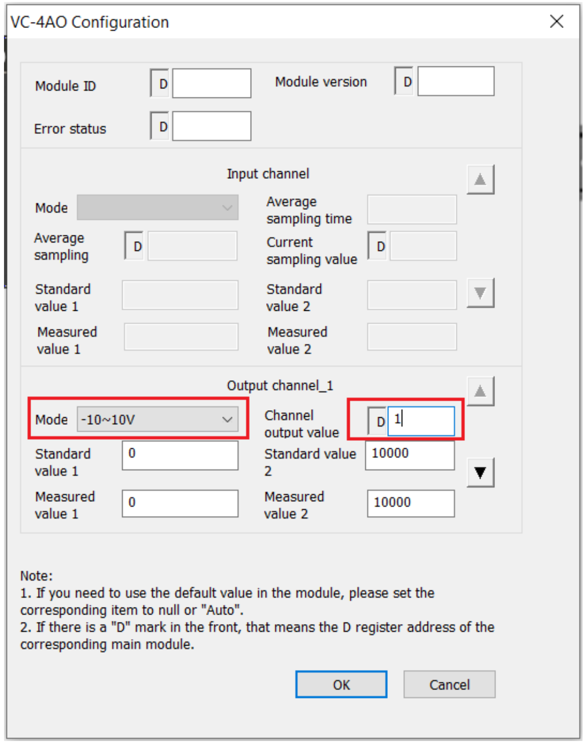
13.2.3 VC-4AO module programming example

Example: The address of the VC-4AO module is 1, so that it closes the first channel, the second channel outputs the voltage signal (-10V～10V), the third channel outputs the current signal (0～20mA), and the fourth channel outputs the current signal (4 ~20mA), and use data registers D1, D2, D3 to set the output voltage or current value.

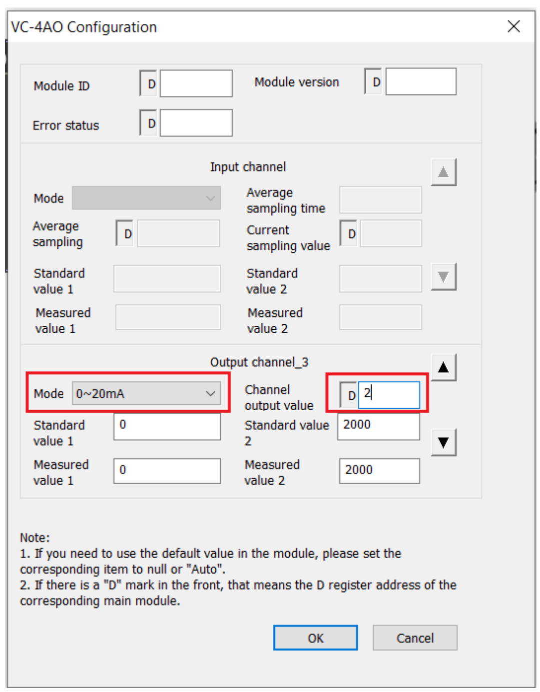
1) Create a new project and configure the hardware for the project, as shown in the following figure



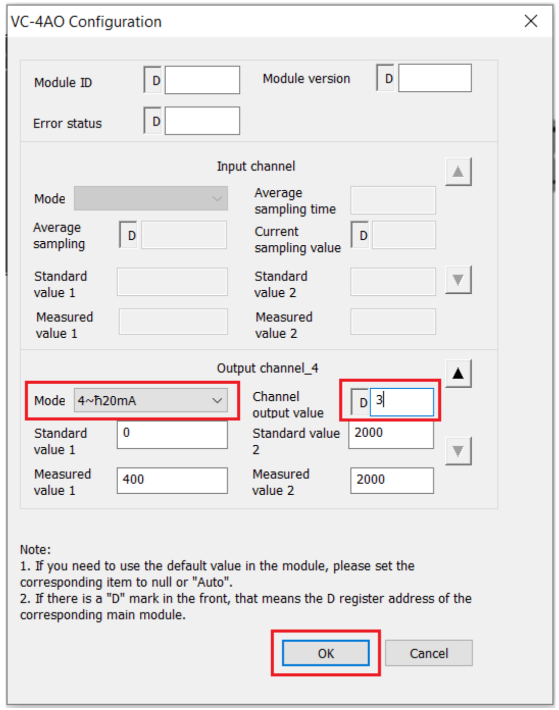
2) Double-click the "VC-4AO" module on the guide rail to enter the 4DA configuration parameters;



3) Click "▼" to configure the third channel mode;



4) Click "▼" to configure the fourth channel mode, click "Confirm" after completion;

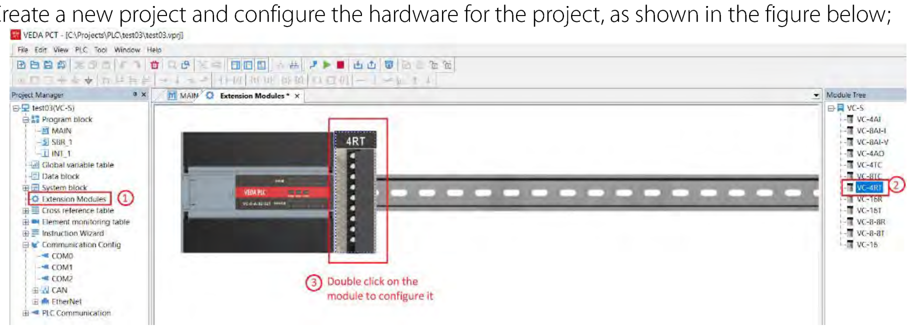


5) After the compilation is passed, download and run, you can assign values to D1, D2, D3 through the monitoring table, or assign values to the register through the MOV instruction. The module can output voltage or current according to the set value.

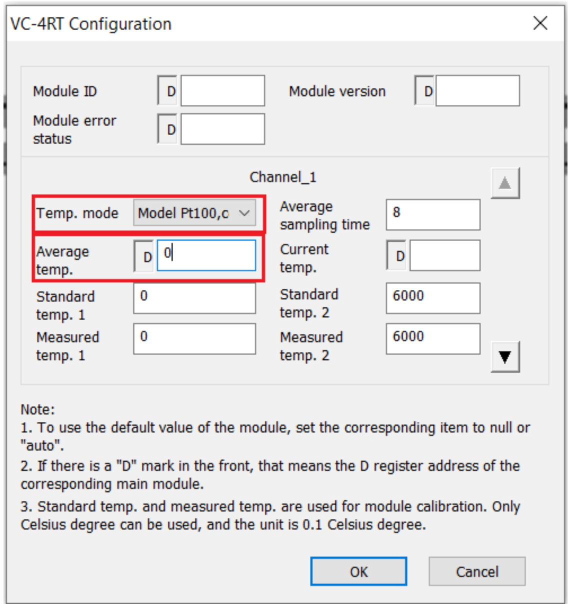
13.2.4 VC-4RT module programming example

Example: VC-4RT is connected to the No. 1 position of the expansion module, the first channel is connected to Pt100 thermal resistance to output the temperature in degrees Celsius, the second channel is connected to Cu100 thermal resistance to output the temperature in degrees Celsius, and the third channel is connected to Cu50 thermal resistance Output the temperature in Fahrenheit, close the 4th channel, set the number of average points to 8, and use the data registers D0, D1, D2 to receive the average conversion result;

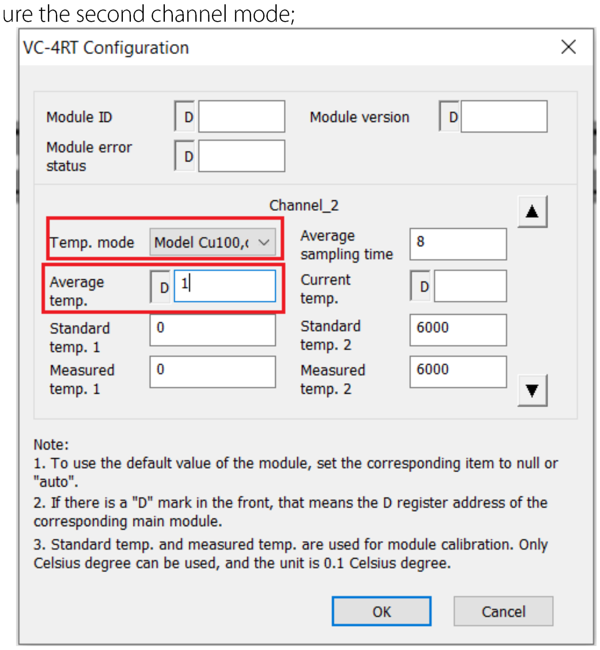
1) Create a new project and configure the hardware for the project, as shown in the figure below;



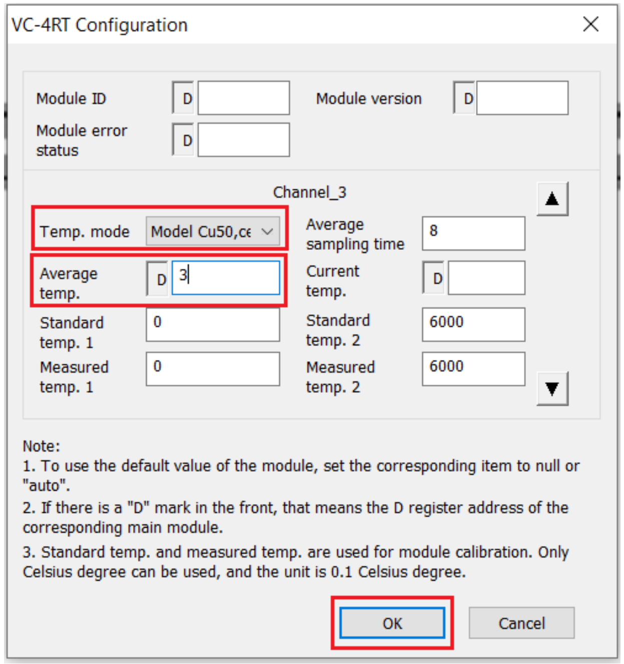
2) Double-click the "4RT" module to enter the 4RT setting interface - as shown below



3) Click "▼" to configure the second channel mode;



4) Click "▼" to configure the third channel mode, click "Confirm" after completion;

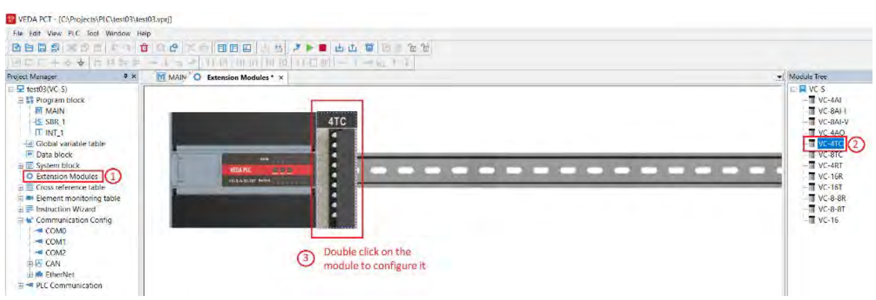


5) After the compilation is passed, download and run, you can view the converted values of D0, D1, D2 temperature modules through the monitoring table or read the register value through the MOV instruction.

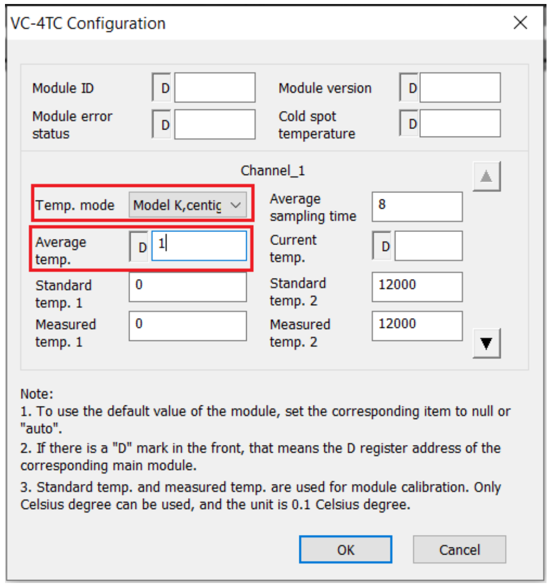
13.2.5 VC-4TC module programming example

Example: VC-4TC is connected to the No. 1 position of the expansion module, the first channel is connected to a K-type thermocouple to output the temperature in degrees Celsius, the second channel is connected to a J-type thermocouple to output the temperature in degrees Celsius, and the third channel is connected to a K-type thermocouple Output the temperature in Fahrenheit, turn off the 4th channel, set the number of average points to 8, and use the data registers D1, D3, D5 to receive the average conversion result.

