IVC Series Micro-PLC Programming Manual

Version: V1.4

Date: 2019-01-23

INVT Auto-Control Technology Co., Ltd. provides customers with a full range of technical support services. You can contact the local office or customer service center of our company, or directly contact our headquarters.

INVT Auto-Control Technology Co., Ltd.

INVT Auto-Control Technology Co., Ltd.

Address: West Side, Floor 6, Building B, INVT Guangming Technology Building, Songbai Road, Matian, Guangming District, Shenzhen

Postal code: 518055

Company website: http://www.invt-control.com/en/index.html

Preface

- Intended readers

  This manual is intended for automation technicians to help them master the programming, system design, and commissioning of INVT programmable logic controllers (PLCs), providing reference for both new and senior learners of PLC programming.

- Content

  This manual describes the programming principles, software and hardware programming resources, supported programming languages, and detailed instructions of IVC series PLCs; provides some technical reference, such as high-speed input/output and communication information; and introduces the application methods of all functions with application instances provided.

- Arrangement

  The chapters in this manual are organized from the whole to details. The content of each chapter is independent from that of another. You can read the manual through to gradually understand IVC series PLCs to the full extent, or you can just read some chapters at any time to get some technical reference.

- Reading guide

  1. For readers unfamiliar with PLCs

     If you have never used PLCs before, it is recommended that you read chapters 1 to 4 first. These chapters explain the basics of PLCs, including function descriptions, programming languages, program elements, data types, addressing modes, soft element definitions, program annotation function and programming, application of main program and subprograms.

  2. For readers familiar with PLCs

     If you are familiar with the basic concepts and programming tools of PLCs, you can directly read Chapter 5 "Basic instructions" and Chapter 6 "Application instructions". These two chapters describe all the instructions of INVT IVC series PLCs. To understand how to use sequential function diagrams, high-speed I/O, interruption, and communication functions, refer to chapters 7 to 10. To understand the functions of positioning control, refer to Chapter 11 "Positioning function guide". In addition, Appendix K "Instruction order index table" and Appendix L "Instruction classification index table" allow you to look up descriptions of corresponding instructions by instruction type and alphabetic ordering of the English names of the instructions, respectively, which makes the reading easier.

- Related documents and references

  You can also refer to the following manuals:

  - IVC1 Series PLC User Manual
  - IVC2 Series PLC User Manual
  - IVC3 Series PLC User Manual
  - Auto Station Programming Software User Manual

Contents

Chapter 1 Product overview

This chapter briefly describes the product elements, programming software platform, and network configuration and application of IVC1, IVC2L, and IVC3 series PLCs.

1.1 Product introduction

IVC series micro-PLC products include IVC1, IVC1S, IVC1L ultra-micro series, and IVC2L and IVC3 micro series. They are high-performance products applicable to various modern industrial control applications. Main modules of the IVC3 series PLCs are configured with CAN communication interfaces and Ethernet interfaces, supporting CANopen and Modbus-TCP.

These two series are integrated PLCs with built-in high-performance microprocessors and core computing control systems, integrating input and output points, expansion module bus, etc. I/O extension modules and special modules are also included in these two series. The main module integrates 2 to 3 communication ports. IVC3 series PLCs can be directly connected to the network while other series PLC main modules can be connected to the field bus network through the fieldbus expansion modules. The I/O configured on the main modules also includes high-speed counting and high-speed pulse output channels, which can be used for precise positioning. They are equipped with abundant built-in programming resources, adopt three standardized programming languages, and can implement commissioning and monitoring through the powerful Auto Station programming software.Furthermore, optimal security protection mechanisms are provided for user programs.

1.1.1 Product performance and specifications

Table 1-1 Performance and specifications of PLC main modules

Name		IVC3	IVC2L	IVC1L	IVC1	IVC1S
I/O	Digital I/O point	16 inputs/16 outputs	20 inputs/12 outputs 32 inputs/ 32 outputs 40 inputs/ 40 outputs	8 inputs/6 outputs, 12 inputs/8 outputs, 14 inputs/10 outputs, 16 inputs/14outputs, 24 inputs/16 outputs, 36 inputs/24 outputs, 16 inputs/14outputs/2 analog inputs/1 analog output	10 inputs/6 outputs, 14 inputs/10 outputs, 16 inputs/14 outputs, 24 inputs/16 outputs, 36 inputs/24 outputs, 16 inputs/14 outputs/2 analog inputs/1 analog output	8 inputs/6 outputs, 12 inputs/8 outputs, 14 inputs/10 outputs, 16 inputs/14 outputs, 24 inputs/16 outputs, 36 inputs/24 outputs, 16 inputs/14 outputs/ 2 analog inputs/1 analog output
	Max. number of logical I/O points	512	512	128	128	60
	Max. number of special modules	8	8	7	7	None
	High-speed pulse output	8×200 kHz	2×100 kHz (for transistor output only)	3×100 kHz (for transistor output only)	2×100 kHz (for transistor output only)	2×100 kHz (for transistor output only)
	Single-phase counting channel	8×200 kHz	Six channels: two with 50 kHz / four with 10 kHz			
	Two-phase counting channel	4×200 kHz	Two channels: one with 30 kHz/ one with 5 kHz			
	Max.frequency sum of the high-speed counters	1600 kHz	80 kHz	60 kHz	60 kHz	60 kHz
	Digital filtering	Digital filtering can be separately set for each of X0 to X7. Input filtering constant range: 0–60000 us	Applying digital filtering for X0 to X7. Input filtering constant range: 0-60 ms	Applying digital filtering for X0 to X7. Input filtering constant: 0, 2, 4, 8, 16, 32, and 64 ms	Applying digital filtering for X0 to X7. Input filtering constant: 0, 2, 4, 8, 16, 32, and 64 ms	Applying digital filtering for X0 to X7. Input filtering constant: 0, 2, 4, 8, 16, 32 and 64 ms

Name			IVC3	IVC2L	IVC1L	IVC1	IVC1S
	Max. current of the relay output point	Resistive load	2 A/1 point 8 A/4 point group common terminal 8 A/8 point group common terminal				
		Inductive load	220 V AC, 80 VA				
		Lamp load	220 V AC, 100 W				
	Max. current of the transistor output point	Resistive load	Output point: 0.5 A/1 point Others: 0.5 A/1 point; 0.8 A/4 point; 1.6 A/8 point Above 8 points, the total current increases by 0.1 A for each addition point				
		Inductive load	Y0–Y7: 7.2 W/24 V DC Others: 12 W/24 V DC	Y0, Y1: 7.2 W/24 V DC Others: 12 W/24 V DC			
		Lamp load	Y0–Y7: 0.9 W/24 V DC Others: 1.5 W/24 V DC	Y0, Y1: 0.9 W/24 V DC Others: 1.5 W/24 V DC			
Storage device	User program		64 kstep (128 kB)	12 kstep (24 kB)	16 kstep (32 kB)	16 kstep (32 kB)	6 kstep (12 kB)
	Permanent storage after power outage		Yes				
	Max. number of elements for which data is saved at power outage		All the soft elements except R	User-defined	Bit element: full range Word element: 1700	Bit element: 320 Word element: 180	Bit element: full range Word element: 1700
	Hardware support and endurance time		Backup battery for storage of 3 years	Backup battery for storage of 3 years	EEPROM, permanent storage	EEPROM, permanent storage	EEPROM, permanent storage
Soft element resources	Timer		100 ms precision: T0–T209 10 ms precision: T210–T479 1 ms precision: T480–T511	100 ms precision: T0–T209 10 ms precision: T210–T251 1 ms precision: T252–T255			
	Counter		16-bit increment counter: C0–C199 32-bit increment and decrement counter: C200–C235 32-bit high-speed counter: C236–C255, C301–C306	16-bit increment counter: C0–C199 32-bit increment and decrement counter: C200–C235 32-bit high-speed counter: C236–C255			
	Data register		D0–D7999, 0–R32767	D0–D7999	D0–D7999		D0–D7999
	Local data register		V0–V63				
	Indexing register		Z0–Z15				

Name		IVC3	IVC2L	IVC1L	IVC1	IVC1S
	Special data register	SD0–SD1023	SD0–SD511	SD0–SD511	SD0–SD255	SD0–SD511
	Auxiliary relay	M0–M10239	M0–M1999	M0–M2047		M0–M2047
	Local auxiliary relay	LM0–LM63				
	Special auxiliary relay	SM0–SM1023	SM0–SM511	SM0–SM511	SM0–SM255	SM0–SM511
	State relay	S0– S4095	S0– S991	S0– S1023		S0– S1023
Interruption resource	Internal timed interruption	3	3	3		3
	External interruption	16	16	16		16
	High-speed counter-based interruption	8	6	6		6
	Serial port-based interruption	12	12	12	8	8
	Interruption after PTO output	8	2	3		2
	Interruption after interpolation	1	/	/		/
	Interruption when passing a position	8	/	/		/
	Interruption at power outage	1	1	1		1
Regular	Running time of a basic instruction	0.065μS	0.09μS	0.2μS	0.3μS	0.2μS
	Real-time clock	Supporting a minimum of uninterrupted output of 3 years after power outage	Supporting a minimum of uninterrupted output of 3 years after power outage	Supporting a minimum of uninterrupted output of 45 days after power outage	Supporting uninterrupted output of 100 hours after power outage	None
	Analog potentiometer	None	2, 8-bit precision	None	2, 8-bit precision	None
Communication	Communication port	PORT0: RS232 PORT1: RS485 PORT2: RS485 PORT3: CAN PORT4: Network port PORT5: USB	PORT0: RS232 PORT1: RS232/RS485	PORT0: RS232 PORT1: RS485 PORT2: RS485	PORT0: RS232 PORT1: RS232/RS485	PORT0: RS232 PORT1: RS485
	Communication protocol	CANopen, Modbus-TCP, Modbus, free-port, N:N, programming port protocol	Modbus, free-port, N:N, and programing port protocols			
Access control and user program protection	Password type	Upload password Download password Monitoring password Subprogram password				
	Disable upload	Supported				
	Disable formatting	Supported				

Name		IVC3	IVC2L	IVC1L	IVC1	IVC1S
	Real-time clock, clock instruction	Available	Available	Available	Available	None
	Data and clock comparison instruction	Available	Available	Available	Available	None
	Floating-point number operation instruction	Available	Available	Available	Available	None
	Positioning instruction	Available	Available	Available	Available	Available
	High-speed I/O instruction	Available	Available	Available	Available	PLS instruction is not supported
	Modbus and inverter instruction	Available	Available	Available	Available	None
	Read/write EEPROM instruction	None	Available	None	Available	None
	Control and calculation instruction	Available	Available	Available	Available	Support PID instruction only
	Character string instruction	Available	None	None	None	None
	Batch data processing instruction	Available	None	None	None	None
	Datasheet instruction	Available	None	None	None	None
	Memory card instruction	None	None	None	None	None
Mean time between failures (MTBF)	Relay output	200,000 hours, fixed on the ground, with mechanical stress approximating zero and temperature and humidity control				
		100,000 hours, fixed on the ground, with mechanical stress approximating zero, no temperature and humidity control				
	Transistor output	300,000 hours, fixed on the ground, with mechanical stress approximating zero and temperature and humidity control				
		150,000 hours, with mechanical stress approximating zero, no temperature and humidity control				
Service life of the output relay contacts	220 V AC/15 VA/sensitive	1s ON/1s OFF, 3,200,000 times				
	220 V AC /30 VA/sensitive	1s ON/1s OFF, 1,200,000 times				
	220 V AC/72 VA/sensitive	1s ON/1s OFF, 300,000 times				

Name		IVC3	IVC2L	IVC1L	IVC1	IVC1S
Power supply characteristics	Input voltage range	85 V AC–264 V AC (normal operation)				
Notes: 1. For details about product specifications, installation instructions, and operation and maintenance of IVC1 series PLCs, refer to the *IVC1 Series PLC User Manual*. 2. For details about product specifications, installation instructions, and operation and maintenance of IVC2L series PLCs, refer to the *IVC2L Series PLC User Manual*. 3. For details about product specifications, installation instructions, and operation and maintenance of IVC3 series PLCs, refer to the *IVC3 Series PLC User Manual*. 4. The backup battery can support storage of 3 years in working environments of 25 ℃.						