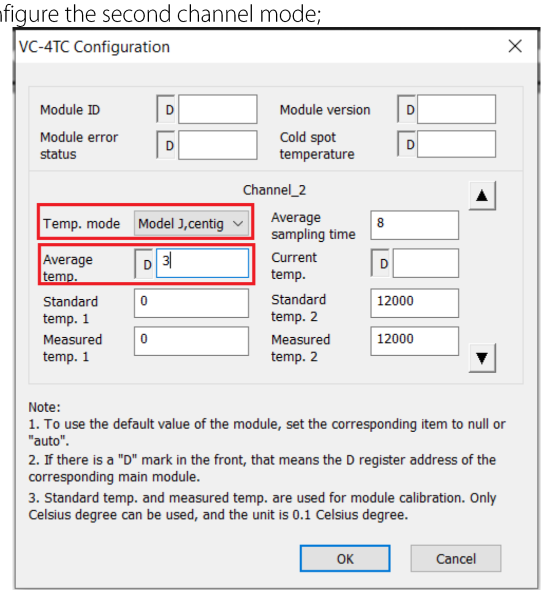
1) Create a new project and configure the hardware for the project, as shown in the following figure



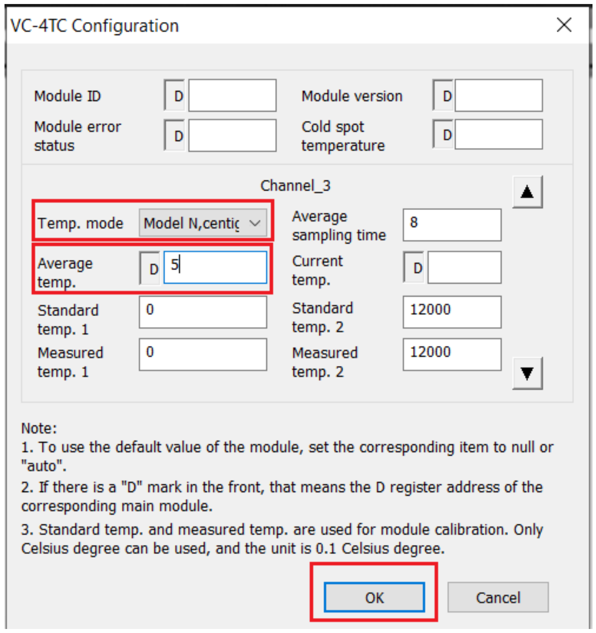
2) Double-click the "4TC" module to enter the 4TC setting interface - as shown below



3) Click "▼" to configure the second channel mode;



4) Click "▼" to configure the third channel mode, click "Confirm" after completion



5) After the compilation is passed, download and run, you can view the values converted by D1, D3, D5 temperature modules through the monitoring table or read the register value through the MOV instruction.

Appendix 1 Special Auxiliary Relay

All special auxiliary relays, in STOP，It is initialized by the system during RUN, and the special auxiliary relay set in the system setting will be re-assigned according to the setting value in the system block setting after the previous initialization is completed.

📖 Notice

The reserved SD is not listed in the SM table, and the read-write attribute of the reserved SM element is the read-write attribute (RW) by default.

1. PLC Working Status Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM0	Monitor run bits	Always on in run state, always off in stop state	R	√	√	
SM1	Initial run pulse bit	User program from stop to run, set high for one run cycle and then set low	R	√	√	
SM2	Power-ON sign	Set to ON when the system is powered ON, and set to OFF when the user program runs for one cycle	R	√	√	
SM3	System error	Set when a system error is detected after power-ON or from stop to run, if no system error occurs, this bit is cleared	R	√	√	
SM4	Battery voltage is too low	Set when the battery voltage is too low, clear when the battery voltage is detected to be higher than 2.4v	R	√	√	
SM7	No battery working mode	When this bit is set to 1, the battery backup data loss error and the forced table loss error will not be reported when the system battery fails. (can only be configured via the system block)	R	√	√	
SM8	Constant scan mode	When this bit is set, the scan time is Constant (can only be configured via the system block)	R	√	√	
SM9	Input point boot mode	After this bit is set, when the set x input point is ON, the PLC can enter the run state from stop (can only be configured through the system block)	R	√	√	

2. Clock Run Bit

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM10	10ms clock	10ms cycle clock oscillation (half cycle flips, the first half cycle is 0 when the user program is running)	R	√	√	
SM11	100ms clock	100ms cycle clock oscillation (half cycle flips, the first half cycle is 0 when the user program is running)	R	√	√	
SM12	1s clock	1s is the cycle clock oscillation (half cycle flips, the first half cycle is 0 when the user program is running)	R	√	√	
SM13	1min clock	1min is the cycle of clock oscillation (half cycle is reversed, and the first half cycle is 0 when the user program is running)	R	√	√	
SM14	1hour clock	1hour clock oscillation (half cycle flips, the first half cycle is 0 when the user program is running)	R	√	√	
SM15	Scan period oscillation	This bit toggles once every scan cycle (the first cycle is 0 when the user program is running)	R	√	√	

Address	Name	Actions and functions	R/W	VC-B	VC-S	
	bit					

3. User Program Execution Error

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM20	Command execution error	Instruction execution error, set. At the same time, fill in sd20 with the specific error type code. Cleared after executing the application instruction correctly	R	√	√	
SM21	Command element number subscript overflow	Instruction execution error, set. At the same time, fill in sd20 with the specific error type code	R	√	√	
SM22	Illegal command parameter	Instruction execution error, set. At the same time, fill in sd20 with the specific error type code. Cleared after executing the application instruction correctly	R	√	√	

4. Interrupt Control

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM25	X0 input rising edge interrupt enable flag bit	When set to 1, enable x0 rising edge interrupt	R/W	√	√	
SM26	X1 input rising edge interrupt enable flag bit	When set to 1, enable x1 rising edge interrupt	R/W	√	√	
SM27	X2 input rising edge interrupt enable sign	When set to 1, enable x2 rising edge interrupt	R/W	√	√	
SM28	X3 input rising edge interrupt enable flag bit	When set to 1, enable x3 rising edge interrupt	R/W	√	√	
SM29	X4 input rising edge interrupt enable sign	When set to 1, enable x4 rising edge interrupt	R/W	√	√	
SM30	X5 input rising edge interrupt enable flag bit	When set to 1, enable x5 rising edge interrupt	R/W	√	√	
SM31	X6 input rising and falling edge interrupt enable flag bit	When set to 1, enable x6 rising edge interrupt	R/W	√	√	
SM32	X7 input rising edge interrupt enable flag bit	When set to 1, enable x7 rising edge interrupt	R/W	√	√	
SM33	X0 input falling edge interrupt enable sign	When set to 1, enable x0 falling edge interrupt	R/W	√	√	
SM34	X1 input falling edge interrupt enable sign	When set to 1, enable x1 falling edge interrupt	R/W	√	√	
SM35	X2 input falling edge interrupt enable sign	When set to 1, enable x2 falling edge interrupt	R/W	√	√	
SM36	X3 input falling edge interrupt enable sign	When set to 1, enable x3 falling edge interrupt	R/W	√	√	
SM37	X4 input falling edge interrupt enable sign	When set to 1, enable x4 falling edge interrupt	R/W	√	√	
SM38	X5 input falling edge interrupt enable sign	When set to 1, enable x5 falling edge interrupt	R/W	√	√	
SM39	X6 input falling edge interrupt enable sign	When set to 1, enable x6 falling edge interrupt	R/W	√	√	
SM40	X7 input falling edge interrupt enable sign	When set to 1, enable x7 falling edge interrupt	R/W	√	√	
SM41	Frame send interrupt enable Sign of COM0	When set to 1, allows	R/W	√	√	

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM42	Frame receive interrupt enable Sign of COM0	When set to 1, allows	R/W	√	√	
SM43	Frame send interrupt enable Sign of COM1	When set to 1, allows	R/W	√	√	
SM44	Frame receive interrupt enable Sign of COM1	When set to 1, allows	R/W	√	√	
SM45	COM2 frame transmission interrupt enable Flag bit	When set to 1, allows	R/W	√	√	
SM46	Frame receive interrupt enable Sign of COM2	When set to 1, allows	RW	√	√	
SM47	Timed interrupt 0 enable Flag bit	When set to 1, enable timed interrupt 0	RW	√	√	
SM48	Timed interrupt 1 enable Flag bit	When set to 1, enable timed interrupt 1	R/W	√	√	
SM49	Timed interrupt 2 enable Flag bit	When set to 1, timed interrupt 2 is enabled	R/W	√	√	
SM50	High-speed output 0 complete interrupt enable Sign	When set to 1, enable Y0 high-speed output completion interrupt	R/W	√	√	
SM51	High-speed output 1 complete interrupt enable Sign	When set to 1, the Y1 high-speed output completion interrupt is enabled	R/W	√	√	
SM52	High-speed output 2 complete interrupt enable Sign	When set to 1, the Y2 high-speed output completion interrupt is enabled	R/W	√	√	
SM53	High-speed output 3 complete interrupt enable Sign	When set to 1, enable Y3 high-speed output completion interrupt	R/W		√	
SM54	High-speed output 4 complete interrupt enable Sign	When set to 1, the Y4 high-speed output completion interrupt is enabled	R/W		√	
SM55	High-speed output 5 complete interrupt enable Sign	When set to 1, enable Y5 high-speed output completion interrupt	R/W		√	
SM56	High-speed output 6 complete interrupt enable Sign	When set to 1, the Y6 high-speed output completion interrupt is enabled	R/W		√	
SM57	High-speed output 7 complete interrupt enable Sign	When set to 1, enable Y7 high-speed output completion interrupt	R/W		√	
SM58	High-speed count interrupt enable Sign	When set to 1, the high-speed counting interrupt is enabled	R/W	√	√	
SM63	Positioning instruction passes through position interrupt 0 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 0	R/W		√	
SM64	Positioning instruction passes position interrupt 1 enable Sign	When set to 1, enables the positioning instruction to pass through the position interrupt 1	R/W		√	
SM65	The positioning instruction passes through the position interrupt 2 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 2	R/W		√	
SM66	The positioning instruction passes through the position interrupt 3 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 3	R/W		√	
SM67	The positioning instruction passes through the position interrupt 4 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 4	R/W		√	
SM68	Positioning instruction passes through position interrupt 5 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 5	R/W		√	

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM69	The positioning instruction passes through the position interrupt 6 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 6	R/W		√	

SM70	The positioning instruction passes through the position interrupt 7 enable Sign	When set to 1, enables the positioning command to pass through the position interrupt 7	R/W		√	

5. Peripheral Instructions

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM78	Print mode selection	When set to 1, 1-16 characters, 0 fixed 8 characters	R/W		√	
SM79	Printing in progress	When set to 1, printing is in progress	R		√	

6. Operation Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM80	Zero flag	When the associated operation result is zero, this bit is set when the associated instruction is executed. It can be manually cleared by the user	R/W	√	√	
SM81	Carry/overflow sign	When there is a carry in the relevant operation, the bit is set when the relevant instruction is executed, and the user can manually clear it.	R/W	√	√	
SM82	Excuse me	When the relevant operation has a borrow, the bit is set when the relevant instruction is executed, and the user can manually clear and set this bit	R/W	√	√	

7. DHST/DHSP Form Comparison Completion Sign

Address	Name	Actions and functions	R/W	VC-B	VC-B	
SM83	Table compare signs	Set when the entire table record is complete	R/W	√	√	

8. ASCII Conversion Instruction Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM85	Asc instruction storage mode sign	0: each high and low byte of each word stores 1 ascii code 1: the low byte of each word stores an ascii code	R/W	√	√	

9. MTR Instruction Execution End Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM86 (SM187)	Instruction execution ends	Turns ON after the first cycle operation of the mtr instruction	R/W		√	

10. Data Block Compare Set Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM88	Data block compare set	Set when the comparison results in the data block are all 1	R/W		√	

11. Pulse Catch Monitor Bit

Address	Name	Function	R/W	VC-B	VC-S	
SM90	Input X0 pulse capture monitor bit	(1) cleared from stop->run; (2) there are hcnt high-speed counting drive instructions	R/W	√	√	
SM91	Input X1 pulse capture	and spd pulse density detection instructions ON this port,	R/W	√	√	

	monitor bit	and the pulse capture of the port is invalid; it is valid in other cases; for details, refer to spd and hcnt instructions				
SM92	Input X2 pulse capture monitor bit		R/W	√	√	
SM93	Input X3 pulse capture monitor bit		R/W	√	√	
SM94	Input X4 pulse capture monitor bit		R/W	√	√	
SM95	Input X5 pulse capture monitor bit		R/W	√	√	
SM96	Input X6 pulse capture monitor bit		R/W	√	√	
SM97	Input X7 pulse capture monitor bit		R/W	√	√	
Note: 1. Cleared from stop to run. There are hcnt high-speed counter drive instructions and spd frequency measurement instructions ON this port, and the pulse capture of the port is invalid. Valid in other cases. For details, refer to 6.10.9 spd: frequency measurement command and 6.10.1 hcnt: high-speed counter drive command.						

12. Quadruple Frequency

Address	Name	Function	R/W	VC-B	VC-S	
SM100	X0, X1 ab phase input 1x/4x switching	Cleared by STOP—>RUN;	R/W	√	√	
SM101	X2, X3 ab phase input 1x/4x switching		R/W	√	√	
SM102	X4, X5 ab phase input 1x/4x switching		R/W	√	√	
SM103	X6, X7 ab phase input 1x/4x switching		R/W	√	√	

13. Communication Port (COM0)

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM110	COM0 send enable sign	This bit is set when the XMT instruction is used, and will be automatically cleared when the transmission is completed. The user can also manually clear it to abort the current sending task of port 0. When the power flow is turned ON again, the sending task of the port can continue	R/W	√	√	
SM111	COM0 receive enable sign	This bit is set when the RCV instruction is used, and will be automatically cleared when the transmission is completed. The user can also manually clear it to abort the current sending task of port 0. When the power flow is turned ON again, the sending task of the port can continue	R/W	√	√	
SM112	COM0 send complete sign	Send complete set	R/W	√	√	
SM113	COM0 receive completion sign	Receive complete set	R/W	√	√	
SM114	COM0 idle sign	When the port has no communication task, the flag bit is set	R	√	√	

📖 Notice

SM110-SM114 is a receive, complete and idle Sign for all communication protocols that use COM0. For example: COM0 of VC-B PLC can be used for Modbus protocol, no matter which protocol is used, SM110-SM114 are applicable.