1.1.2 Appearance of IVC1, IVC1L, and IVC1S series main modules

Figure 1-1 shows the appearance and structure of IVC1, IVC1L, and IVC1S series main modules (using IVC1-1614MAR as an Application instance).

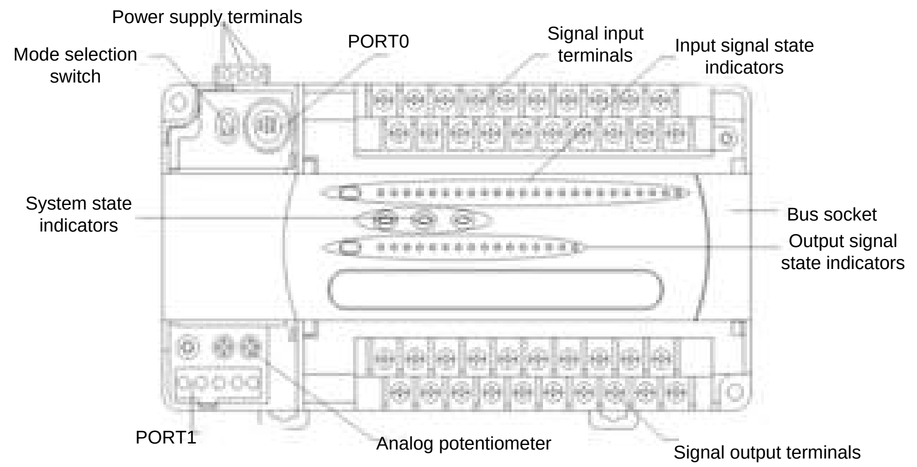


Figure 1-1 Appearance and structure of IVC1, IVC1L, and IVC1S series main modules

PORT0 and PORT1 are communication ports. PORT0 adopts the RS232 level and the Mini DIN8 socket. PORT1 of the IVC1 series adopts the RS485 or RS232 level. PORT1 and PORT2 of the IVCL series adopt the RS485 level, and PORT1 of the IVC1S series adopts the RS485 level. The bus socket is used to connect extension modules, but the IVC1S series does not support extension modules. The mode selection switch provides three options, namely ON, TM, and OFF, but IVC1S supports only two options, namely ON and OFF.

1.1.3 Appearance of IVC2L series main modules

Figure 1-2 shows the appearance and structure of IVC2L series main modules, using a 64-point main module as an Application instance.

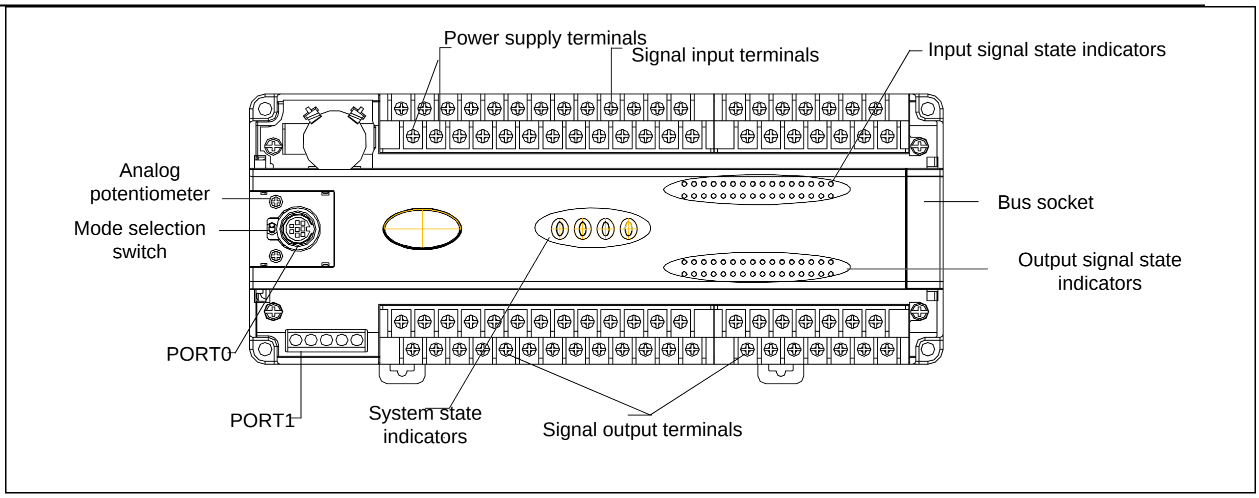


Figure 1-2 Appearance and structure of IVC2L series main modules

The battery slot is applicable to CR2354 lithium batteries. The bus socket is used to connect extension modules. PORT1 is an RS485 or RS232 wiring terminal. PORT0 adopts the RS232 leveland the Mini DIN8 socket. The mode selection switch provides three options: ON, TM, and OFF.

1.1.4 Appearance of IVC3 series main modules

Figure 1-3 shows the appearance and structure of IVC3 series main modules, using a32-point main module as an Application instance.

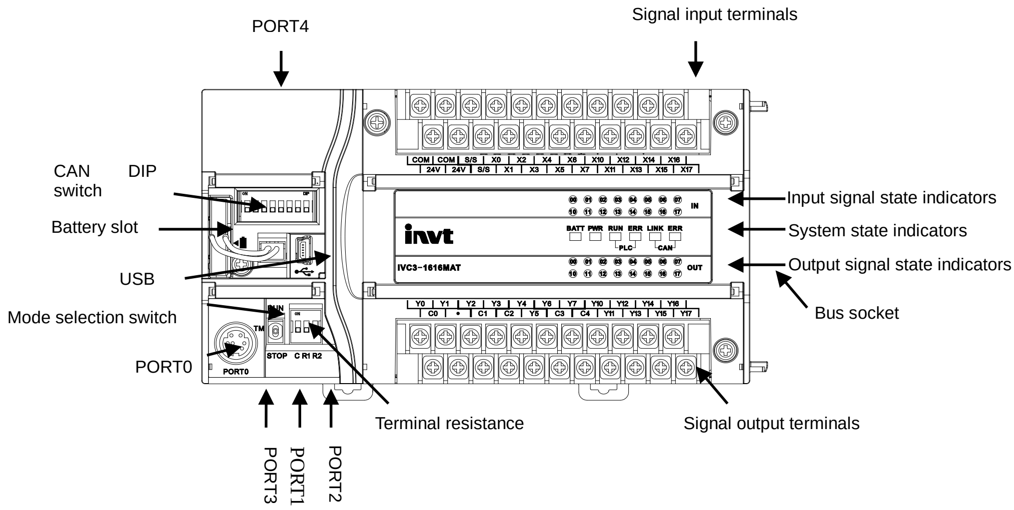


Figure 1-3 Appearance and structure of IVC3 series main modules

The bus socket is used to connect extension modules. The battery slot is applicable to CR2354 lithium batteries. The USB port can be used to upgrade the underlying CPU and FPGA programs. It provides the user program upload/download and monitoring functions, facilitating commissioning. PORT0 adopts the RS232 level and the Mini DIN8 socket.PORT1 and PORT2 are RS485 terminals. PORT3 is a CAN communication interface provided by the main module, supporting the CANopen protocol. PORT4 is an Ethernet interface, supporting Modbus-TCP. The mode selection switch provides three options: ON, TM, and OFF.

1.2 Auto Station programming software

Auto Station is programming software designed for IVC1, IVC1S, IVC1L, IVC2L, and IVC3 series PLC products. This software can be downloaded at the company website.

Auto Station programming software is standard Windows software and a graphical PLC programming tool, operated by using a mouse and keyboard. Three standard languages are available for programming: ladder diagram (LAD), instruction language (IL), and sequential function chart (SFC).

The Auto Station programming software is connected to a PLC by using a serial programming cable. Modbus network programming can also be implemented through serial port conversion, and remote programming can be implemented through Modbus. For details about Modbus programming and remote monitoring, refer to the *Auto Station Programming Software User Manual*.

1.2.1 Basic configuration

The Auto Station programming software operates on IBM PC microcomputers or compatibles, and needs to be installed ina Microsoft Windows series operating system. Compatible operating systems include Windows 98, Windows Me, NT 4.0, Windows 2000, and Windows XP.

Table 1-2 describes the minimum and recommended configuration required by Auto Station.

Table 1-2 Basic configuration conditions of Auto Station

Item	Minimum configuration	Recommended configuration
CPU	Equivalent to Intel's Pentium 233 or higher	Equivalent to Intel's Pentium 1G or higher
Memory	64 MB	128 MB
Graphics card	Can operate at the resolution of 640 x 480 in 256-color mode	Can operate at the resolution of 800 x 600 in 65535-color mode
Communication interface	An RS232 serial port provided by a DB9 type socket is required (or you can use the USB interface, but a USB-RS232 converter is required).	
Other devices	Programming cables designed for INVT PLCs	

1.2.2 Installation process of the Auto Station programming software

The Auto Station installation package released by INVT Auto-Control Technology Co., Ltd. is a standalone executable program. Double-click it to start the installation process and then install it step by step according to the installation wizard. You can choose an installation path as required.

After the installation is complete, the INVT program group appearson the Start Menu, and the installer also generates a shortcut icon of Auto Station on the desktop. You can double-click the icon to run the program.

Uninstallation: The software can be uninstalled on the Windows Control Panel. To upgrade and install a new version of Auto Station, the earlier version needs to be uninstalled first.

1.2.3 Running interface of Auto Station

The main interface of the program generally consists of seven parts: menu, toolbar, project manager window, instruction tree window, message window, status bar, and workspace.