14. Communication Port (COM1)

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM120	Com1 send enable sign	This bit is set when the XMT instruction is used, and will be automatically cleared when the transmission	R/W	√	√	

Address	Name	Actions and functions	R/W	VC-B	VC-S	
		is completed. The user can also manually clear it to abort the current sending task of port 1. When the power flow is turned ON again, the sending task of the port can continue				
SM121	Com1 receive enable sign	This bit is set when the RCV instruction is used, and will be automatically cleared when the reception is completed. The user can also manually clear it to abort the current sending task of port 1. When the power flow is turned ON again, the receiving task of this port can continue	R/W	√	√	
SM122	Com1 send complete sign	Send complete set	R/W	√	√	
SM123	Com1 receive completion sign	Receive complete set	R/W	√	√	
SM124	Com1 idle sign	When the port has no communication task, the flag bit is set	R	√	√	
SM125	Com1 modbus communication completed	After the communication is completed, the flag bit is set	R/W	√	√	
SM126	Com1 modbus communication error	After a communication error, this sign is set	R/W	√	√	

📖 Notice

SM120-SM126is for applicable COM1A send complete, receive complete and idle Sign for all communication protocols. For example: COM of VC-B PLC1Available for N: N, Modbus and FREEPORT protocols, whichever protocol is used, SM120-SM126both apply.

15. Extended Communication Port (COM 2)

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM130	COM2 send enable sign	This bit is set when the XMT instruction is used, and will be automatically cleared when the transmission is completed. The user can also manually clear it to abort the current sending task of port 2. When the power flow is turned ON again, the sending task of the port can continue	R/W	√	√	
SM131	COM2 receive enable sign	This bit is set when the RCV instruction is used, and will be automatically cleared when the reception is completed. The user can also manually clear it to abort the current sending task of port 2. When the power flow is turned ON again, the receiving task of this port can continue	R/W	√	√	
SM132	COM2 send complete sign	Send complete set	R/W	√	√	
SM133	COM2 receive complete sign	Receive complete set	R/W	√	√	
SM134	COM2 idle sign	When the port has no communication task, the flag bit is set	R	√	√	
SM135	COM2 MODBUS communication completed	After the communication is completed, the flag bit is set	R/W	√	√	
SM136	COM2 MODBUS communication error	After a communication error, this sign is set	R/W	√	√	

📖 Notice

SM130-SM136is for applicableCOM2 A send complete, receive complete and idle Sign for all communication protocols.

16. N: N Communication

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM140	Communication error sign of station 0	Station no. 0 communication error SM element turns ON	R	√	√	
SM141	Station no. 1 communication error sign	Station no. 1 communication error SM element turns ON	R	√	√	
SM142	Station no. 2 communication error sign	Station no. 2 communication error SM element turns ON	R	√	√	
SM143	Station no. 3 communication error sign	Station no. 3 communication error SM element turns ON	R	√	√	
SM144	Station no. 4 communication error sign	Station no. 4 communication error SM element turns ON	R	√	√	
SM145	Communication error sign of station 5	Station no. 5 communication error SM element turns ON	R	√	√	
SM146	Station no. 6 communication error sign	Station no. 6 communication error SM element turns ON	R	√	√	
SM147	Station no. 7 communication error sign	Station no. 7 communication error SM element turns ON	R	√	√	
SM148	Communication error sign of station no. 8	Station no. 8 communication error SM element turns ON	R	√	√	
SM149	Communication error sign of station no. 9	Station no. 9 communication error SM element turns ON	R	√	√	
SM150	Communication error sign of station no. 10	Station no. 10 communication error SM element turns ON	R	√	√	
SM151	Communication error sign of station no. 11	Station no. 11 communication error SM element turns ON	R	√	√	
SM152	Communication error sign of station no. 12	Station no. 12 communication error SM element turns ON	R	√	√	
SM153	Communication error sign of station 13	Station no. 13 communication error SM element turns ON	R	√	√	
SM154	Communication error sign of station no. 14	Station no. 14 communication error SM element turns ON	R	√	√	
SM155	Communication error sign of station 15	Station no. 15 communication error SM element turns ON	R	√	√	
SM156	Communication error sign of station 16	Station no. 16 communication error SM element turns ON	R	√	√	
SM157	Communication error sign of station 17	Station no. 17 communication error SM element turns ON	R	√	√	
SM158	Communication error sign of station 18	Station no. 18 communication error SM element turns ON	R	√	√	
SM159	Communication error sign of station 19	Station no. 19 communication error SM element turns ON	R	√	√	
SM160	Communication error sign of station 20	Station no. 20 communication error SM element turns ON	R	√	√	
SM161	Communication error sign of station no. 21	Station no. 21 communication error SM element turns ON	R	√	√	
SM162	Communication error sign of station 22	Station no. 22 communication error SM element turns ON	R	√	√	
SM163	Communication error sign of station no. 23	Station no. 23 communication error SM element turns ON	R	√	√	
SM164	Communication error sign of station no. 24	Station no. 24 communication error SM element turns ON	R	√	√	
SM165	Communication error sign of	Station no. 25 communication	R	√	√	

Address	Name	Actions and functions	R/W	VC-B	VC-S	
	station no. 25	error SM element turns ON				
SM166	Communication error sign of station 26	Station no. 26 communication error SM element turns ON	R	√	√	
SM167	Communication error sign of station no. 27	Station no. 27 communication error SM element turns ON	R	√	√	
SM168	Communication error sign of station no. 28	Station no. 28 communication error SM element turns ON	R	√	√	
SM169	Communication error sign of station 29	Station no. 29 communication error SM element turns ON	R	√	√	
SM170	Communication error sign of station 30	Station no. 30 communication error SM element turns ON	R	√	√	
SM171	Communication error sign of station 31	Station no. 31 communication error SM element turns ON	R	√	√	

17. System Bus Error Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM190	Main module bus error sign	1. Power-ON addressing is cleared correctly 2. STOP→RUN without this error clear 3. Clear when downloading new programs 4. This bit causes system shutdown	R	√	√	
SM191	General module bus error sign	1. When a general module bus operation error occurs, this bit is set, and the system alarms 2. System fault elimination sign is automatically cleared	R	√	√	
SM192	Special module bus error sign	1. When a special module bus operation error occurs, this bit is set, and the system alarms 2. System fault elimination sign is automatically cleared	R	√	√	

18. Real Time Clock Error Sign

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM193	Read and write real time clock error	When a real-time clock error occurs, this bit is set; the system fault elimination sign is automatically cleared	R	√	√	

19. Up/Down Counter Counting Direction

Address number	Corresponding counter address number	Function	R/W	VC-B	VC-S	
SM200	C200	When SM2 _ _ is high level, its corresponding C2_ _ becomes count down When SM2 _ _ is low, its corresponding C2_ _ becomes an increment count	R/W	√	√	
SM201	C201		R/W	√	√	
SM202	C202		R/W	√	√	
SM203	C203		R/W	√	√	
SM204	C204		R/W	√	√	
SM205	C205		R/W	√	√	
SM206	C206		R/W	√	√	
SM207	C207		R/W	√	√	
SM208	C208		R/W	√	√	
SM209	C209		R/W	√	√	
SM210	C210		R/W	√	√	
SM211	C211		R/W	√	√	

Address number	Corresponding counter address number	Function	R/W	VC-B	VC-S	
SM212	C212		R/W	√	√	
SM213	C213		R/W	√	√	
SM214	C214		R/W	√	√	
SM215	C215		R/W	√	√	
SM216	C216		R/W	√	√	
SM217	C217		R/W	√	√	
SM218	C218		R/W	√	√	
SM219	C219		R/W	√	√	
SM220	C220		R/W	√	√	
SM221	C221		R/W	√	√	
SM222	C222		R/W	√	√	
SM223	C223		R/W	√	√	
SM224	C224		R/W	√	√	
SM225	C225		R/W	√	√	
SM226	C226	When SM2 _ _ is high level, its corresponding C2_ _ becomes count down When SM2 _ _ is low, its corresponding C2_ _ becomes an increment count	R/W	√	√	
SM227	C227		R/W	√	√	
SM228	C228		R/W	√	√	
SM229	C229		R/W	√	√	
SM230	C230		R/W	√	√	
SM231	C231		R/W	√	√	
SM232	C232		R/W	√	√	
SM233	C233		R/W	√	√	
SM234	C234		R/W	√	√	
SM235	C235		R/W	√	√	

20. Counting Direction and Monitoring of High-Speed Counter

Distinguish	Address number	Name	Register content	R/W	VC-B	VC-S	
Single-phase single-ended counting input	SM236	C236	Its corresponding SM2 _ _ becomes a high level and a low level corresponding to the decrease and increase of the counter, respectively. When C2 _ _ of the single-phase up-down count input counter and the two-phase count input counter is in the down-counting mode, its corresponding SM2 _ _ becomes high level. When counting up, it is low level	R/W	√	√	
	SM237	C237		R/W	√	√	
	SM238	C238		R/W	√	√	
	SM239	C239		R/W	√	√	
	SM240	C240		R/W	√	√	
	SM241	C241		R/W	√	√	
	SM242	C242		R/W	√	√	
	SM243	C243		R/W	√	√	
	SM244	C244		R/W	√	√	
	SM245	C245		R/W	√	√	
	SM246	C246		R/W	√	√	
	SM247	C247		R/W	√	√	
Single-phase up/down count input	SM248	C248	Register content Its corresponding SM2 _ _ becomes a high level and a low level corresponding to the decrease and increase of the counter, respectively.	R	√	√	
	SM249	C249		R	√	√	
	SM250	C250		R	√	√	
	SM251	C251		R	√	√	
	SM252	C252		R	√	√	
	SM253	C253		R	√	√	
	SM254	C254		R	√	√	
	SM255	C255		R	√	√	

Distinguish	Address number	Name	Register content	R/W	VC-B	VC-S	
Two-phase counting input Distinguish	SM256	C256		R	√	√	
	SM257	C257		R	√	√	
	SM258	C258		R	√	√	
	SM259	C259		R	√	√	
	SM260	C260		R	√	√	
	SM261	C261		R	√	√	
	SM262	C262		R	√	√	
	SM263	C263		R	√	√	

21. High-Speed Output and Positioning Command

（1） Y0 related Signs

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM270	Pulse output stop control	Set this element to stop the high-speed pulse output function of Y0; reset it to turn ON the output function	R/W	√	√	
SM271	Pulse output monitoring	Used to monitor the status of high-speed output channel Y0, ON when busy and OFF when ready	R/W	√	√	
SM272	PWM output valid in microseconds	When this bit is set, the pwm instruction of Y0 is output in microseconds	R/W	√	√	
SM273	Interrupt drive pulse output valid	When it is ON, the plsy instruction of Y0 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W	√	√	
SM274	Envelope loop execution	When ON, the pls instruction of Y0 is executed repeatedly	R/W	√	√	
SM275	PLSV progressive frequency conversion	When ON, the pslv command frequency of Y0 changes gradually, enabling the acceleration/deceleration function.	R/W	√	√	
SM276	Clear function is valid	Applicable to dszr/zrn, acting ON the axis corresponding to Y0: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W	√	√	
SM277	Clear signal specified element is valid	Applicable to dszr, acting ON the axis corresponding to Y0: when set, use the Y element Y (n) corresponding to the value n in sd175 to represent the clearing signal; for reset, define y10 as the clearing signal according to the default value	Rw	√	√	
SM278	Return-to-origin direction	Applicable to dszr, acting ON the axis corresponding to Y0: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	Rw	√	√	
SM279	Forward limit	Applicable to dszr/dvit, acting ON the axis corresponding to Y0: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached	Rw	√	√	

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM280	Inversion limit	Applicable to dszr/dvit, acting ON the axis corresponding to Y0: when set, it indicates that the limit of the reverse direction is reached; when reset, it indicates that the limit has not been reached	Rw	√	√	
SM281	Near-point signal logic inversion	Applicable to dszr, acting ON the axis corresponding to Y0: when it is set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when it is reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	Rw	√	√	
SM282	Zero flagal logic inversion	Applicable to dszr, acting ON the axis corresponding to Y0: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	Rw	√	√	
SM283	Interrupt signal logic inversion	Applicable to dvit, acting ON the axis corresponding to Y0: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W	√	√	
SM284	Interrupt input function specification is valid	Applicable to dvit when the specified function of interrupt input is not used, then yo corresponds to the interrupt of x0. When the specified function is used, the bit is set, and then the corresponding sd176 value corresponds to the interrupt signal of Y0.	R/W	√	√	
SM285	User interrupt input command	Y0 for dvit	R/W	√	√	
SM286	S-type acceleration and deceleration are valid	Y0 s-type acceleration and deceleration valid SM is ON to enable s-type acceleration and deceleration function, OFF is t-type acceleration and deceleration	R/W	√	√	
SM287	Dvit interrupt signal masking is valid	Y0 is suitable for dvit SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W	√	√	