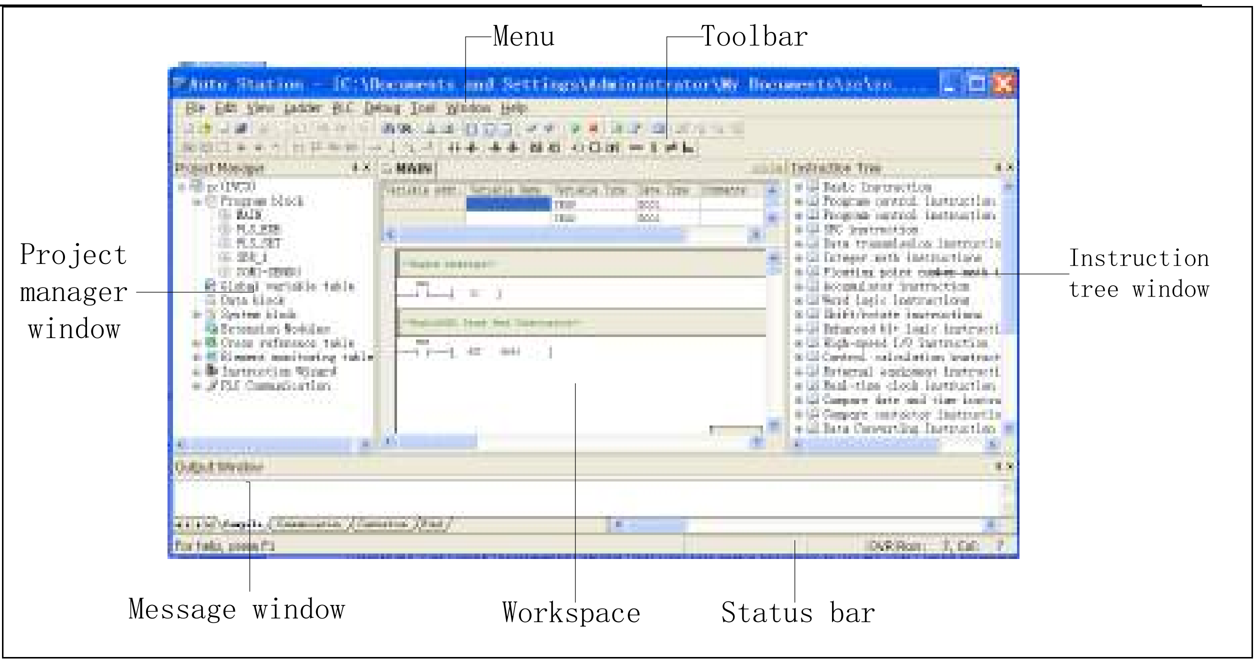


Figure 1-4 Main interface of Auto Station

Refer to the *Auto Station Programming Software User Manual* for details about how to use the Auto Station programming software.

1.2.4 Programming cable

Customers can use the serial programming cable provided by INVT Auto-Control Technology Co., Ltd. to program and perform commissioning on the PLC. There are three kinds of programming cables. One provides photo electric isolation, supporting removal and insertion under power (RUIP). The second is of the non-isolation type and does not support RUIP. The third can be used to convert USB to RS232,supporting RUIP. All these three types of programming cables do not require any jumper settings.

Figure 1-5 shows the connection of a programming cable.

Figure 1-5 Programming cable connection

1.3 Communication function

The main module of an IVC1 or IVC1S series micro-PLC integrates two serial ports: PORT0 and PORT1. The main module of an IVC1L series micro-PLC integrates three serial ports: PORT0, PORT1, and PORT2. The main module of an IVC2L series micro-PLC integrates two serial ports, namely PORT0 and PORT1, and besides the main module, there are also extension modules, such as the 485 communication module, which can be applied to fieldbus network communication. The main module of an IVC3 series micro-PLC integrates three serial ports, namely PORT0, PORT1, and PORT2; PORT3 is a CAN communication interface provided by the main module, supporting the CANopen protocol; and PORT4 is an Ethernet interface, supporting Modbus-TCP. The three serial ports adopt the Modbus and N: N communication protocols, and the user-defined free-port protocol is also applicable.

1.3.1 CANopen communication network

The standard CANopen protocol DS301 is supported. It can be applied to configure master and slave stations, facilitating networking.

1. Complying with the standard CANopen protocol DS301

2. Supporting network management (NMT) services

3. Supporting error control protocols

4. Supporting the SDO protocol

5. Supporting EDS file configuration in the CANopen configuration software

6. Supporting PDO services: supporting a maximum of 64 RxPDOs and 64 TxPDOs

7. PDO transmission type: supporting event triggering, timing triggering, synchronous cycle and synchronous non-cycle

1.3.2 Modbus-TCP communication network

Modbus TCP is an open standard protocol, and any device that complies with it can be connected to the main module of an IVC series PLC.

The remote programming port protocol is a user-defined protocol, which can be applied only through Auto Station. You can upload and download the program and monitor PLC operation remotely through the Auto Station programming software.

1.3.3 Modbus communication network

An IVC series PLC can form a Modbus RS485 network with other intelligent instruments or devices, such as inverters and PLCs, through the RS485 interfaces PORT1 and PORT2 or through PORT0 (when using PORT0, a RS232-RS485 converter is required). The maximum communication distance is 1200 meters and the highest communication baud rate is 115200 bit/s. You can select the RTU transmission mode.

An IVC series PLC can be connected to an inverter, PLC, touch screen, or instrument through the RS232 interface PORT0. The maximum communication distance is 15 meters and the highest communication baud rate is 115200 bit/s.

For details about Modbus communication,refer to Chapter 10"Communication function guide" and Appendix G "Modbus communication protocols".

1.3.4 N:N communication network

IVC1, IVC1L, IVC2L, and IVC3 series PLCsare configured with the N:N communication protocol developed by INVT, which can be applied to form an N:N communication network through the RS485 interfaces PORT1 and PORT2. With the conversion of RS232 to RS485, the PORT0 port can also be used for N:N network connection.

The N:N communication protocol allows 2 to 32 PLC stations to exchange data with each other. The highest communication baud rate is 115200 bps. This protocol can be applied to form a single-layer or double-layer network.

For details about N:N communication, refer to Chapter 10 "Communication function guide".

1.3.5 Free-port network

The free-port protocol is a protocol mode in which the communication is performed based on user-defined communication data formats. It supports two data formats: ASCII and binary. In this communication port mode, a PLC can be used to communicate with various devices that use user-defined formats, such as inverters, barcode scanners, instruments, and other intelligent devices that adopt free communication protocols. A PLC can communicate with one device in RS232 or RS485 mode, and it can also form an RS485 network with multiple devices.

For details about the free-port protocol communication, refer to Chapter 10 "Communication function guide".

1.4 IVC series micro-PLC manuals

IVC series micro-PLC manuals can be downloaded at the website of INVT Auto-Control Technology Co., Ltd. For printed versions, contact the local dealer.

1.4.1 Model selection manuals

IVC1 Model Selection Manual

IVC2L Technical Manual

IVC3 Model selection manual

1.4.2 Main module user manuals

IVC1 series
IVC1 Series PLC Quick Start User Manual
IVC1 Series PLC User Manual

IVC1S series
IVC1S Series PLC Quick Start User Manual
IVC1S Series PLC User Manual

IVC1L series
IVC1L Series PLC Quick Start User Manual
IVC1L Series PLC User Manual

IVC2L series
IVC2L Series PLC Quick Start User Manual
IVC2L Series PLC User Manual

IVC3 series
IVC3 Series PLC Quick Reference Manual
IVC3 Series PLC User Manual

1.4.3 Programming manual

IVC Series Micro-PLC Programming Manual

1.4.4 Programming software user manual

Auto Station Programming Software User Manual

1.4.5 I/O extension module user manuals

IVC1 series
IVC1 Series Passive I/O Extension Module User Manual

IVC3 series
IVC3 Series Passive I/O Extension Module User Manual

IVC2L series
IVC2L Series Passive I/O Extension Module User Manual
IVC2L Series Active I/O Extension Module User Manual

1.4.6 Special module user manuals

IVC1 series
IVC1-4AD Analog Input Module User Manual
IVC1-2DA Analog Output Module User Manual
IVC1-4DA Analog Output Module User Manual
IVC1-2PT RTD Temperature Input Module User Manual
IVC1-4PT RTD Temperature Input Module User Manual
IVC1-2TC Thermocouple Temperature Input Module User Manual
IVC1-4TC Thermocouple Temperature Input Module User Manual
IVC2L series
IVC2L-4AD Analog Input Module User Manual
IVC2L-4AM Analog I/O Module User Manual
IVC2L-4DA Analog Output Module User Manual

IVC2L series
IVC2L-4PT RTD Temperature Input Module User Manual
IVC2L-4TC Thermocouple Temperature Input Module User Manual
IVC2L-8AD Analog Input Module User Manual
IVC2L-8TC Thermocouple Temperature Input Module User Manual
IVC3 series
IVC3-4AD Analog Input Module User Manual
IVC3-4AM Analog I/O Module User Manual
IVC3-4DA Analog Output Module User Manual
IVC3-4PT RTD Temperature Input Module User Manual
IVC3-4TC Thermocouple Temperature Input Module User Manual

1.4.7 Communication module user manuals

IVCS-EPM Communication Module User Manual

IVC2L-RS485 Communication Module User Manual

IVC2L-COPM Communication Module User Manual

Chapter 2 Function Description

This chapter briefly describes the programming resources and principles and system configuration method of IVC series PLCs, how to set the PLC running and operation mode, and the various functions and software tools related to system commissioning. For details, refer to the *Auto Station Programming Software User Manual*.

2.1 Programming resources and principles

2.1.1 Programming resources

Table 2-1IVC1 programming resources

Name		Specification and description
I/O Configuration	Max. number of I/O points	128 (theoretical value)
	Number of extension modules	The total number of I/O extension modules and special modules does not exceed seven.
User file capacity	User program capacity	16 ksteps
	Block size	8000 D elements
Instruction speed	Basic instruction	0.3 us/instruction
	Application instruction	A few us – hundreds us /instruction
Number of instructions	Basic instruction	32
	Application instruction	226
Soft element resources[note 7]	I/O points	128 inputs/128 outputs (inputs: X0–X177; outputs: Y0–Y177)[note 1]
	Auxiliary relay	2048(M0–M2047)
	Local auxiliary relay	64 (LM0–LM63)
	Special auxiliary relay	256 (SM0–SM255)
	State relay	1024 (S0–S1023)
	Timer	256 (T0–T255)[note 2]
	Counter	256 (C0–C255)[note 3]
	Data register	8000 (D0–D7999)
	Local data register	64 (V0–V63)
	Indexing register	16 (Z0–Z15)
	Special data register	256 (SD0–SD255)
Interruption resources	External input-based interruption	16 (interruption triggering edges can be defined by users, corresponding to the rising and falling edges of the X0 to X7 terminals)
	High-speed counter-based interruption	6
	Internal timed interruption	3
	Serial port-based interruption	8
	Interruption after PTO output	2
	Interruption at power outage	1
Communication function	Communication interface	Two asynchronous serial communication ports: PORT0: RS232 PORT1: RS232 or RS485
	Communication protocol	Modbus, free-port, N:N (INVT proprietary protocol, which can be applied to form 1:N or N:N communication networks)

Name		Specification and description	
Special function	High-speed counter	X0, X1	Single input: 50 kHz Simultaneous inputs of X0 to X5: The total frequency is no more than 80 kHz.
		X2–X5	Single input: 10 kHz
	High-speed pulse output	Y0, Y1	Two independent outputs: 100 kHz (for transistor output type only)
	Digital filtering	Applying digital filtering for X0–X7 and hardware filtering for other ports	
	Analog potentiometer[note 4]	2	
	Subprogram call	A maximum of 64 user subprograms and six levels of nested subprograms are allowed. Local variables are supported, and each subprogram can provide a maximum of 16 parameters for information transmission. Variable aliases are also supported.	
Special function	User program protection	Upload password	Three types of passwords are provided. A password is no more than 8 characters, and each character is alphanumeric and case sensitive.
		Download password	
		Monitoring password	
		Subprogram encryption	A password is no more than 16 characters, and each character is alphanumeric and case sensitive.
		Other protective measures	The functions of disabling formatting and upload are provided.
	Programming mode[note 5]	Auto Station programming software[note 6]	It needs to be installed and run on IBM PC microcomputer or compatibles.
	Real-time clock	Built-in, supporting uninterrupted output of 100 hours after power outage (to implement this function, the main module needs to continuously work for more than two minutes before power outage)	

Table 2-2 IVC1S programming resources

Name		Specification and description
I/O Configuration	Max. number of I/O points	60
	Number of extension modules	None
User file capacity	User program capacity	6 ksteps
	Block size	8000 D elements
Instruction speed	Basic instruction	0.2 us/instruction
	Application instruction	A few us – hundreds us /instruction
Number of instructions	Basic instruction	32
	Application instruction	200
Soft element resources[note 7]	I/O points	128 inputs/128 outputs (inputs: X0–X177; outputs: Y0–Y177)[note 1]
	Auxiliary relay	2048(M0–M2047)
	Local auxiliary relay	64 (LM0–LM63)
	Special auxiliary relay	512 (SM0–SM511)
	State relay	1024 (S0–S1023)
	Timer	256 (T0–T255)[note 2]
	Counter	256 (C0–C255)[note 3]
	Data register	8000 (D0–D7999)
	Local data register	64 (V0–V63)
	Indexing register	16 (Z0–Z15)
	Special data register	512 (SD0–SD512)

Name		Specification and description	
Interruption resources	External input-based interruption	16 (interruption triggering edges can be defined by users, corresponding to the rising and falling edges of the X0 to X7 terminals)	
	High-speed counter-based interruption	6	
	Internal timed interruption	3	
	Serial port-based interruption	4	
	Interruption after PTO output	2	
	Interruption at power outage	1	
Communication function	Communication interface	Two asynchronous serial communication ports: PORT0: RS232 PORT1: RS485	
	Communication protocol	Modbus, free-port, programming port, and N:N protocols	
Special function	High-speed counter	X0, X1	Single input: 10 kHz Simultaneous inputs of X0 to X5: The total frequency is no more than 80 kHz.
		X2–X5	Single input: 10 kHz
	High-speed pulse output	Y0, Y1	Two independent outputs: 100 kHz (for transistor output type only)
	Digital filtering	Applying digital filtering for X0–X7 and hardware filtering for other ports	
	Analog potentiometer[note 4]	2	
	Subprogram call	A maximum of 64 user subprograms and six levels of nested subprograms are allowed. Local variables are supported, and each subprogram can provide a maximum of 16 parameters for information transmission. Variable aliases are also supported.	
	User program protection	Upload password	Three types of passwords are provided. A password is no more than 8 characters, and each character is alphanumeric and case sensitive.
		Download password	
		Monitoring password	
		Subprogram encryption	A password is no more than 16 characters, and each character is alphanumeric and case sensitive.
		Other protective measures	The functions of disabling formatting and upload are provided.
	Programming mode[note 5]	Auto Station programming software[note 6]	It needs to be installed and run on IBM PC microcomputer or compatibles.
	Real-time clock	Built-in, supporting uninterrupted output of 100 hours after power outage (to implement this function, the main module needs to continuously work for more than two minutes before power outage)	

Table 2-3 IVC1L programming resources

Name		Specification and description
I/O Configuration	Max. number of I/O points	128 (theoretical value)
	Number of extension modules	The total number of I/O extension modules and special modules does not exceed seven.
User file capacity	User program capacity	16 ksteps
	Block size	8000 D elements
Instruction	Basic instruction	0.2 us/instruction

Name		Specification and description	
speed	Application instruction	A few us – hundreds us /instruction	
Number of instructions	Basic instruction	32	
	Application instruction	234	
Soft element resources[note 7]	I/O points	128 inputs/128 outputs (inputs: X0–X177; outputs: Y0–Y177)[note 1]	
	Auxiliary relay	2048 (M0–M2047)	
	Local auxiliary relay	64 (LM0–LM63)	
	Special auxiliary relay	512 (SM0–SM511)	
	State relay	1024 (S0–S1023)	
	Timer	256 (T0–T255)[note 2]	
	Counter	256 (C0–C255)[note 3]	
	Data register	8000 (D0–D7999)	
	Local data register	64 (V0–V63)	
	Indexing register	16 (Z0–Z15)	
	Special data register	512 (SD0–SD512)	
Interruption resources	External input-based interruption	16 (interruption triggering edges can be defined by users, corresponding to the rising and falling edges of the X0 to X7 terminals)	
	High-speed counter-based interruption	6	
	Internal timed interruption	3	
	Serial port-based interruption	12	
	Interruption after PTO output	4	
	Interruption at power outage	1	
Communication function	Communication interface	Three asynchronous serial communication ports: PORT0: RS232 PORT1: RS485 PORT2: S485	
	Communication protocol	Modbus, free-port, N:N (INVT proprietary protocol, which can be applied to form 1:N or N:N communication networks)	
Special function	High-speed counter	X0, X1	Single input: 50 kHz Simultaneous inputs of X0 to X5: The total frequency is no more than 80 kHz.
		X2–X5	Single input: 10 kHz
	High-speed pulse output	Y0, Y1, Y2	Three independent outputs: 100 kHz (for transistor output type only)
	Digital filtering	Applying digital filtering for X0–X7 and hardware filtering for other ports	
	Analog potentiometer[note 4]	None	
	Subprogram call	A maximum of 64 user subprograms and six levels of nested subprograms are allowed. Local variables are supported, and each subprogram can provide a maximum of 16 parameters for information transmission. Variable aliases are also supported.	

<table>
<tr><th colspan="2">Name</th><th colspan="2">Specification and description</th></tr>
<tr><td rowspan="7">Special function</td><td rowspan="5">User program protection</td><td>Upload password</td><td rowspan="3">Three types of passwords are provided. A password is no more than 8 characters, and each character is alphanumeric and case sensitive.</td></tr>
<tr><td>Download password</td></tr>
<tr><td>Monitoring password</td></tr>
<tr><td>Subprogram encryption</td><td>A password is no more than 16 characters, and each character is alphanumeric and case sensitive.</td></tr>
<tr><td>Other protective measures</td><td>The functions of disabling formatting and upload are provided.</td></tr>
<tr><td>Programming mode[note 5]</td><td>Auto Station programming software[note 6]</td><td>It needs to be installed and run on IBM PC microcomputer or compatibles.</td></tr>
<tr><td>Real-time clock</td><td colspan="2">Built-in, and power supply by backup battery.</td></tr>
</table>

Table 2-4 IVC2L programming resources

<table>
<tr><th colspan="2">Name</th><th colspan="3">Specification and description</th></tr>
<tr><td rowspan="2">I/O Configuration</td><td>Max. number of I/O points</td><td colspan="3">512 points (256 inputs/256 outputs)</td></tr>
<tr><td>Number of extension modules</td><td colspan="3">8 modules, and the total number of special modules cannot exceed 8</td></tr>
<tr><td rowspan="2">User file capacity</td><td>User program capacity</td><td colspan="3">12 ksteps</td></tr>
<tr><td>Block size</td><td colspan="3">8000 D elements</td></tr>
<tr><td rowspan="2">Instruction speed</td><td>Basic instruction</td><td colspan="3">0.09 us/instruction</td></tr>
<tr><td>Application instruction</td><td colspan="3">5 us–280 us/instruction</td></tr>
<tr><td rowspan="2">Number of instructions</td><td>Basic instruction</td><td colspan="3">32</td></tr>
<tr><td>Application instruction</td><td colspan="3">221</td></tr>
<tr><td rowspan="11">Soft element resources[note 7]</td><td>I/O points</td><td colspan="3">256 inputs/256 outputs (inputs: X0–X377; outputs: Y0–Y377)[note 1]</td></tr>
<tr><td>Auxiliary relay</td><td colspan="3">2000 (M0–M1999)</td></tr>
<tr><td>Local auxiliary relay</td><td colspan="3">64 (LM0–LM63)</td></tr>
<tr><td>Special auxiliary relay</td><td colspan="3">256 (SM0–SM255)</td></tr>
<tr><td>State relay</td><td colspan="3">992 (S0–S991)</td></tr>
<tr><td>Timer</td><td colspan="3">256 (T0–T255)[note 2]</td></tr>
<tr><td>Counter</td><td colspan="3">256 (C0–C255)[note 3]</td></tr>
<tr><td>Data register</td><td colspan="3">8000 (D0–D7999)</td></tr>
<tr><td>Local data register</td><td colspan="3">64 (V0–V63)</td></tr>
<tr><td>Indexing register</td><td colspan="3">16 (Z0–Z15)</td></tr>
<tr><td>Special data register</td><td colspan="3">256 (SD0–SD255)</td></tr>
<tr><td rowspan="6">Interrupt resources</td><td>External input-based interruption</td><td colspan="3">16 (interruption triggering edges can be defined by users, corresponding to the rising and falling edges of the X0 to X7 terminals)</td></tr>
<tr><td>High-speed counter-based interruption</td><td colspan="3">6</td></tr>
<tr><td>Internal timed interruption</td><td colspan="3">3</td></tr>
<tr><td>Serial port-based interruption</td><td colspan="3">2</td></tr>
<tr><td>Interruption after PTO output</td><td colspan="3">12</td></tr>
<tr><td>Interruption at power outage</td><td colspan="3">1</td></tr>
<tr><td rowspan="2">Communication function</td><td>Communication interface</td><td colspan="3">Two asynchronous serial communication ports:
PORT0: RS232
PORT1: RS232 or RS485
PORT2 (externally connected to 485 communication modules): RS422 or RS485</td></tr>
<tr><td>Communication protocol</td><td colspan="3">Modbus and free-port protocols can be applied to form 1:N networks</td></tr>
<tr><td rowspan="8">Special function</td><td rowspan="2">High-speed counter</td><td>X0, X1</td><td colspan="2">Single input: 50 kHz
Simultaneous inputs of X0 to X5: The total frequency is no more than 80 kHz.</td></tr>
<tr><td>X2–X5</td><td colspan="2">Single input: 10 kHz</td></tr>
<tr><td>High-speed pulse output</td><td>Y0, Y1</td><td colspan="2">Two independent outputs: 100 kHz (for transistor output type only)</td></tr>
<tr><td>Digital filtering</td><td colspan="3">Applying digital filtering for X0–X7 and hardware filtering for other ports</td></tr>
<tr><td>Analog potentiometer[note 4]</td><td colspan="3">2</td></tr>
<tr><td>Subprogram call</td><td colspan="3">A maximum of 64 user subprograms and six levels of nested subprograms are allowed. Local variables are supported, and each subprogram can provide a maximum of 16 parameters for information transmission. Variable aliases are also supported.</td></tr>
<tr><td rowspan="2">User program protection</td><td colspan="2">Upload password</td><td rowspan="2">Three types of passwords are provided. A password is no more than 8 characters, and each character is alphanumeric and case</td></tr>
<tr><td colspan="2">Download password</td></tr>
</table>