（2） Y1 related Signs

Address	Name	Actions and Functions	R/W	VC-B	VC-S	
SM290	Pulse output stop control	Set this element to stop the high-speed pulse output function of Y1; reset it to turn ON the output function	R/W	√	√	
SM291	Pulse output monitoring	Used to monitor the status of high-speed output channel Y1, ON when busy and OFF when ready	R/W	√	√	
SM292	PWM output valid in microseconds	When this bit is set, the PWM instruction of Y1 is output in microseconds	R/W	√	√	

SM293	Interrupt drive pulse output valid	When it is ON, the PLSY instruction of Y1 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W	√	√	
SM294	Envelope loop execution	When ON, the PLS instruction of Y1 is executed repeatedly	R/W	√	√	
SM295	Plsv progressive frequency conversion	When ON, the pslv command frequency of Y1 changes gradually, enabling the acceleration/deceleration function.	R/W	√	√	
SM296	Clear function is valid	Applicable to dszr/zrn, acting ON the axis corresponding to Y1: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W	√	√	
SM297	Clear signal specified element is valid	Applicable to dszr, acting ON the axis corresponding to Y1: when set, use the Y element Y (n) corresponding to the value n in sd195 to represent the clearing signal; for reset, define y11 as the clearing signal according to the default value	R/W	√	√	
SM298	Return-to-origin direction	Applicable to dszr, acting ON the axis corresponding to Y1: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	R/W	√	√	
SM299	Forward limit	Applicable to dszr/dvit, acting ON the axis corresponding to Y1: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached	R/W	√	√	
SM300	Inversion limit	Applicable to dszr/dvit, acting ON the axis corresponding to Y1: when set, it means the limit of the reverse direction is reached; when reset, it means that the limit is not reached	R/W	√	√	
SM301	Near-point signal logic inversion	Applicable to dszr, acting ON the axis corresponding to Y1: when it is set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when it is reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	R/W	√	√	
SM302	Zero flagal logic inversion	Applicable to dszr, acting ON the axis corresponding to Y1: when set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W	√	√	
SM303	Interrupt signal logic inversion	Applicable to dvit, acting ON the axis corresponding to Y1: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the	R/W	√	√	

		input is ON, the zero flagal is ON)				
SM304	Interrupt input function specification is valid	Applicable to dvit when the specified function of interrupt input is not used, then Y1 corresponds to the interrupt of x1. When the specified function is used, the bit is set, and then the corresponding sd196 value corresponds to the interrupt signal of Y1.	R/W	√	√	
SM305	User interrupt input command	Y1 is suitable for dvit	R/W	√	√	
SM306	S-type acceleration and deceleration are valid	Y1 s-type acceleration and deceleration valid SM is ON to enable s-type acceleration and deceleration function, OFF is t-type acceleration and deceleration	R/W	√	√	
SM307	Dvit interrupt signal masking is valid	Y1 is suitable for dvit SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W	√	√	

（3） Y2 related Signs

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM310	Pulse output stop control	Set this element to stop the high-speed pulse output function of y2; reset it to turn ON the output function	R/W	√	√	
SM311	Pulse output monitoring	Used to monitor the status of high-speed output channel y2, ON when busy and OFF when ready	R/W	√	√	
SM312	Pwm output valid in microseconds	When this bit is set, the PWM instruction of y2 is output in microseconds	R/W	√	√	
SM313	Interrupt drive pulse output valid	When it is ON, the plsy instruction of y2 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W	√	√	
SM314	Envelope loop execution	When ON, the PLS instruction of y2 is executed repeatedly	R/W	√	√	
SM315	Plsv progressive frequency conversion	When ON, the PSLV command frequency of y2 changes gradually, enabling the acceleration/deceleration function.	R/W	√	√	
SM316	Clear function is valid	Applicable to DSZR/ZRN, acting ON the axis corresponding to y2: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W	√	√	
SM317	Clear signal specified element is valid	Applicable to DSZR, acting ON the axis corresponding to y2: when set, use the Y element Y (n) corresponding to the value n in sd215 to represent the clearing signal; for reset, define y12 as the clearing signal according to the default value	R/W	√	√	
SM318	Return-to-origin direction	Applicable to DSZR, acting ON the axis corresponding to y2: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	R/W	√	√	
SM319	Forward limit	Applicable to DSZR/DVIT, acting ON the axis	R/W	√	√	

		corresponding to y2: when it is set, it means that the limit of the forward direction is reached; when it is reset, it means that the limit is not reached				
SM320	Inversion limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to y2: when set, it means the limit of the reverse direction is reached; when reset, it means that the limit is not reached	R/W	√	√	
SM321	Near-point signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to y2: when set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	R/W	√	√	
SM322	Zero flagal logic inversion	Applicable to DSZR, acting ON the axis corresponding to y2: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W	√	√	
SM323	Interrupt signal logic inversion	Applicable to DVIT, acting ON the axis corresponding to y2: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W	√	√	
SM324	Interrupt input function specification is valid	Applicable to DVIT when the specified function of interrupt input is not used, then y2 corresponds to the interrupt of x2. When the specified function is used, the bit is set, and then the corresponding sd216 value corresponds to the interrupt signal of y2.	R/W	√	√	
SM325	User interrupt input command	Y2 is suitable for DVIT	R/W	√	√	
SM326	S-type acceleration and deceleration are valid	Y2 S-TYPE acceleration and deceleration is valid; SM is ON to enable s-type acceleration and deceleration function, and OFF is t-type acceleration and deceleration	R/W	√	√	
SM327	Dvit interrupt signal masking is valid	Y2 is suitable for DVIT SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W	√	√	

（4） Y3 related Signs

address	name	Actions and Functions	R/W	VC-B	VC-S	
SM330	Pulse output stop control	Set this component to stop the high-speed pulse output function of Y3; reset it to turn ON the output function	R/W		√	
SM331	Pulse output monitoring	Used to monitor the status of high-speed output channel Y3, ON when busy and OFF when ready	R/W		√	
SM332	Pwm output valid in microseconds	When this bit is set, the PWM instruction of Y3 is output in microseconds	R/W		√	
SM333	Interrupt drive pulse output valid	When it is ON, the plsy instruction of Y3 can be called in the interrupt program and	R/W		√	

		subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.				
SM334	Envelope loop execution	When ON, the PLS instruction of Y3 is executed repeatedly	R/W		√	
SM335	Plsv progressive frequency conversion	When ON, the PSLV command frequency of Y3 changes gradually, enabling the acceleration/deceleration function.	R/W		√	
SM336	Clear function is valid	Applicable to DSZR/ZRN, acting ON the axis corresponding to Y3: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W		√	
SM337	Clear signal specified element is valid	Applicable to DSZR, acting ON the axis corresponding to Y3: when set, use the Y element Y (n) corresponding to the value n in sd235 to represent the clearing signal; for reset, define y13 as the clearing signal according to the default value	R/W		√	
SM338	Return-to-origin direction	Applicable to DSZR, acting ON the axis corresponding to Y3: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	R/W		√	
SM339	Forward limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y3: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached	R/W		√	
SM340	Inversion limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y3: when set, it indicates that the limit of the reverse direction is reached; when reset, it indicates that the limit has not been reached	R/W		√	
SM341	Near-point signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y3: when set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	R/W		√	
SM342	Zero flagal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y3: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	
SM343	Interrupt signal logic inversion	Applicable to DVIT, acting ON the axis corresponding to Y3: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	

SM344	Interrupt input function specification is valid	Applicable to DVIT when the specified function of interrupt input is not used, then Y3 corresponds to the interrupt of x3. When the specified function is used, the bit is set, and then the corresponding sd236 value corresponds to the interrupt signal of Y3.	R/W		√	
SM345	User interrupt input command	Y3 for DVIT	R/W		√	
SM346	S-type acceleration and deceleration are valid	Y3 S-TYPE acceleration and deceleration is valid; SM is ON to enable s-type acceleration and deceleration function, OFF is t-type acceleration and deceleration	R/W		√	
SM347	Dvit interrupt signal masking is valid	Y3 is suitable for DVIT SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W		√	

（5） Y4 related Signs

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM350	Pulse output stop control	Set this element to stop the high-speed pulse output function of Y4; reset it to turn ON the output function	R/W		√	
SM351	Pulse output monitoring	Used to monitor the status of high-speed output channel Y4, ON when busy and OFF when ready	R/W		√	
SM352	Pwm output valid in microseconds	When this bit is set, the PWM instruction of Y4 is output in microseconds	R/W		√	
SM353	Interrupt drive pulse output valid	When it is ON, the plsy instruction of Y4 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W		√	
SM354	Envelope loop execution	When ON, the PLS instruction of Y4 is executed repeatedly	R/W		√	
SM355	PLSV progressive frequency conversion	When ON, the PSLV command frequency of Y4 changes gradually, enabling the acceleration/deceleration function.	R/W		√	
SM356	Clear function is valid	Applicable to DSZR/ZRN, acting ON the axis corresponding to Y4: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W		√	
SM357	Clear signal specified element is valid	Applicable to DSZR, acting ON the axis corresponding to Y4: when set, use the Y element Y (n) corresponding to the value n in SD255 to represent the clearing signal; for reset, define Y14 as the clearing signal according to the default value	R/W		√	
SM358	Return-to-origin direction	Applicable to DSZR, acting ON the axis corresponding to Y4: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	R/W		√	
SM359	Forward limit	Applicable to DSZR/DVIT, acting ON the axis	R/W		√	

		corresponding to Y4: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached				
SM360	Inversion limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y4: when set, it indicates that the limit of the reverse direction is reached; when reset, it indicates that the limit has not been reached	R/W		√	
SM361	Near-point signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y4: when it is set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when it is reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	R/W		√	
SM362	Zero flagal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y4: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	
SM363	Interrupt signal logic inversion	Applicable to DVIT, acting ON the axis corresponding to Y4: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	
SM364	Interrupt input function specification is valid	Applicable to DVIT when the specified function of interrupt input is not used, then Y4 corresponds to the interrupt of x4. When the specified function is used, the bit is set, and then the corresponding sd256 value corresponds to the interrupt signal of Y4.	R/W		√	
SM365	User interrupt input command	Y4 is suitable for DVIT	R/W		√	
SM366	S-type acceleration and deceleration are valid	Y4 s-type acceleration and deceleration is valid; SM is ON to enable s-type acceleration and deceleration function, OFF is t-type acceleration and deceleration	R/W		√	
SM367	Dvit interrupt signal masking is valid	Y4 is suitable for DVIT SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W		√	

（6） Y5 related Signs

Address	Name	Actions and Functions	R/W	VC-B	VC-S	
SM370	Pulse output stop control	Set this element to stop the high-speed pulse output function of Y5; reset it to turn ON the output function	R/W		√	
SM371	Pulse output monitoring	Used to monitor the status of high-speed output channel Y5, ON when busy and OFF when ready	R/W		√	
SM372	PWM output valid in microseconds	When this bit is set, the PWM instruction of Y5 is output in microseconds	R/W		√	

SM373	Interrupt drive pulse output valid	When it is ON, the PLSY instruction of Y5 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W		√	
SM374	Envelope loop execution	When ON, the PLS instruction of Y5 is executed repeatedly	R/W		√	
SM375	PLSV progressive frequency conversion	When ON, the PSLV command frequency of Y5 changes gradually, enabling the acceleration/deceleration function.	R/W		√	
SM376	Clear function is valid	Applicable to DSZR/ZRN, acting ON the axis corresponding to Y5: when set, the CLR signal output function of the origin return command is valid; when reset, no CLR signal output is provided	R/W		√	
SM377	Clear signal specified element is valid	Applicable to DSZR, acting ON the axis corresponding to Y5: when set, use the Y element Y (N) corresponding to the value N in SD275 to represent the clearing signal; for reset, define Y15 as the clearing signal according to the default value	R/W		√	
SM318	Return-to-origin direction	Applicable to DSZR, acting ON the axis corresponding to Y5: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	R/W		√	
SM379	Forward limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y5: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached	R/W		√	
SM380	Inversion limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y5: when set, it indicates that the limit of the reverse direction is reached; when reset, it indicates that the limit has not been reached	R/W		√	
SM381	Near-point signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y5: when set, it is processed according to negative logic (when the input is OFF, the near-point signal is ON); when it is reset, it is processed according to positive logic (when the input is ON, the near-point signal is ON)	R/W		√	
SM382	Zero signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y5: when set, it is processed by negative logic (when the input is OFF, the Zero signal is ON); when reset, it is processed by positive logic (when the input is ON, the Zero signal is ON)	R/W		√	
SM383	Interrupt signal logic inversion	Applicable to DVIT, acting ON the axis corresponding to Y5: when it is set, it is processed by negative logic (when the input is OFF, the Zero flagal is ON); when it is reset,	R/W		√	

		it is processed by positive logic (when the input is ON, the Zero flagal is ON)				
SM384	Interrupt input function specification is valid	Applicable to DVIT When the specified function of interrupt input is not used, then Y5 corresponds to the interrupt of X5. When the specified function is used, the bit is set, and then the corresponding SD276 value corresponds to the interrupt signal of Y5.	R/W		√	
SM385	User interrupt input command	Y5 for DVIT	R/W		√	
SM386	S-type acceleration and deceleration are valid	Y5 S-type acceleration and deceleration is valid; SM is ON to enable S-type acceleration and deceleration function, OFF is T-type acceleration and deceleration	R/W		√	
SM387	DVIT interrupt signal masking is valid	Y5 is suitable for DVIT SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W		√	

（7） Y6 related Signs

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM390	Pulse output stop control	Set this component to stop the high-speed pulse output function of Y6; reset it to turn ON the output function	R/W		√	
SM391	Pulse output monitoring	Used to monitor the status of high-speed output channel Y6, ON when busy and OFF when ready	R/W		√	
SM392	Pwm output valid in microseconds	When this bit is set, the PWM instruction of Y6 is output in microseconds	R/W		√	
SM393	Interrupt drive pulse output valid	When it is ON, the plsy instruction of Y6 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W		√	
SM394	Envelope loop execution	When ON, the PLS instruction of Y6 is executed repeatedly	R/W		√	
SM395	Plsv progressive frequency conversion	When ON, the PSLV command frequency of Y6 changes gradually, enabling the acceleration/deceleration function.	R/W		√	
SM396	Clear function is valid	Applicable to DSZR/ZRN, acting ON the axis corresponding to Y6: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W		√	
SM397	Clear signal specified element is valid	Applicable to DSZR, acting ON the axis corresponding to Y6: when set, use the Y element Y (n) corresponding to the value n in SD295 to represent the clearing signal; for reset, define y16 as the clearing signal according to the default value	R/W		√	
SM398	Return-to-origin direction	Applicable to DSZR, acting ON the axis corresponding to Y6: when set, it means that the direction of origin return is forward rotation; when reset, it means that the	R/W		√	

		direction of origin return is reverse direction				
SM399	Forward limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y6: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached	R/W		√	
SM400	Inversion limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y6: when set, it means that the limit of the reverse direction has been reached; when reset, it means that the limit has not been reached	R/W		√	
SM401	Near-point signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y6: when it is set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when it is reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	R/W		√	
SM402	Zero flagal logic inversion	Applicable to DSZR, acting ON the axis corresponding to y2: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	
SM403	Interrupt signal logic inversion	Applicable to DVIT, acting ON the axis corresponding to Y6: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	
SM404	Interrupt input function specification is valid	Applicable to DVIT when the specified function of interrupt input is not used, then Y6 corresponds to the interrupt of x6. When the specified function is used, the bit is set, and then the corresponding SD296 value corresponds to the interrupt signal of Y6.	R/W		√	
SM405	User interrupt input command	Y6 for DVIT	R/W		√	
SM406	S-type acceleration and deceleration are valid	Y6 S-TYPE acceleration and deceleration is valid; SM is ON to enable s-type acceleration and deceleration function, OFF is T-TYPE acceleration and deceleration	R/W		√	
SM407	Dvit interrupt signal masking is valid	Y6 is suitable for DVIT SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W		√	

（8） Y7 related Signs

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM410	Pulse output stop control	Set this element to stop the high-speed pulse output function of Y7; reset it to turn ON the output function	R/W		√	
SM411	Pulse output monitoring	Used to monitor the status of high-speed output channel Y7, ON when busy and OFF when ready	R/W		√	

SM412	Pwm output valid in microseconds	When this bit is set, the PWM instruction of Y7 is output in microseconds	R/W		√	
SM413	Interrupt drive pulse output valid	When it is ON, the plsy instruction of Y7 can be called in the interrupt program and subprogram, and the call in the main program will be continuously and repeatedly driven with the power flow.	R/W		√	
SM414	Envelope loop execution	When ON, the PLS instruction of Y7 is executed repeatedly	R/W		√	
SM415	Plsv progressive frequency conversion	When ON, the PSLV command frequency of Y7 changes gradually, enabling the acceleration/deceleration function.	R/W		√	
SM416	Clear function is valid	Applicable to DSZR/ZRN, acting ON the axis corresponding to Y7: when set, the CLR signal output function of the origin return command is valid; when reset, the CLR signal output is not provided	R/W		√	
SM417	Clear signal specified element is valid	Applicable to DSZR, acting ON the axis corresponding to Y7: when set, use the Y element Y (n) corresponding to the value n in sd315 to represent the clearing signal; for reset, define y17 as the clearing signal according to the default value	R/W		√	
SM418	Return-to-origin direction	Applicable to DSZR, acting ON the axis corresponding to Y7: when set, it means that the direction of origin return is forward rotation; when reset, it means that the direction of origin return is reverse direction	R/W		√	
SM419	Forward limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y7: when it is set, it means that the limit of the forward rotation direction is reached; when it is reset, it means that the limit is not reached	R/W		√	
SM420	Inversion limit	Applicable to DSZR/DVIT, acting ON the axis corresponding to Y7: when it is set, it means that the limit of the reverse direction is reached; when it is reset, it means that the limit is not reached	R/W		√	
SM421	Near-point signal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y7: when set, it is processed by negative logic (when the input is OFF, the near-point signal is ON); when it is reset, it is processed by positive logic (when the input is ON, the near-point signal is ON)	R/W		√	
SM422	Zero flagal logic inversion	Applicable to DSZR, acting ON the axis corresponding to Y7: when it is set, it is processed by negative logic (when the input is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)	R/W		√	
SM423	Interrupt signal logic inversion	Applicable to DVIT, acting ON the axis corresponding to Y7: when it is set, it is processed by negative logic (when the input	R/W		√	

		is OFF, the zero flagal is ON); when it is reset, it is processed by positive logic (when the input is ON, the zero flagal is ON)				
SM424	Interrupt input function specification is valid	Applicable to DVIT when the specified function of interrupt input is not used, then Y7 corresponds to the interrupt of x7. When the specified function is used, the bit is set, and then the corresponding sd316 value corresponds to the interrupt signal of Y7.	R/W		√	
SM425	User interrupt input command	Y7 for DVIT	R/W		√	
SM426	S-type acceleration and deceleration are valid	Y7 s-type acceleration and deceleration is valid; SM is ON to enable s-type acceleration and deceleration function, OFF is t-type acceleration and deceleration	R/W		√	
SM427	Dvit interrupt signal masking is valid	Y7 is suitable for DVIT SM is ON, the interrupt signal is shielded, and the interrupt signal is allowed to be detected when it is OFF.	R/W		√	

22. Timing Output Command

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM430	Timer clock output 1	For DUTY command	R/W		√	
SM431	Timer clock output 2	For DUTY command	R/W		√	
SM432	Timer clock output 3	For DUTY command	R/W		√	
SM433	Timer clock output 4	For DUTY command	R/W		√	
SM434	Timer clock output 5	For DUTY command	R/W		√	

23. Signal Alarm

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM435	Signal alarm is valid	After SM400 is turned ON, the following SM401 and SD401 work	R/W		√	
SM436	Signal alarm action	Any action in the state S900-S999, SM401 is ON	R/W		√	

24. CANOPEN Instruction

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SM440	Canopen instruction completed		R/W		√	
SM441	Canopen command error		R/W		√	
SM442	Canopen instruction is being executed		R		√	

Appendix 2 Special Data Register

📖 Notice

1. The reserved SD is not listed in the SM table, and the read-write attribute of the reserved SD element is readable and writable (RW) by default.

1. PLC Working Status Data

Address	Name	Actions and functions	R/W	VC-B	VC-S		Scope
SD00	PLC type	VC-B: 50 VC-S: 80 VC-SM: 90 VC-P:100	R	√	√		
SD01	Version number	For example: 100 is 1.00	R	√	√		
SD02	User program capacity	For example: 8 means 8k steps program	R	√	√		
SD03	System error code	Stores the system error code that occurred	R	√	√		
SD04	Battery voltage value	In 0.1V units, 3.6V is 36	R	√	√		
SD05	Ac loss detection delay time setting value	If the set value is less than 10ms, it will be processed as 10ms; If the set value is greater than 100ms, it will be processed as 100ms; (can only be configured via the system block)	R				10-100ms
SD06	Co processor version number				√		
SD07	Number of expansion I/0 modules		R	√	√		
SD08	Number of special modules		R	√	√		
SD09	Set the input point of running control in decimal (x0 is displayed as 0, x10 is displayed as 8, and the maximum is 15) (can only be configured via the system block)		R	√	√		0-15
SD10	Main module IO points	High byte: input. Low byte: output	R	√	√		
SD11	Number of expansion module IO points	High byte: input. Low byte: output	R	√	√		
SD12	Main module analog points	High byte: input. Low byte: output	R	√	√		
SD13	The number of high-speed output channels of the main module	Low byte: number of high-speed pulse output channels	R		√		

2. Running Error Code FIFO Area

Address	Name	Actions and functions	R/W	VC-B	VC-S		Scope
SD20	Keep running error code 0	In the order of the queue, keep the 5 most recent operating	R	√	√		

SD21	Keep running error code 1	error type codes, SD20 always saves the type codes of the latest errors	R	√	√		
SD22	Keep running error code 2		R	√	√		
SD23	Keep running error code 3		R	√	√		
SD24	Keep running error code 4		R	√	√		

3. Expansion Bus Error

Address	Name	Actions and functions	R/W	VC-B	VC-S		Scope
SD25	Special module bus error module number		R	√	√		
SD26	IO module bus error module serial number		R	√	√		

4. Scan Time

Address	Name	Actions and functions	R/W	VC-B	VC-S		Scope
SD30	Current scan value	Current scan time (in units of 0.1ms)	R	√	√		
SD31	Minimum scan time	Minimum scan time (in units of 0.1ms)	R	√	√		
SD32	Maximum scan time	Maximum scan time (in units of 0.1ms)	R	√	√		
SD33	Constant scan time setpoint	The initial value is 0ms (can only be configured through the system block), in units of 1ms, when the Constant scan time is greater than the user monitoring timeout setting value, the user program timeout alarm. When a certain scan period of the user program is greater than the Constant scan, the Constant scan mode of the period will be invalid automatically, and no alarm processing will be performed. When the SD33 setting value is greater than 1000ms, it will be processed as 1000	R	√	√		0～1000ms
SD34	User program timeout setting value	The initial value is 200ms (can only be configured through the system block). When the SD34 value is less than 100, it is processed as 100; When the SD34 value is greater than 1000, it is processed as 1000	R	√	√		100～1000ms

Notice

1. SD30, SD31, SD32 have 1ms error.

2. When the Constant scan time setting value SD33 is close to the user program timeout setting value SD34, the user program timeout error is likely to occur due to the system operating conditions and user program. It is recommended that the user program timeout setting value is greater than the Constant scan time setting value. (SD33) 5ms.

5. Input Filter Constant Setting

Address	Name	Actions and functions	R/W	VC-B	VC-S	
SD35	Input filter Constant	Configurable via system block	RW	√	√	

SD36	Input filter Constant	Configurable via system block	RW	√	√	
SD37	Input filter Constant	Configurable via system block	RW	√	√	
SD38	Input filter Constant	Configurable via system block	RW	√	√	
SD39	Input filter Constant	Configurable via system block	RW	√	√	
SD40	Input filter Constant	Configurable via system block	RW	√	√	
SD41	Input filter Constant	Configurable via system block	RW	√	√	
SD42	Input filter Constant	Configurable via system block	RW	√	√	

6. Timed Interrupt Period

address	name	register content	R/W	VC-B	VC-S	
SD47	Timed interrupt 0 cycle setting value	When the value is not within the range of 1 to 32767ms, the interrupt is not triggered	R/W	√	√	
SD48	Timer interrupt 1 cycle setting value	When the value is not within the range of 1 to 32767ms, the interrupt is not triggered	R/W	√	√	
SD49	Timer interrupt 2 cycle setting value	When the value is not within the range of 1 to 32767ms, the interrupt is not triggered	R/W	√	√	

Note: When the system handles user timed interrupts, there are±1ms error, in order to ensure that the timer interrupt can work normally, it is recommended that the user set the value of the timer interrupt period to be greater than or equal to 5ms.