Name		Specification and description	
		Monitoring password	sensitive.
		Other protective measures	The functions of disabling formatting and upload are provided.
	Programming mode[note 5]	Auto Station programming software[note 6]	It needs to be installeded and run on IBM PC microcomputer or compatibles.
	Real-time clock	Built-in, and power supply by backup battery.	

Table 2-5 IVC3 programming resources

Name		Specification and description
I/O Configuration	Max. number of I/O points	512 points (256 inputs/256 outputs)
	Number of special function modules	8 modules, and the total number of special modules cannot exceed 8
User file capacity	User program capacity	64 ksteps
	Block size	8000 D elements, 32k R elements
Instruction speed	Basic instruction	0.065 us /instruction
	Application instruction	A few us – hundreds us /instruction
Number of instructions	Basic instruction	32
	Application instruction	286
Soft element resources[note 7]	I/O points	512 inputs/512 outputs (inputs: X0–X777; outputs: Y0–Y777)[note 1]
	Auxiliary relay	10240 (M0–M0239)
	Local auxiliary relay	64 (LM0–LM63)
	Special auxiliary relay	1024 (SM0–SM1023)
	State relay	4096 (S0–S4095)
	Timer	512 (T0–T511)[note 2]
	Counter	307 (C0–C306)[note 3]
	Data register	40768 (D0–D7999, R0–R32767)
	Local data register	64 (V0–V63)
	Indexing register	16 (Z0–Z15)
	Special data register	1024 (SD0–SD1023)
Interrupt resources	External input-based interrupt	16 (interruption triggering edges can be defined by users, corresponding to the rising and falling edges of the X0 to X7 terminals)
	High-speed counter-based interrupt	8
	Internal timed interrupt	3
	Serial port-based interrupt	12
	Interrupt after PTO output	8
	Interrupt at power outage	1
	Position-based interrupt 0 of high-speed output	8
	Interpolation interrupt	1
Communication function	Communication interface	Three asynchronous serial communication ports: PORT0: RS232 PORT1: RS485 PORT1: RS485 1 CAN communication port 1 Ethernet communication port 1 USB port

<table>
<tr><th colspan="2">Name</th><th colspan="3">Specification and description</th></tr>
<tr><td></td><td>Communication protocol</td><td colspan="3">Modbus, free-port, N:N (INVT proprietary protocol, which can be applied to form 1:N or N:N communication networks)
CANopen and Modbus TCP communication protocols</td></tr>
<tr><td rowspan="10">Special function</td><td>High-speed counter</td><td colspan="3">X0–X7, 8×200 kHz</td></tr>
<tr><td>High-speed pulse output</td><td>Y0–Y7</td><td colspan="2">8×200 kHz</td></tr>
<tr><td>Digital filtering</td><td colspan="3">Applying digital filtering for X0–X7 and hardware filtering for other ports</td></tr>
<tr><td>Subprogram call</td><td colspan="3">A maximum of 64 user subprograms and six levels of nested subprograms are allowed. Local variables are supported, and each subprogram can provide a maximum of 16 parameters for information transmission. Variable aliases are also supported.</td></tr>
<tr><td rowspan="4">User program protection</td><td colspan="2">Upload password</td><td rowspan="3">Three types of passwords are provided. A password is no more than 8 characters, and each character is alphanumeric and case sensitive.</td></tr>
<tr><td colspan="2">Download password</td></tr>
<tr><td colspan="2">Monitoring password</td></tr>
<tr><td colspan="2">Other protective measures</td><td>The functions of disabling formatting and upload are provided.</td></tr>
<tr><td>Programming mode[note 5]</td><td colspan="2">Auto Station programming software[note 6]</td><td>It needs to be installeded and run on IBM PC microcomputer or compatibles.</td></tr>
<tr><td>Real-time clock</td><td colspan="3">Built-in, and power supply by backup battery.</td></tr>
</table>

Notes:

Note 1: The addresses of the X and Y elements are numbered in octal, for Application instance: Address X10 represents the 8th input point.

Note 2: Addresses of the T elements are classified into three categories according to timing precision:

IVC1/IVC1S/ IVC1L/IVC2L

(1) 100 ms precision: T0 – T209

(2) 10 ms precision: T210 – T251

(3) 1 ms precision: T252 – T255

IVC3

(1) 100 ms precision: T0 – T209

(2) 10 ms precision: T210 – T479

(3) 1ms precision: T480 – T511

Note 3: Addresses of the C elements are classified into three categories according to the width and function of count values:

IVC1/IVC1S/IVC2L

(1) 16-bit increment counter: C0 – C199

(2) 32-bit increment and decrement counter: C200 – C235

(3) 32-bit high-speed counter: C236 – C255

IVC3

(1) 16-bit increment counter: C0 – C199

(2) 32-bit increment and decrement counter: C200 – C235

(3) 32-bit high-speed counter: C236 – C255, C301-C307, C256-C300 reserved

Note 4: The analog potentiometer is an external channel for usersto set the values of the internal soft elements. The values range from 0 to 255 and can be read by the user program. You can use a Phillips screwdriver to adjust the setting by turning the analog potentiometer in the clockwise direction. The setting changes from small to large and its maximum rotation angle is 270°. Do not turn the analog potentiometer too hard. Otherwise, it may be damaged.

Note 5: The element forcing function is provided to facilitate commissioning and user program analysis, and thus improve commissioning efficiency. A maximum of 128 bit elements and 16 word elements can be forced simultaneously.

Note 6: The user program can be modified online when the PLC is running.

Note 7: Some internal soft element resources of PLCs have been reserved for internal use. Do not use these elements on the user program, if possible. For details, refer to Appendix C"Reserved elements".

2.1.2 PLC operating mechanism (Scan cycle model)

IVC series PLC main modules operate according to the scan cycle model.

The system performs four tasks in the sequential and cyclical manner: executing the user program, communication, housekeeping, and refreshing I/O. Each round of tasks is called a scan cycle.

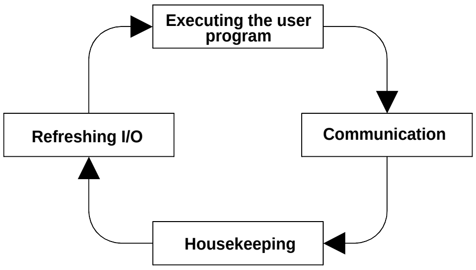


Figure 2-1PLC operating mechanism

- Executing the user program
  The system sequentially executes the instructions of the user program, starting from the first instruction to the end instruction of the main program.
- Communication
  The system communicates with the programming software and responds to the programming communication instructions, such as the download, run, and stop instructions sent by the programming software.
- Housekeeping
  The system handles various system housekeeping tasks such as refreshing panel indicators, updating timing value of the software timer, refreshing special auxiliary relaysand special data registers.
- Refreshing I/O
  I/O refresh includes an output refresh phase and an input refresh phase.
  Output refresh phase: Switching the corresponding hardware output point on or off according to the value of the Y element (ON or OFF).
  Input refresh phase: Converting the on/off state of the hardware input point to the corresponding X element value (ON or OFF).

2.1.3 User program running watchdog

The system monitors the running time of the user program in each scan cycle. Once detecting that the running time of the user program exceeds the set value, the system stops the user program. You can set the watchdog time on the **Set Time** tab of the system block dialog boxon the Auto Station background software interface.

2.1.4 Constant scan operation mode

In the constant scan operation mode, the time of each scan cycle is the same when the system is running. You can activate the constant scan mode and set the constant scan time on the **Set Time** tab of the system block dialog box of the Auto Station background software interface. The default constant scan cycle is 0, that is, constant scan is disabled. When the actual scan cycle is longer than the constant scan cycle, the program runsin the actual scan cycle.

Note

The setting of the constant scan time cannot be longer than the watchdog time.

2.1.5 User file download and storage

You can program and control a main module by downloading a specific user file to it.

There are four types of user files: user program files, data block files, system block files, and user assistance information files. User assistance information files includeglobal variable tables and user data source files.

You can choose to download a user program file, data block file, or system block file. When a file is selected, the corresponding user assistance information file is bundled and downloaded.

After downloading a user program file, data block file, or system block file of the IVC2L series, it iswritten into the EEPROM area of the main module for permanent storage. The corresponding user assistance information file is stored in the battery backup RAM area.

All user files of the IVC1 series are written into the FLASH area of the main module for permanent storage.

After downloading a user program file, data block file, and system block file of the IVC3 series, it is written into the FLASH and EEPROM areas of the main module for permanent storage. The corresponding user assistance information file is stored in the battery backup RAM area.

Note

1. Ensure that the power supply of the main module works properly in a period of time (longer than 30s) after the file is downloaded, so that the file can be properly written into the main module.

2. The failure of the IVC2L and IVC3 series PLC backup batteries results in the loss of user program auxiliary information files, and thus the related an notation of the user program cannot be properly uploaded. In this case, the system reports a **User information file error**, but does not affect the execution of the user program.

2.1.6 Element initialization

When the PLC enters the running state (STOP→RUN), the device initializes the related soft elements according to the data saved at power outage, data stored on EEPROM, data blocks, and element values. Table 2-6 describes the priorities of the data.

Table 2-6 PLC data initialization priorities

Storage device type	Power OFF→ON	STOP→RUN
Data saved at power outage	Highest	Highest
Data storage on EEPROM	High	High
Data blocks (when "Datablock enabled" is selected in the advanced settings of the system block)	Medium	Medium
Element values (when "Element value retained" is selected in the advanced settings of the system block)	—	Low

2.1.7 Saving data at power outage

- Conditions for saving data at power outage
  When detecting that a power outage occurs, the system stops the user program and saves the data values of the elements set within the storage range in the system block to the power outage backup file.
- Element recovery at power-on
  If the power outage backup file is correct after power-on, the values of the specified soft elements are restored to the values saved at the last power outage.
  After power-on, the system deletes the data values of the elements that are out of the storage range.
  If the backup file is lost or incorrect, data of all elements are deleted.
- Storage range setting

The range of elements for which data values are to be saved can be set in the **Saving Range** of the system block, as shown in Figure 2-2.

The storage range of the IVC1 and IVC1L series can be set only to a group.

The storage range of the IVC2L and IVC3 series can be set to two groups, and the actual range is the union of these two groups.

Application instance (IVC2L):

The first group is set to M100 – M200.

The second group is set to M300 – M400.

Therefore, bothM100 – M200 and M300 – M400 are elements for which data values are to be saved.

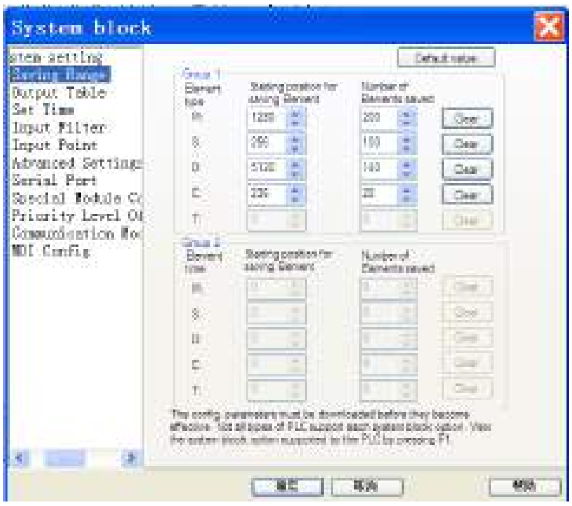


Figure 2-2 Setting the saving range

Note

1. The function of IVC2L and IVC3 series PLCs for saving data at power outage needs the support of PLC backup batteries. Ensure that the backup battery works properly. Otherwise, data values of all elements are unknown after power outage.

2. When an IVC1 series PLC encounters a power outage, data of its elements set within the storage range is stored in EEPROM.

2.1.8 Permanent storage of D element data

For the IVC1 series, you can store D element data (D6000 – D6999) in EEPROM by using the EROMWR instructionson the user program. The operation on the permanent memory will increase the scan period by 2 ms to 5 ms, and the data written by the save operation will overwrite the data stored in the permanent memory before.

Note

The number of saving operations on the EEPROM is limited (generally 1 million times). Perform the storage operation only when necessary. If the saving operation is continuously performed in the program, the service life of EEPROM ends quickly, and thus CPU failure is caused.

2.1.9 Digital filtering for input points

Input points X0 to X7 of IVC2L, IVC1, IVC1L and IVC3 series main modules are configured with the digital filtering function which can be used to filter the interfering signal of the ports. You can change the input filtering constant by configuring the **Input Filter** in the system block.

2.1.10 No battery mode

IVC1L, IVC2L, and IVC3 series main modules can operate without batteries. When you select the no battery mode, the system reports no system error caused by the lack of battery (loss of element data, loss of forcing tables, user program file error).

Refer to the description of the **No battery mode** configuration in the **Advanced Settings** of the system block.

Note

No battery is equipped on IVC1 series PLCs. Therefore, they do not support the no battery mode.

2.1.11 User program protection

IVC1, IVC1L, IVC2L, and IVC3 series PLCs are designed with multi-level password protection and other security strategies.

Table 2-7 User program protection

User program protection	Description
Disable formatting	After disabling formatting in the system block configuration and downloading the system block to a PLC, you cannot delete the user program, system block, and data block on the PLC by formatting. To enable formatting, you need to download a new system block in which formatting is not disabled.
Download password	Used to restrict the download function
Disable upload	In the download operation, select the upload disabled option in the download dialog box, and then you cannot upload the data later even you enter the upload password. To enable upload, you need to download the user data again and select the upload enabled option in the download dialog box.
Upload password	Used to restrict the upload function
Monitoring password	Used to restrict the monitoring function
Program password	Programmers can set passwords to encrypt the main program, subprograms and interruption subprograms. When opening the project in the programming software, you cannot view and edit the content of the encrypted programs. To view and edit it, you need to open the decryption dialog box and enter the correct password for decryption. Encryption method: Right-click the program to be encrypted, select **Encrypt/Decrypt** in the right-click menu, and then enter the password and confirm the password. Decryption method: Right-click the program to be decrypted, select **Encrypt/Decrypt** in the right-click menu, and then enter the correct password.

Note

If you enter a wrong password for five consecutive times, the IVC series micro-PLC disables password input for five minutes.

2.2 System configuration

2.2.1 System block

PLC configuration informationconfigured in a system block is compiled as a system block file, which is an important PLC user file. Before operating a PLC, you need to compile and download a system block file.

The system block configuration includes the following items:

- Saving range (element saving range)
- Set time (watchdog, constant scan time and power outage detection time)
- Input point (startup mode of the input point)
- Communication ports (communication ports and protocol settings)
- Interrupt priority (interrupt priority configuration)
- Inverter configuration
- Ethernet configuration
- Output table (output table settings)
- Input filter
- Advanced settings (data block, element value retained, no battery mode, anddisable formatting)
- Special module configuration
- Communication module
- CANopen configuration

After configuring a system block, you can select **PLC/Compile All** menu to compile a system block file and then download it.

- Saving range

Up on power outage, IVC1, IVC1L, IVC2L, and IVC3 series PLCs can save data of the elements that are set within the storage range to the power-outage storage area, and the data can be retained and used after the machine is powered on again.

Figure2-3 shows the address range of the elements for which data is to be saved on the first tab of the dialog box.

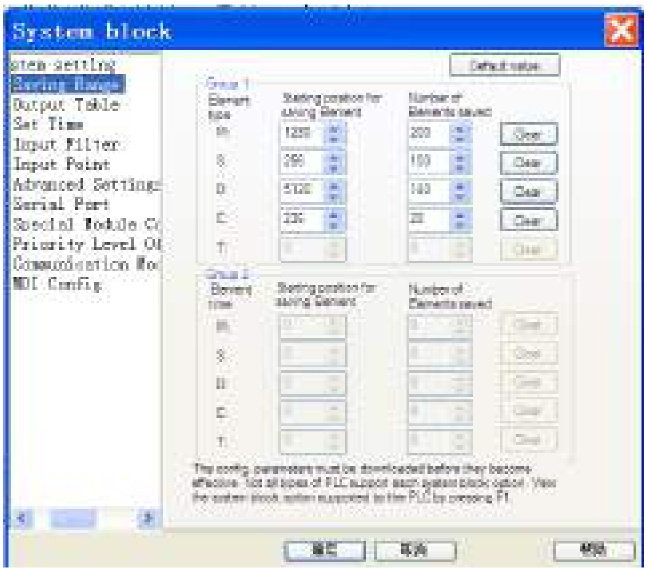


Figure 2-3 Configuring the address range of the elements for which data are to be saved

Note

The address ranges and groups of elements for which data can be saved vary according to PLC models.

By default, a certain storage range is automatically set for each of the D, M, S, T, and C elements.

You can change the address range of the elements for which data is to be saved on the tab as you need. Click the Clear button on the right to set the number of the elements (for which data is to be saved) to zero.

A maximum of two groups can be set for each of the IVC2L and IVC3 series, and the final range is the union of the two groups.

Only one group can be set for each of the IVC1 and IVC1L series.

Note

For IVC1 and IVC1L series PLCs, no T element can be set within the storage range.

System operation at power outage: The PLC saves the data of the elements set within the storage range to the power-outage backup file.

System operation at power-on: The PLC checks data in power-outage storage area. If the data saved in power-outage storage area is correct, the storage area on SRAM remains unchanged. If the data is incorrect, the PLC deletes data of all elements (including the elements that are within and out of the storage range) on SRAM.

- Output table

Click the **Output Table** tab and then you can set the state an output point enters after the PLC is stopped, as shown in Figure 2-4.

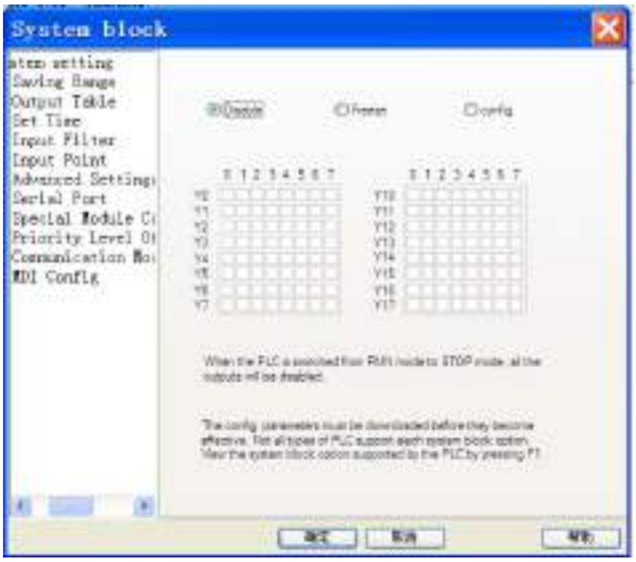


Figure 2-4 Setting the output table

The output table setting function is used to set the state an output point enters after the PLC is stopped. An output point can enter one of the following state after the CPU of the PLC is stopped:

(1) Disable:The PLC disables all the output points when switching from the running state to the stopped state.

(2) Freeze: The PLC freezes all the output points in their final state when being stopped.

(3) Configuration: The PLC sets all the output points to a known state when being stopped. The default state of all the output points is off (0).

- Set time

Figure 2-5 shows the set time tab.

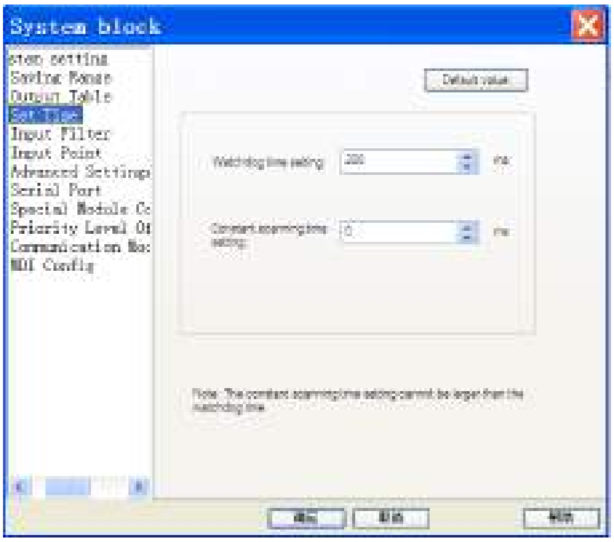


Figure 2-5 Set time

1. Watchdog time setting

You can set the user program running watchdog time. The watchdog time is the allowed maximum time for the user program to run. When the actual execution time of the user program exceeds the watchdog time, the PLC stops the execution of the user program, turns on the program alarm indicator (red), and provides output according to the system configuration. The watchdog time can be set from 0ms to 1000ms and the default value is 200ms.