7. Real Time Clock

Address	Name	Register content	R/W	VC-B	VC-S		Scope
SD60	Year	For real time clock	R	√	√		2000～2099
SD61	Moon	For real time clock	R	√	√		January to december
SD62	Day	For real time clock	R	√	√		1 to 31 days
SD63	Hour	For real time clock	R	√	√		0～23 hours
SD64	Minute	For real time clock	R	√	√		0 to 59 minutes
SD65	Second	For real time clock	R	√	√		0 to 59 seconds
SD66	Week	For real time clock	R	√	√		0 (sunday) to 6 (saturday)

8. Integrated Analog Setting and Reading

Address	Name	R/W	VC-B	VC-S		Scope
SD70	Sample average of ad channel 0	R				-10000-10000
SD71	Sampling times of ad channel 0	R/W				1-1000 default is 8
SD72	Sample average of ad channel 1	R				-10000-10000
SD73	Sampling times of ad channel 1	R/W				1-1000 default is 8
SD80	Da channel 0 output value	R/W				-10000-10000

9. DHSP and DHST Instruction Usage

Address	Name	R/W	VC-B	VC-S		Scope
SD86	Dhsp table compares the upper bits of the output data	R/W	√	√		
SD87	Dhsp table compares the low-order bits of the output data	R/W	√	√		
SD88	Dhst or dhsp table to compare data high order	R/W	√	√		
SD89	Dhst or dhsp table to compare data low order	R/W	√	√		
SD90	The record number of the form currently being executed	R/W	√	√		

10. Communication Port Receiving Control and Status (COM0)

Address	Name		Register content	R/W	VC-B	VC-S		Scope
SD100	Communication port 0 mode status word	SD100.0～sd100.2 Port baud rate	B2, b1, b0 000: 38,400 baud rate 001: 19,200 baud rate 010: 9,600 baud rate 011: 4,800 baud rate 100:2,400 baud rate 101:1,200 baud rate 110:57,600 baud rate 111: 115,200 baud rate	R	√	√		
		SD100.3 Stop bit	0: 1 stop bit 1: 2 stop bits					
		SD100.4 parity	0: even parity; 1: odd parity					
		SD100.5 parity enabled	0: no verification; 1: verification					
		SD100.6 Character data bits	Data bits per character 0: 8-bit character 1: 7-bit character					
		SD100.7 Freeport receive start character mode	1: has a specific starting character 0: no specific start character					
		SD100.8 freeport receive end character mode	1: has a specific end character 0: no specific end character					

Address	Name		Register content	R/W	VC-B	VC-S		Scope
		SD100.9 Timeout between free port characters is valid	1: timeout between characters is valid 0: no inter-character timeout is valid					
		SD100.10 free port inter-frame timeout is valid	1: there is an inter-frame timeout 0: no inter-frame timeout					
		SD100.11	Reserve					
		SD100.12 High and low bytes are valid	0: the low byte of the word element is valid 1: the high and low bytes of the word element are valid					
		SD100.13～sd100.15	Reserve					
SD101	Start character			R/W	√	√		
SD102	End character			R/W	√	√		
SD103	Inter-character timeout		Default 0ms (ignore inter-character timeout)	R/W	√	√		1～32767ms
SD104	Frame timeout		Default 0ms (ignore frame timeout)	R/W	√	√		1～32767ms
SD105	Receive completion information code		Bit 0: set by the user to terminate the reception Bit 1: set when the specified end word is received Bit 2: set the maximum number of characters received Bit 3: timeout set between characters Bit 4: (frame) receive timeout set Bit 5: parity error, set Bits 6 to 15: reserved, user can ignore	R	√	√		
SD106	Character currently received			R	√	√		
SD107	The total number of characters currently received			R	√	√		
SD108	Character currently sent			R	√	√		
SD109	COM0 host station number setting			R/W	√	√		

Address	Name	Register content	R/W	VC-B	VC-S		Scope
SD110	COM0 maximum timeout setting (after sending and before receiving) ecbus additional delay.		R/W	√	√		
SD111	COM0 retries		R/W	√	√		
SD112	N:N network refresh mode (COM0 reserved)		R/W	√	√		
SD113	Error code of modbus master (COM0)		R	√	√		

11. Communication Port Receiving Control and Status (COM1)

Address	Name		Register content	R/W	VC-B	VC-S		Scope
SD120	Communication port 1 mode status word	SD120.0～SD120.2 Port baud rate	B2, b1, b0 000: 38,400 baud rate 001: 19,200 baud rate 010: 9,600 baud rate 011: 4,800 baud rate 100:2,400 baud rate 101:1,200 baud rate 110:57,600 baud rate 111: 115,200 baud rate	R	√	√		
		SD120.3 stop bit	0: 1 stop bit 1: 2 stop bits					
		SD120.4 parity	0: even parity 1: odd parity					
		SD120.5 parity enabled	0: no verification 1: check					
		SD120.6 data bits per character	Data bits per character 0: 8-bit character 1: 7-bit character					
		SD120.7 free port receive start character mode	1: has a specific starting character 0: no specific start character					
		SD120.8 freeport receive end character mode	1: has a specific end character 0: no specific end character					
		SD120.9 timeout between free port characters is valid	1: timeout between characters is valid 0: no inter-character timeout is valid					

Address	Name		Register content	R/W	VC-B	VC-S		Scope
		SD120. 10 freeport inter-frame timeout is valid	1: there is an inter-frame timeout 0: no inter-frame timeout					
		SD120.11	Reserve					
		SD120.12 High and low bytes are valid	0: the low byte of the word element is valid 1: the high and low bytes of the word element are valid					
		SD120.13～SD120.15	Reserve					
SD121	Start character			R/W	√	√		
SD122	End character			R/W	√	√		
SD123	Inter-character timeout		Default 0ms (ignore inter-character timeout)	R/W	√	√		0～32767ms
SD124	Frame timeout		Default 0ms (ignore frame timeout)	R/W	√	√		0～32767ms
SD125	Receive completion information code		Bit 0: set by the user to terminate the reception Bit 1: set when the specified end word is received Bit 2: set the maximum number of characters received Bit 3: timeout set between characters Bit 4: (frame) receive timeout set Bit 5: set when parity error occurs Bits 6 to 15: reserved, user can ignore	R	√	√		
SD126	Character currently received			R	√	√		
SD127	The total number of characters currently received			R	√	√		
SD128	Character currently sent			R	√	√		
SD129	Com1 host station number setting			R/W	√	√		
SD130	COM1 maximum timeout setting (after sending and before receiving) ecbus additional delay.			R/W	√	√		
SD131	COM1 retries			R/W	√	√		
SD132	N: n network refresh mode (COM1)			R/W	√	√		
SD133	Error code of modbus master (COM1)			R	√	√		

Address	Name	Register content	R/W	VC-B	VC-S		Scope
SD134	Modbus table command execution error command number (COM1)	When there is no communication error, the value of this component is 0	R	√	√		

12. Extended Communication Port Receiving Control and Status (COM2)

Address	Name		Register content	R/W	VC-B	VC-S		Scope
SD140	Freeport 2 mode status word	SD140.0 ～ SD140.2 Port baud rate	B2, b1, b0 000: 38,400 baud rate 001: 19,200 baud rate 010: 9,600 baud rate 011: 4,800 baud rate 100:2,400 baud rate 101:1,200 baud rate 110:57,600 baud rate 111: 115,200 baud rate	R	√	√		
		SD140.3 stop bit	0: 1 stop bit; 1: 2 stop bits					
		SD140.4 parity	0: even parity; 1: odd parity					
		SD140.5 parity enabled	0: no verification; 1: verification					
		SD140.6 data bits per character	Data bits per character 0: 8-bit character; 1: 7-bit character					
		SD140.7 free port receive start character mode	1: has a specific starting character 0: no specific start character					
		SD140.8 freeport receive end character mode	1: has a specific end character 0: no specific end character					
		SD140.9 timeout between free port characters is valid	1: timeout between characters is valid 0: no inter-character timeout is valid					
		SD140. 10 freeport inter-frame timeout is valid	1: inter-frame timeout; 0: no inter-frame timeout					
		SD140.11	Reserve	R/W	√	√		
		SD140.12 High and low bytes are valid	0: the low byte of the word element is valid 1: the high and low bytes of the word element are valid					
		SD140.13 ～ SD140.15	Reserve					

Address	Name	Register content	R/W	VC-B	VC-S		Scope
SD141	Start character		R/W	√	√		
SD142	End character		R/W	√	√		
SD143	Inter-character timeout	Default 0ms (ignore inter-character timeout)	R/W	√	√		0～32767 ms
SD144	Frame timeout	Default 0ms (ignore frame timeout)	R/W	√	√		0～32767 ms
SD145	Receive completion information code	Bit 0: set by the user to terminate the reception Bit 1: set when the specified end word is received Bit 2: set the maximum number of characters received Bit 3: timeout set between characters Bit 4: (frame) receive timeout set Bit 5: set when parity error occurs Bits 6 to 15: reserved, user can ignore	R	√	√		
SD146	Character currently received		R	√	√		
SD147	The total number of characters currently received		R	√	√		
SD148	Character currently sent		R	√	√		
SD149	COM2 host station number setting		R/W	√	√		
SD150	COM2 maximum timeout setting (after sending and before receiving) ecbus additional delay.		R/W	√	√		
SD151	COM2 retries		R/W	√	√		
SD152	N: n network refresh mode (COM2)		R/W	√	√		
SD153	Error code of modbus master (COM2)		R	√	√		
SD154	Modbus table command execution error command number (COM2)	When there is no communication error, the value of this component is 0	R	√	√		

13. High-Speed Output and Positioning Command

（1） Y0 related register

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD160 SD161	The cumulative total number of Y0 pulse output.	0	R/W	√	√	
SD162 SD163	Y0 positioning command current position	0	R/W	√	√	
SD164	Current frequency of Y0 positioning command	0	R	√	√	

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD165	(hz)					
SD166 SD167	The maximum speed when Y0 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (10-100000)	100000	R/W	√	√	
SD168	Base speed when Y0 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W	√	√	
SD169	The acceleration time (50ms-5000ms) from the base speed (sd168) to the maximum speed (SD166, SD167) when Y0 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W	√	√	
SD170	When Y0 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W	√	√	
SD171	Creep speed Y0 applies to DSZR	1000	R/W	√	√	
SD172 SD173	Origin return speed Y0 applies to DSZR	50000	R/W	√	√	
SD174	The number of segments currently executed by the PLS output command (applicable to Y0)	0	R	√	√	
SD175	Y0 clear signal device designation	0	R/W	√	√	
SD176	Dvit interrupt signal device designation (applicable to Y0)	0	R/W	√	√	

（2） Y1 related register

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD180 SD181	The cumulative total number of Y1 pulse outputs.	0	R/W	√	√	
SD182 SD183	Y1 positioning command current position	0	R/W	√	√	
SD184 SD185	Current frequency of Y1 positioning command (HZ)	0	R	√	√	
SD186 SD187	The maximum speed when Y1 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (10-100000)	100000	R/W	√	√	
SD188	The base speed when Y1 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W	√	√	
SD189	The acceleration time (50ms-5000ms) from the base speed (SD188) to the maximum speed (SD186, SD187) when Y1 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W	√	√	
SD190	When Y1 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W	√	√	
SD191	Creep speed Y1 applies to DSZR	1000	R/W	√	√	
SD192 SD193	Origin return speed Y1 applies to DSZR	50000	R/W	√	√	

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD194	The number of segments currently executed by the PLS output instruction (applicable to Y1)	0	R	√	√	
SD195	Y1 clear signal device designation	0	R/W	√	√	
SD196	Dvit interrupt signal device designation (for Y1)	0	R/W	√	√	

（3） **Y2 related register**

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD200 SD201	The cumulative total number of y2 pulse outputs.	0	R/W	√	√	
SD202 SD203	Y2 positioning command current position	0	R/W	√	√	
SD204 SD205	Y2 positioning command current frequency (HZ)	0	R	√	√	
SD206 SD207	The maximum speed when y2 executes zrn, plsv, drvi, drva, DSZR, DVIT commands (10-100000)	100000	R/W	√	√	
SD208	Base speed when y2 executes zrn, plsv, drvi, drva, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W	√	√	
SD209	The acceleration time (50ms-5000ms) from the base speed (sd208) to the maximum speed (sd206, sd207) when y2 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W	√	√	
SD210	When y2 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W	√	√	
SD211	Creep speed Y2 applies to DSZR	1000	R/W	√	√	
SD212 SD213	Origin return speed Y2 applies to DSZR	50000	R/W	√	√	
SD214	The number of segments currently executed by the PLS output instruction (applicable to Y2)	0	R	√	√	
SD215	Y2 clear signal device designation	0	R/W	√	√	
SD216	Dvit interrupt signal device designation (for Y2)	0	R/W	√	√	

（4） **Y3 related registers**

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD220 SD221	The cumulative total number of Y3 pulse output.	0	R/W		√	
SD222 SD223	Y3 positioning command current position	0	R/W		√	
SD224 SD225	Y3 positioning command current frequency (HZ)	0	R		√	
SD226 SD227	The maximum speed when Y3 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (10-100000)	100000	R/W		√	

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD228	Base speed when Y3 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W		√	
SD229	The acceleration time (50ms-5000ms) from the base speed (SD228) to the maximum speed (SD226, SD227) when Y3 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W		√	
SD230	When Y3 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W		√	
SD231	Creep speed Y3 applies to DSZR	1000	R/W		√	
SD232 SD233	Origin return speed Y3 applies to DSZR	50000	R/W		√	
SD234	The number of segments currently executed by the PLS output instruction (applicable to Y3)	0	R		√	
SD235	Y3 clear signal device designation	0	R/W		√	
SD236	Dvit interrupt signal device designation (for Y3)	0	R/W		√	

（5） Y4 related register

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD240 SD241	The cumulative total number of Y4 pulse output.	0	R/W		√	
SD242 SD243	Y4 positioning command current position	0	R/W		√	
SD244 SD245	Y4 positioning command current frequency (HZ)	0	R		√	
SD246 SD247	The maximum speed when Y4 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (10-100000)	100000	R/W		√	
SD248	Base speed when Y4 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W		√	
SD249	The acceleration time (50ms-5000ms) from the base speed (SD248) to the maximum speed (SD246, SD247) when Y4 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W		√	
SD250	When Y4 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W		√	
SD251	Creep speed Y4 applies to DSZR	1000	R/W		√	
SD252 SD253	Origin return speed Y4 applies to DSZR	50000	R/W		√	
SD254	The number of segments currently executed by the PLS output command (applicable to Y4)	0	R		√	
SD255	Y4 clear signal device designation	0	R/W		√	
SD256	Dvit interrupt signal device designation (for Y4)	0	R/W		√	