2. Constant scan time setting

Constant scan time is a constant time when the system scans a register. The system's constant scan time setting register is read and the user program is scanned only once at the constant time. The constant time can be set from 0ms to 1000ms and the default is 0ms.

- Input filter

Clicking the **Input Filter** tab to set an input filtering constant for the PLC input points. The external interfering signals introduced by the input points can be filtered out by digital filtering. X0 to X7 (of the IVC1, IVC1L, and IVC2L series) are digital input points with the digital filtering function. Other digital input points adopt hardware filtering. IVC1 series input filtering can be used by group (two groups: X0 to X3 and X4 to X7) with filter constants of 0, 2, 4, 8, 16, 32, and 64. IVC3 series input filtering can be set separately for each input port and the filtering constants can be set consecutively from 0us to 60ms. IVC2L series input filtering cannot be used by group and the filtering constants can be set consecutively from 0 to 60ms. Figure 2-6 shows the input filtering settings for the IVC1 series.

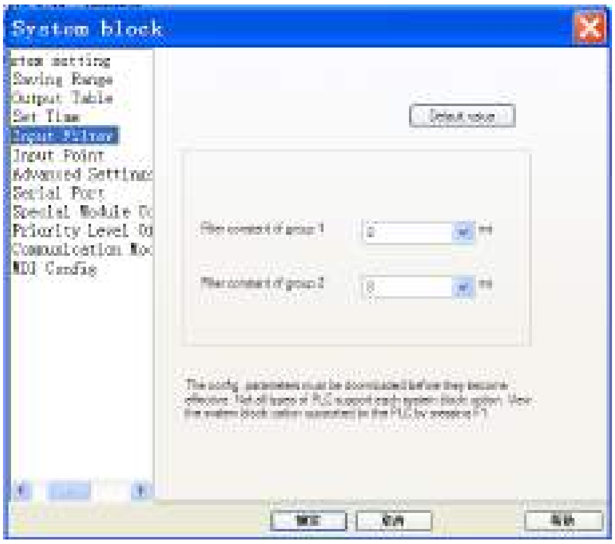


Figure 2-6 Setting the input filter

- Input point

Figure 2-7 shows the input point setting tab.

1. Set the input point for starting the PLC

When the **Disable input point** is not selected, you can set an input point (among X0 to X17) to force the PLC to enter RUN state. The PLC switches from the STOP state to RUN state when it is detected that the input point is ON.

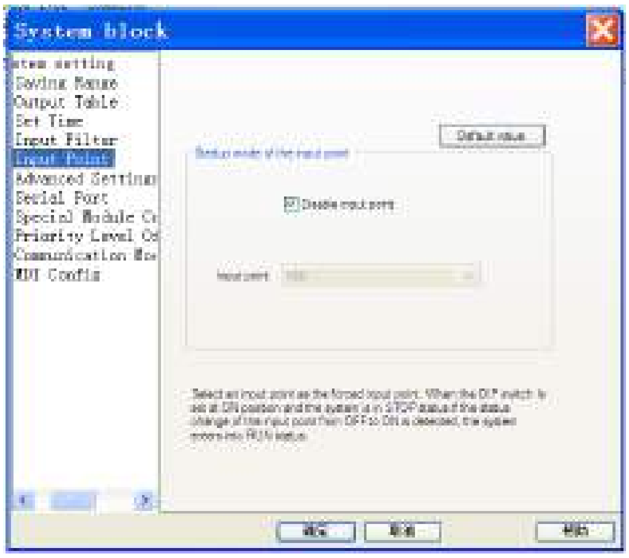


Figure 2-7 Input point setting

2. Disable input point

Select the **Disable input point** to disable the input point-based start function.

- Advanced settings

The advanced settings include Datablock enabled, Element value retained, no battery mode, etc.

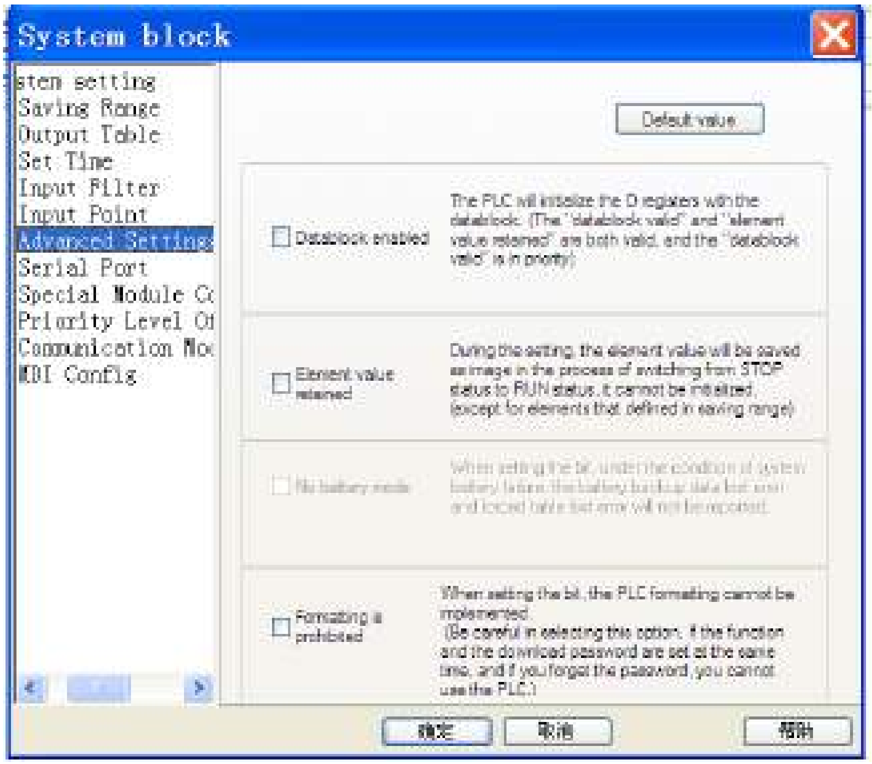


Figure 2-8 Advanced settings

1. Datablock enabled

Select Datablock enabledand the PLC uses data blocks to initialize D elements when switching from STOP to RUN.

2. Element value retained

Select Element value retained,and the PLC does not initialize the element value but save them as image when switching from STOP to RUN.

Note

When "**Datablock enabled**" and "**Element value retained**" are both valid, "**Datablock enabled**"prevails. For details, refer to section 2.1.6 "Element initialization".

---

3. No battery mode

Select **No battery mode**, and the system does not report errors caused by loss of battery backup data or loss of forcing tables upon backup battery failure.

- Communication port

You can set two or three communication ports on the communication port tab of the system block. Figure 2-9 shows the communication port settings. The item includes communication protocol selection and specific protocol parameter settings.

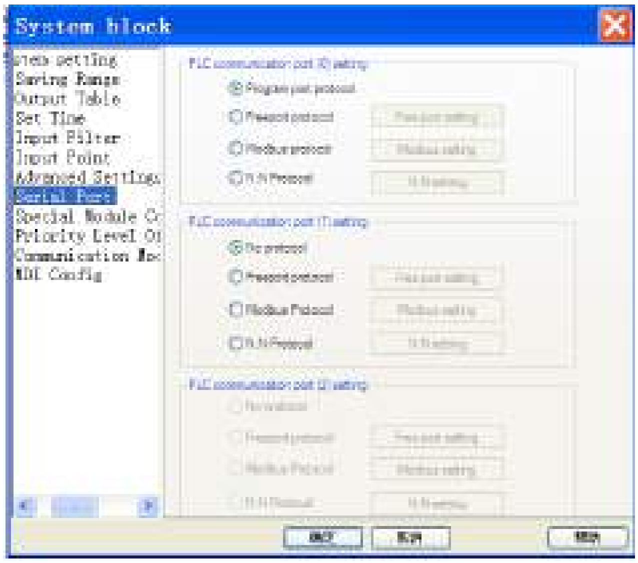


Figure 2-9 Communication port setting

By default, PORT0 adopts the programming port protocol while PORT1 and PORT2 adopt no protocol. You can set PORT0, PORT1, and PORT2 as you need.

1. Programming port protocol

By default, PORT0 adopts the programming port protocol, a dedicated protocol for the communication of the IVC series PLC programming software. With this protocol, you can set the communication baud rate between a PC and PORT0 through the serial port configuration tool of Auto Station. In TM state, PORT0 can be used only for programming port communication.

2. Free-port protocol

The free-port protocol is a communication modeusing user-defined data file formats. The free-port communication mode supports two data formats, namely ASCII and binary code. A PLC can adopt free-port communication only in the RUN state. When adopting the free-port mode, the PLC cannot communicate with programming devices.

The configurable parameters include baud rate, data bit, parity check, stop bit, start character detection, end character detection, intercharacter timeout, and interframe timeout.

3. Modbus communication protocol

Modbus communication devices include master and slave stations. A master station can communicate witha slave station (such as an inverter) and send control frames to the slave station according to the function codes of the Modbus communication protocol, and the slave station responds to the request of the master station.

PORT0 can be set as a slave station while PORT1 and PORT2 can be set as a master or slave station.

The configurable parameters include baud rate, data bit, parity check, stop bit, master-slave mode, station number, transmission mode, and timeout time and retry times of the master mode.

4. N:N communication protocol

N:N is an N:N communication protocol developed by INVT Auto-Control Technology Co., Ltd. for micro-PLC networks. PLCs in the N:N network can automatically exchange some of their D and M element values.

All of the PORT0, PORT1, and PORT2 can adopt the N:N communication protocol.

Note

For details about how to use the free-port, Modbus, and N:N protocols, refer to Chapter 10"Communication function guide".

- Special module configuration

You can set the module property on the special module configuration tab, as shown in Figure 2-10.