（6） Y5 related register

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD260 SD261	The accumulated total number of Y5 pulse output.	0	R/W		√	
SD262 SD263	Y5 positioning command current position	0	R/W		√	
SD264 SD265	Y5 positioning command current frequency (HZ)	0	R		√	
SD266 SD267	The maximum speed when Y5 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (10-100000)	100000	R/W		√	
SD268	Base speed when Y5 executes zrn, plsv, drvi, drva, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W		√	
SD269	The acceleration time (50ms-5000ms) from the base speed (sd268) to the maximum speed (sd266, sd267) when Y5 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W		√	
SD270	When Y5 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W		√	
SD271	Crawl speed Y5 for DSZR	1000	R/W		√	
SD272 SD273	Origin return speed Y5 applies to DSZR	50000	R/W		√	
SD274	The number of segments currently executed by the PLS output command (applicable to Y5)	0	R		√	
SD275	Y5 clear signal device designation	0	R/W		√	
SD276	Dvit interrupt signal device designation (for Y5)	0	R/W		√	

（7） Y6 related register

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD280 SD281	The accumulated total number of Y6 pulse output.	0	R/W		√	
SD282 SD283	Y6 positioning command current position	0	R/W		√	
SD284 SD285	Y6 positioning command current frequency (HZ)	0	R		√	
SD286 SD287	The maximum speed when Y6 executes zrn, plsv, drvi, drva, DSZR, DVIT commands (10-100000)	100000	R/W		√	
SD288	Base speed when Y6 executes zrn, plsv, drvi, drva, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W		√	
SD289	The acceleration time (50ms-5000ms) from the base speed (sd288) to the maximum speed (sd286, sd287) when Y6 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W		√	

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD290	When Y6 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W		√	
SD291	Crawl speed Y6 for DSZR	1000	R/W		√	
SD292	Origin return speed Y6 applies to DSZR	50000	R/W		√	
SD293						
SD294	The number of segments currently executed by the PLS output command (applicable to Y6)	0	R		√	
SD295	Y6 clear signal device designation	0	R/W		√	
SD296	Dvit interrupt signal device designation (for Y6)	0	R/W		√	

（8） Y7 related register

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD300	Y7 pulse output cumulative total number.	0	R/W		√	
SD301						
SD302	Y7 positioning command current position	0	R/W		√	
SD303						
SD304	Y7 positioning command current frequency (HZ)	0	R		√	
SD305						
SD306	The maximum speed when Y7 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (10-100000)	100000	R/W		√	
SD307						
SD308	Base speed when Y7 executes ZRN, PLSV, DRVI, DRVA, DSZR, DVIT commands (less than 1/10 of the maximum speed)	5000	R/W		√	
SD309	The acceleration time (50ms-5000ms) from the base speed (sd308) to the maximum speed (sd306, sd307) when Y7 executes ZRN, DRVI, DRVA, DSZR, DVIT commands	1000	R/W		√	
SD310	When Y7 executes ZRN, DRVI, DRVA, DSZR, DVIT commands, the deceleration time from the current speed to 0 speed (50ms-5000ms)	1000	R/W		√	
SD311	Creep speed Y7 applies to DSZR	1000	R/W		√	
SD312	Origin return speed Y7 applies to DSZR	50000	R/W		√	
SD313						
SD314	The number of segments currently executed by the PLS output command (applicable to Y7)	0	R		√	
SD315	Y7 clear signal device designation	0	R/W		√	
SD316	Dvit interrupt signal device designation (for Y7)	0	R/W		√	

14. Timing Output Command

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD330	Number of scans for timing clock output 1		R/W		√	
SD331	Number of scans for timing clock output 2		R/W		√	
SD332	Number of scans for timing clock output 3		R/W		√	
SD333	Number of scans for timing clock output 4		R/W		√	

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD334	Number of scans for timing clock output 5		R/W		√	

15. SIGNAL ALARM COMMAND

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD339	Keep the minimum number of actions in S900-S999	0	R/W		√	

16. CANOPEN Communication

Address number	Data length	Initial value	Function	R/W	VC-B	VC-S	
SD340	16	0	Configured network nodes (1-16) indicates whether it is configured1-16site, when the position is 1, indicating that the corresponding site is configured. Bit0represent1station, bit15represent16station.	R		√	
SD341	16	0	Indicates whether it is configured17-32station, when the bit is 1, it means that it is configured, and the small station is low. Bit0 represent 17 station, bit15 represent 32 station	R		√	
SD342	16	0	Network baud rate,1-8, correspond 10k, 20k, 50k, 125k, 250k, 500k, 800k, 1m	R		√	
SD343	16	0x7f	COB-ID synchronize	R		√	
SD344	16	0	Sync period (1-1000ms)	R		√	
SD345	16	0	The first address of the image area (D1000 show 1000)	R		√	
SD346	16	0	The first address of the image area (D1000 show 1000)	R			
SD350	16	0	The online node in the network, when the bit is1, which means online. 1-16number site, bit0 represent 1 station, bit15 represent 16 station.	R		√	
SD351	16	0	The online node in the network is1, which means online. 17-32station, bit0 represent 17 station, bit15 represent 32 station	R		√	
SD352	16	0	CANOPEN network status	R		√	
SD353	16	0	CANOPEN command error status	R		√	
SD354	16	0	EMCY ID	R		√	
SD355	16	0	EMCY DATA	R		√	
SD359	16		Communication error status with canopen main module,	R		√	

Address number	Data length	Initial value	Function	R/W	VC-B	VC-S	
SD360	16		Canopen master status information	R		√	
SD361 to SD392	16		CANOPEN slave 1 status information to Canopenslaves 32 status information			√	
SD400 to SD415	16		Make the CANOPEN slave data receiving area	R		√	
SD432 to SD447	16		Make CANOPEN slave data transmission area	R/W		√	

SD352 network status and errors:

Bit	Error type	Remark
Bit0	Optional module error	0: no error; 1 at least one module does not conform to the network configuration
Bit1	Required module error	0: no error; 1 at least one configuration module is no longer ON the network
Bit2	The required module has an error in network monitoring	Reserve
Bit3	Configuration process error	0: no error; 1 with error
Bit4	Network communication error	0: no error; 1 with error
Bit5	One or more slaves have errors and are not in operation	0: no error; 1 with error
Bit6	The length of the PDO received by the CANopen master is too short	0: no error; 1 with error
Bit7~bit10	Reserve	
Bit11	Whether the master is alone ON the bus	0: no; 1 yes
Bit15~bit12	Reserve	

SD359

Bit0	The PLC cannot detect the CANOPEN master. CANopen is configured, but cannot communicate	0: ok ; 1 not detected
Bit1	PLC download canopen configuration error	1 error, the configuration error occurs 3 times, set, the communication with the canopen main module stops

Bit2 Bit3	PLC data refresh error 01 An error occurred during CANOPEN data refresh 10 CANOPEN data refresh process timed out	
Bit4 Bit5	PLC reads CANOPEN master network status error	01 An error occurred 10 Timeout occurred reading status

SD360 master status and errors:

Note: When SD360 = 0x0000, it means that the master station is offline/uninitialized and cannot communicate with the PLC.

Bit0	System self-check succeeded	0: Uninitialized successfully; 1: Successful
Bit1	Network initialization/configuration start	0: Unsuccessful; 1: Startup
Bit2	An error occurred while configuring the slave	0: No error; 1: At least one module does not conform to the network configuration
Bit3	Critical error Sign	0: no error; 1: serious error, must be restarted
Bit4 Bit5 Bit6 Bit7	Error code	=0 OK =1 Download error =2 initialization error
Bit8 Bit9 Bit10 Bit11 Bit12 Bit13 Bit14	Master status	=0x01, initialize =0x02, Reset node =0x04, Reset communication =0x10, pre-operation =0x20, operation =0x30, stop
Bit15	Reserve	Reserve

Status and error of slave station

Bit	Error type	Remark
Bit0	Is the user configured	=1 configuration =0 not configured
Bit1	Slave online	=0 no such slave ON canopen network =1 has this slave
Bit2	Slave ready to start	=0 not ready =1 ready
Bit3	Slave configuration is complete	=0 configuration not complete =1 configuration complete
Bit4 Bit5 Bit6 Bit7	Error code	=0 ok bit4=1 emcy error Bit5=1 configuration error bit6=1 pdo length is too short Bit7=1 life guard or heartbeat error =f other errors = other reserved

Bit8~bit15	Slave status	=0x00 is in initialization state =0x04 is in stop state =0x7f is in pre-operational state =0x05 is in operation =0xff unknown (supervision status is configured as none)

17. Ethernet Communication

Address	Actions and functions	Initial value	R/W	VC-B	VC-S	
SD470	IP0		R		√	
SD471	IP1		R		√	
SD472	IP2		R		√	
SD473	IP3		R		√	
SD474	Ethernet slave listening port		R		√	
SD475	MAC address 0		R		√	
SD476	MAC address 1		R		√	
SD477	MAC address 2		R		√	
SD478	MAC address 3		R		√	
SD479	MAC address 4		R		√	
SD480	MAC address 5		R		√	
SD481	Communication error slave IP3		R		√	

Appendix 3 Electronic Cam Special SM Relay

Address	Name	Actions and functions	R/W	VC-S-m model
SM600	0-axis electronic gear/cam enable	Off: disable; ON enable;	RW	√
SM601	0-axis cam table unit method	Off: pulse unit; ON mechanical unit	RW	√
SM602	0-axis electronic cam cycle completion sign	Off: not completed; ON completed	RW	√
SM603	The 0-axis electronic cam generates the curve successfully	Flying shear and flying shear modification key use Generate a curve after turning ON, and automatically turn OFF	RW	√
SM604	0-axis electronic gear/cam primary phase compensation start	Turn ON and start, automatically turn OFF	RW	√
SM605	0 axis stop mode	Turn OFF to execute this cycle to stop turn ON to stop immediately	RW	√
SM607	0 axis event interrupt trigger start	High-speed comparison interrupt triggers start of electronic cam	RW	√
SM608	0 axis event interrupt trigger stop	High-speed comparison interrupt triggers stop electronic cam	RW	√
SM609~sm619reserve				
SM620	2axis electronic gear/cam enable	Off: disable; ON enable;	RW	√
SM621	2shaft cam table unit method	Off: pulse unit; ON mechanical unit	RW	√
SM622	2axis electronic cam cycle completion sign	Off: not completed; ON completed	RW	√
SM623	2the axis electronic cam generates the curve successfully	Flying shear and flying shear modification key use Generate a curve after turning ON, and automatically turn OFF	RW	√
SM624	2axis electronic gear/cam primary phase compensation start	Turn ON and start, automatically turn OFF	RW	√
SM625	2axis stop method	Turn OFF to execute this cycle to stop turn ON to stop immediately	RW	√
SM627	2-axis event interrupt trigger start	High-speed comparison interrupt triggers start of electronic cam	RW	√
SM628	2 axis event interrupt trigger stop	High-speed comparison interrupt triggers stop electronic cam	RW	√
SM629~sm639reserve				
SM640	4axis electronic gear/cam enable	Off: disable; ON enable;	RW	√

SM641	4shaft cam table unit method	Off: pulse unit; ON mechanical unit	RW	√
SM642	4axis electronic cam cycle completion sign	Off: not completed; ON completed	RW	√
SM643	4-axis electronic cam generated curve successfully	Flying shear and flying shear modification key use Generate a curve after turning ON, and automatically turn OFF	RW	√
SM644	4axis electronic gear/cam primary phase compensation start	Turn ON and start, automatically turn OFF	RW	√
SM645	4axis stop method	Turn OFF to execute this cycle to stop turn ON to stop immediately	RW	√
SM647	4-axis event interrupt trigger start	High-speed comparison interrupt triggers start of electronic cam	RW	√
SM648	4 axis event interrupt trigger stop	High-speed comparison interrupt triggers stop electronic cam	RW	√
SM649~sm659reserve				
SM660	6axis electronic gear/cam enable	Off: disable; ON enable;	RW	√
SM661	6shaft cam table unit method	Off: pulse unit; ON mechanical unit	RW	√
SM662	6axis electronic cam cycle completion sign	Off: not completed; ON completed	RW	√
SM663	6the axis electronic cam generates the curve successfully	Flying shear and flying shear modification key use Generate a curve after turning ON, and automatically turn OFF	RW	√
SM664	6axis electronic gear/cam primary phase compensation start	Turn ON and start, automatically turn OFF	RW	√
SM665	6axis stop method	Turn OFF to execute this cycle to stop turn ON to stop immediately	RW	√
SM667	6-axis event interrupt trigger start	High-speed comparison interrupt triggers start of electronic cam	RW	√
SM668	6 axis event interrupt trigger stop	High-speed comparison interrupt triggers stop electronic cam	RW	√
Note	Only flying shears and flying shears are used for successful cam generation curves. There is no need for special operations to modify key points in ordinary cams.smelement			

Appendix 4 Electronic Cam Special SD Register

Address	Actions and functions	Initial value	R/W	VC-S-m model
SD600 SD601	0-axis electronic gear/cam slave axis pulses per revolution	2000	R/W	√
SD602 SD603	0-axis electronic gear/cam movement distance from the axis of one revolution (um or 0.001 degrees)	1000	R/W	√
SD604	0-axis electronic gear molecule	1	R/W	√
SD605	0-axis electronic gear denominator	1	R/W	√
SD606	0-axis electronic cam selection table	0	R/W	√
SD607	0-axis electronic gear cam primary input source selection	0	R/W	√
SD608	0-axis electronic cam aperiodic electronic cam execution times	0	R/W	√
SD609	0-axis electronic gear/cam output mode selection	0	R/W	√
SD610 SD611	The number of pulses for the start delay of the 0-axis electronic cam unit: pulse	0	R/W	√
SD612 SD613	0 primary current position unit: pulse	0	R	√
SD614 SD615	Number of cycles executed by the 0-axis electronic cam	1	R	√
SD616 SD617	The number of pulses per revolution of the 0-axis primary	2000	R/W	√
SD618 SD619	The moving distance of the 0-axis primary in one circle (unit: um or 0.001 degrees)	1000	R/W	√
SD620	0 axis scales from axis	100	R/W	√
SD621	Reserve			√
SD622 SD623	0-axis primary phase compensation distance unit: number of pulses	0	R/W	√
SD624 SD625	0-axis primary phase compensation speed unit: pulse/s	0	R/W	√
SD626 SD627	0-axis primary phase compensation acceleration unit: pulse/s2	0	R/W	√
SD628	0-axis electronic cam/electronic gear start mode selection	0	R/W	√
SD629	0-axis electronic cam/electronic gear stop mode selection	0	R/W	√
SD630	0-axis electronic gear acceleration section distance unit: pulse	0	R/W	√
SD631	0-axis electronic gear reduction section distance unit: pulse	0	R/W	√
SD632-SD649 reserve				
SD650 SD651	2-axis electronic gear/cam slave axis pulses per revolution	2000	R/W	√
SD652 SD653	2-axis electronic gear/cam travel distance from the axis of one revolution (um or 0.001 degrees)	1000	R/W	√

SD654	2-axis electronic gear molecule	1	R/W	√
SD655	2-axis electronic gear denominator	1	R/W	√
SD656	2-axis electronic cam selection table	0	R/W	√
SD657	2-axis electronic gear cam primary input source selection	0	R/W	√
SD658	2-axis electronic cam aperiodic electronic cam execution times	0	R/W	√
SD659	2-axis electronic gear/cam output mode selection	0	R/W	√
SD660 SD661	Number of pulses for 2-axis electronic cam start delay unit: pulse	0	R/W	√
SD662 SD663	2 primary current position unit: pulse		R	√
SD664 SD665	Number of cycles executed by the 2-axis electronic cam	1	R	√
SD666 SD667	The number of pulses per revolution of the 2-axis primary	2000	R/W	√
SD668 SD669	2-axis primary movement distance in one circle (unit: um or 0.001 degrees)	1000	R/W	√
SD670	2 axis scaling from axis	100	R/W	√
SD671	Reserve			√
SD672 SD673	2-axis primary phase compensation distance unit: number of pulses	0	R/W	√
SD674 SD675	2-axis primary phase compensation speed unit: pulse/s	0	R/W	√
SD676 SD677	2-axis primary phase compensation acceleration unit: pulse/s2	0	R/W R/W	√
SD678	2-axis electronic cam/electronic gear start mode selection	0	R/W	√
SD679	2-axis electronic cam/electronic gear stop mode selection	0	R/W	√
SD680	2-axis electronic gear acceleration section distance unit: pulse	0	R/W	√
SD681	2-axis electronic gear reduction section distance unit: pulse	0	R/W	√
SD682-SD699 reserve				
SD700 SD701	4-axis electronic gear/cam slave axis pulses per revolution	2000	R/W	√
SD702 SD703	4-axis electronic gear/cam movement distance of one revolution of the slave axis (um or 0.001 degrees)	1000	R/W	√
SD704	4-axis electronic gear molecule	1	R/W	√
SD705	4-axis electronic gear denominator	1	R/W	√
SD706	4-axis electronic cam selection table	0	R/W	√
SD707	4-axis electronic gear cam primary input source selection	0	R/W	√
SD708	4-axis electronic cam aperiodic electronic cam execution times	0	R/W	√
SD709	4-axis electronic gear/cam output mode selection	0	R/W	√
SD710 SD711	Number of pulses for 4-axis electronic cam start delay unit: pulse	0	R/W	√
SD712	4 primary current position unit: pulse		R	√

SD713				
SD714	4-axis electronic cam executed cycles	1	R	√
SD715				
SD716	The number of pulses per revolution of the 4-axis primary	2000	R/W	√
SD717				
SD718	4-axis primary movement distance in one circle (unit: um or 0.001 degrees)	1000	R/W	√
SD719				
SD720	4 axis scaling from axis	100	R/W	√
SD721	Reserve			√
SD722	4-axis primary phase compensation distance unit: number of pulses	0	R/W	√
SD723				
SD724	4-axis primary phase compensation speed unit: pulse/s	0	R/W	√
SD725				
SD726	4-axis primary phase compensation acceleration unit: pulse/s2	0	R/W	√
SD727			R/W	
SD728	4-axis electronic cam/electronic gear start mode selection	0	R/W	√
SD729	4-axis electronic cam/electronic gear stop mode selection	0	R/W	√
SD730	4-axis electronic gear acceleration section distance unit: pulse	0	R/W	√
SD731	4-axis electronic gear reduction section distance unit: pulse	0	R/W	√
SD732-SD749 reserve				
SD750	6-axis electronic gear/cam slave axis pulses per revolution	2000	R/W	√
SD751				
SD752	6-axis electronic gear/cam travel distance from the axis of one revolution (um or 0.001 degrees)	1000	R/W	√
SD753				
SD754	6-axis electronic gear molecule	1	R/W	√
SD755	6-axis electronic gear denominator	1	R/W	√
SD756	6-axis electronic cam selection table	0	R/W	√
SD757	6-axis electronic gear cam primary input source selection	0	R/W	√
SD758	6-axis electronic cam aperiodic electronic cam execution times	0	R/W	√
SD759	6-axis electronic gear/cam output mode selection	0	R/W	√
SD760	The number of pulses for the start delay of the 6-axis electronic cam unit: pulse	0	R/W	√
SD761				
SD762	6 primary current position unit: pulse		R	√
SD763				
SD764	The number of cycles executed by the 6-axis electronic cam	1	R	√
SD765				
SD766	The number of pulses per revolution of the 6-axis primary	2000	R/W	√
SD767				
SD768	6-axis primary movement distance in one circle (unit: um or 0.001 degrees)	1000	R/W	√
SD769				
SD770	6-axis scaling from axis	100	R/W	√

SD771	Reserve			√
SD772	6-axis primary phase compensation distance unit: number of pulses	0	R/W	√
SD773				
SD774	6-axis primary phase compensation speed unit: pulse/s	0	R/W	√
SD775				
SD776	6-axis primary phase compensation acceleration unit: pulse/s2	0	R/W	√
SD777				
SD778	6-axis electronic cam/electronic gear start mode selection	0	R/W	√
SD779	6-axis electronic cam/electronic gear stop mode selection	0	R/W	√
SD780	6-axis electronic gear acceleration section distance unit: pulse	0	R/W	√
SD781	6-axis electronic gear reduction section distance unit: pulse	0	R/W	√
SD782-SD899 reserve				

Appendix 5 Modbus Communication Error Codes

Exception code	Exception code meaning
0x01	Illegal function code
0x02	Illegal register address
0x03	Wrong data
0x10	Communication timeout, the communication time exceeds the maximum communication time set by the user.
0x11	Receive data frame error
0x12	Parameter error, setting parameter (mode or master/slave) error
0x13	The own station number is the same as the station number set by the instruction, and an error occurs
0x14	Element address overflow (the amount of data received or sent exceeds the element storage space)
0x15	Command execution failed
0x16	The received slave address does not match the requested slave address, the detailed error code element stores the received slave address
0x17	The received function code does not match the requested function code, and the detailed error code element stores the received function code
0x18	Information frame error: currently only refers to the component start address does not match, the detailed error code component stores the received component start address
0x19	The length of the received data does not meet the protocol specification or the number of components exceeds the maximum limit specified by the function code.
0x20	CRC/LRC validation error
0x21	Reserve
0x22	Command parameter element start address setting error
0x23	The command parameter is set with an unsupported function code or an illegal function code
0x24	The number of components in the command parameter is set incorrectly
0x25	Reserve
0x26	Parameters cannot be modified during runtime
0x27	Parameters are password protected

Appendix 6 System Error Code Table

Error code	Meaning	Error type	Illustrate
0	No errors occurred		
1 to 9	System reserved		
10	SRAM error	System error	Stop user program The error light is always ON; to eliminate this error, power OFF to check the hardware;
11	FLASH error	System error	Stop user program The error light is always ON; to eliminate this error, power OFF to check the hardware;
12	Communication port error	System error	Stop user program The error light is always ON; to eliminate this error, power OFF to check the hardware;
13	Real time clock error	System error	Stop user program The error light is always ON; to eliminate this error, power OFF to check the hardware;
14	I2c error	System error	Stop user program The error light is always ON; to eliminate this error, power OFF to check the hardware;
15	FPGA configuration error	System error	Stop user program
20	Native I/O fatal error	System error	Stop user program The error light is always ON. To eliminate this error, power OFF and check the hardware
21	Extended I/O fatal error	System error	Error light blinking The error disappears, it clears automatically
22	Special module critical error	System error	Error light blinking The error disappears, it clears automatically
23	Real time clock error refresh (the time when the reading error is found when the system is refreshed)	System error	Error light blinking The error disappears, it clears automatically
24	EEPROM read and write operation error	System error	Error light flashes;
25	Local analog error	System error	Error light blinking The error disappears, it clears automatically
26	System special module configuration error	System error	Error light flashes; The error disappears, it clears automatically
27	Dual port RAM error	System error	Error light flashes;
28	CANOPEN run error	System error	Error light flashes;
29	Ethernet error	System error	Error light flashes;
40	User program file error	System error	Stop the user program (the error light is always ON) Eliminate condition: download new program/format
41	System configuration file error	System error	Stop user program Error light is always ON Eliminate condition: download new system profile/format
42	Block file error	System error	Stop the user program (the error light is always ON) Eliminate condition: download new chunk file/format

Error code	Meaning	Error type	Illustrate
43	Battery backup data loss error	System error	Does not stop the user program (error light flashes) Elimination conditions: no error detected after clearing components/formatting/resetting
44	Force table missing error	System error	Does not stop the user program (error light flashes) Elimination conditions: no error detected after clearing components/forcing operation/formatting/resetting
45	User information file error	System error	Do not stop user program (error light does not indicate) Eliminate conditions: download new program and new block file/format
46	Power failure error	System error	Stop the user program (the error light is always ON) Elimination condition: power returns to normal
45～59	Reserve		
60	User program compilation error	Execution error	Stop the user program (the error light is always ON)
61	User program running time out	Execution error	Stop the user program (the error light is always ON)
62	An illegal user program instruction was executed	Execution error	Stop the user program (the error light is always ON)
63	Illegal element type of instruction operand	Execution error	Stop the user program (the error light is always ON)
64	Illegal value of instruction operand	Execution error	Does not stop the user program execution, the error light does not indicate, but the error type code will be indicated in sd20
65	Instruction operand element number range exceeds	Execution error	
66	Subroutine stack overflow	Execution error	
67	User interrupt request queue overflow	Execution error	
68	Illegal label jump or subroutine call	Execution error	
69	Division by zero error	Execution error	
70	Illegal stack definition	Execution error	When the stack size and the number of elements in the stack are less than zero The number of elements in the stack is greater than the stack size limit
71	Reserve		
72	User subroutine or interrupt subroutine not defined	Execution error	
73	Invalid special module address	Execution error	
74	Error accessing special module	Execution error	
75	I/O immediate flush error	Execution error	
76	Wrong clock setting	Execution error	
77	PLSR instruction parameter error	Execution error	
78	Special module bfm buffer overrun	Execution error	

Error code	Meaning	Error type	Illustrate
79	ABS data reading timeout		
80	ABS data reading and verification error		
81	DSZR command enters abnormal state		
82	CANOPEN axis control command execution error	Execution error	

Appendix 7 ASCII Character Encoding Table

ASCII HEX code		High 3							
		0	1	2	3	4	5	6	7
	0	NUL	DLE	SPACE	0	@	P	, ,	P
	1	SOH	DC1	!	1	A	Q	A	Q
	2	STX	DC2	"	2	B	R	B	R
	3	ETX	DC3	#	3	C	S	C	S
	4	EOT	DC4	$	4	D	T	D	T
	5	ENQ	NAK	%	5	E	U	E	U
	6	ACK	SYN	&	6	F	V	F	V
Lower 4 bits	7	BEL	ETB	'	7	G	W	G	W
	8	BS	CAN	(	8	H	X	H	X
	9	HT	EM	)	9	I	Y	I	Y
	A	LF	SUB	*	:	J	Z	J	Z
	B	VT	ESC	+	;	K	[	K	{
	C	FF	FS	, , ,	<	L		L	|
	D	CR	GS	-	=	M	]	M	}
	E	SO	RS	.	>	N	^	N	～
	F	SI	US	/	?	O	____	O	DEL