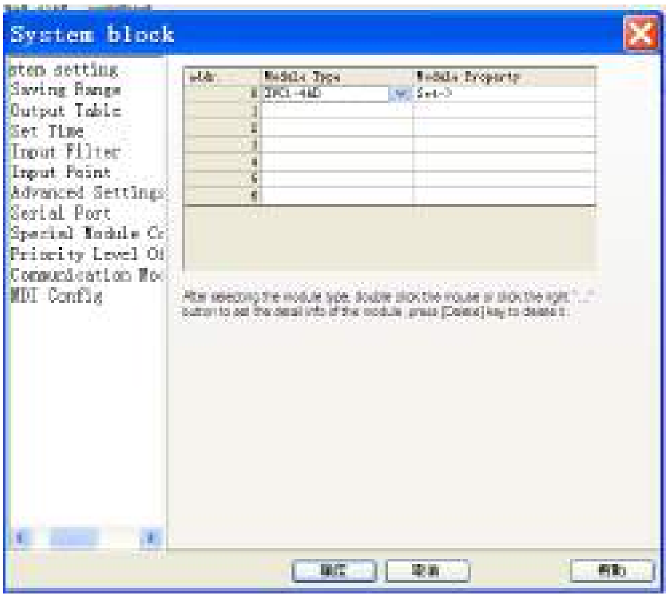


Figure 2-10 Special module configuration

1. Module type

You can select the module type for No.0 to No.3 special modules as required, as shown in Figure 2-10.

2. Module property

After selecting the **Module Type**, the corresponding **Module Property** is activated and the following dialog box may be displayed.

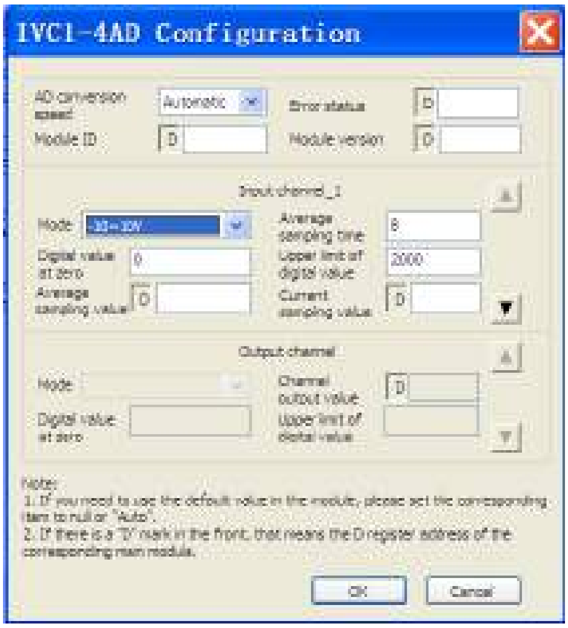


Figure 2-11 Module property setting

In the dialogue box shown in Figure 2-11, you can configure the channel for the special module, including **Mode** (signal features), **Digital value at zero**, **Upper limit of digital value**, **Average sampling** and **Current sampling value**. For details about the meanings and configuration methods of the parameters, refer to user manuals of the corresponding special modules.

- Interrupt priority

Figure 2-12 shows the interrupt priority setting tab.

The priority level of the PLC built-in interrupt can be selected. There are two priority levels, namely Low and High.

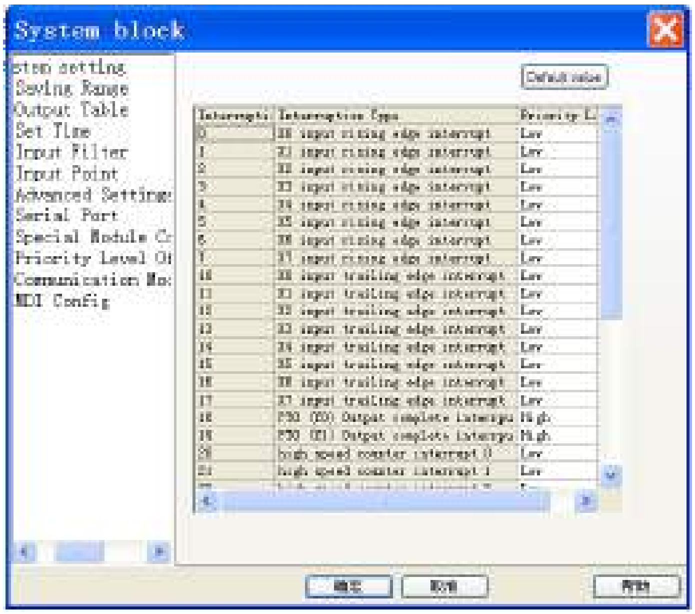


Figure 2-12 Interrupt priority setting

- Communication modules

You can set the connected communication modules, as shown in Figure 2-13.

Figure 2-13 Communication module setting

Click Settings, and the window shown in Figure 2-14 is displayed.

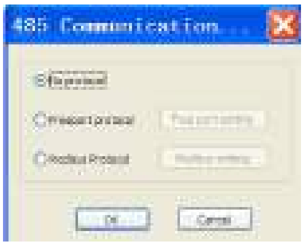


Figure 2-14 485 module configuration

- Inverter configuration

You can select inverter models and set station numbers as required, as shown in Figure 2-15.

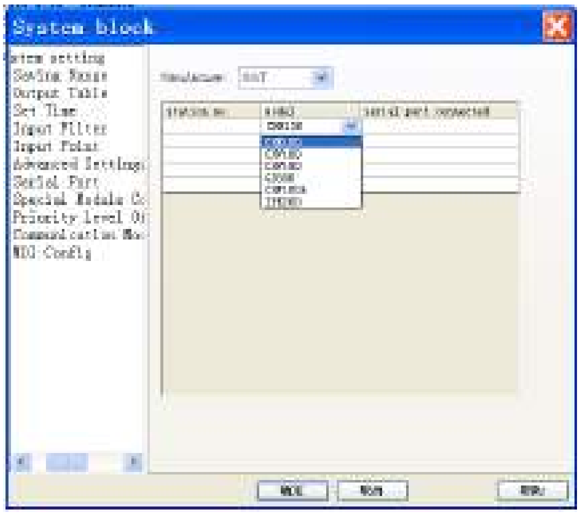


Figure 2-15 Inverter configuration

2.2.2 Data blocks

Data blocks are used to set the default values in D elements. If you download the compiled data block settings to a PLC, the PLC uses the data block to initialize the related D elements upon the startup.

The data block editor enables you to assign initial data to a D register (data memory). You can assign data to words or double words, but not to bytes. You can also add comments by inputting "//" to the front of a character string.

IVC3 supports data blocks of D and R elements.

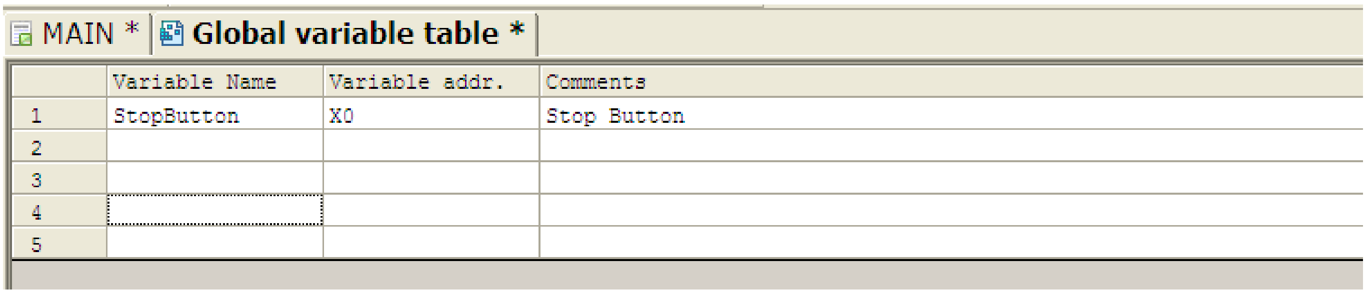
For details about operation instructions of data blocks, refer to the *Auto Station Programming Software User Manual*.

2.2.3 Global variable table

Global variable is a meaningful symbol name defined for a certain PLC address, and it can be visited within the whole project range, which is equivalent to using the corresponding element of the variable. Global variable is defined in the global variable table. Global variable includes three attributes, namely **Variable name**, **Variable address**, and **Comments**.

The definition rule of global variable is: comprised of A–Z, a–z, 0–9, underline, and Chinese characters, and the variable name cannot start with numbers nor can it be independent numbers. The name can be capitalized or lower-case, and the length cannot exceed 8 bytes. No element type letters+numbers can be adopted as the name of the program and variable. There should be neither blank space nor any name which is the same with keywords. The reserved keywords include basic data type name, instruction name, and operation symbol in instruction list language.

IVC3, IVC2L, and IVC1 series micro-PLCs are allowed to be downloaded no more than 1000, 500, and 140 global variables respectively (according to the maximum comment amount). Otherwise, it can only be saved locally but cannot be downloaded. Figure 2-16 shows the global variable table.


MAIN * | Global variable table *

	Variable Name	Variable addr.	Comments
1	StopButton	X0	Stop Button
2			
3			
4			
5			

Figure 2-16 Global variable table

2.2.4 Setting BFM for IVC2L and IVC3 series special modules

You do not need to set addresses of IVC2L and IVC3 series special modules. The PLC main modules are automatically detected and addressed upon power on.

Among the special modules, analog extension modules include analog input modules and analog output modules.

These two special modules are configured with default values in factory, including channel characteristics, zero point, and max. digital value.When necessary, you can modify these parameters to meet your actual needs for signal type selection, range scaling, error correction, etc.

- IVC2L and IVC3 analog input modules
  IVC2L and IVC3 analog-to-digital conversion (ADC) modules exchange information with the main modules through the BFM area.

  When a user program runs on the main module, a TO instruction writes the configuration data to relevant registers in the BFM area of IVF2L special modules, thereby modifying the default settings. The configuration data that can be changed includes: zero-point digital value, max. digital value, input channel signal characteristics, input channel enable flag, and so on.

  The main modules read the data in the BFM area of IVC2L ADC modules through the FROM instruction, including the ADC results and other information.

- IVC2L and IVC3 analog output modules
  IVC2L and IVC3 digital-to-analog conversion (DAC) modules exchange information with the main modules through the BFM area.
  When a user program runs on the main module, a TO instruction writes the configuration data to the relevant registers in the BFM area of IVC2L DAC special modules, thereby modifying the default settings. The configuration data that can be changed includes: zero-point digital value, max. digital value, input channel signal characteristics, input channel enable flag, and so on. The main module read the data in the BFM area of the IVC2L DAC modules through the FROM instruction, and then write the data that needs to be converted from digital value to analog value into the output buffer in the BFM area through the TO instruction.

For details about TO and FROM instructions, refer to the Chapter 6 "Application instructions" in this manual. For details about BFM registers and configuration methods of multiple special modules, refer to the manuals delivered with the corresponding modules.

2.3 Running mode and state control

You can start or stop PLCs in any of the following three ways.

1. Using the mode selection switch.

2. Setting the startup mode of input points and external terminals in the system blocks, and controlled by the specified terminals.

3. Using the programming software to run or stop the PLC if the mode selection switch is set to TM or ON.

2.3.1 Concepts of system run and stop states

The working states of main modulesinclude run and stop states.

- RUN
  When a main module is in Run state, a PLC executes the user program. That is to say, a scan cycle includes four tasks, namely executing the user program, communication, housekeeping, and refreshing I/O.
- STOP
  When a main module is in Stop state, a PLC does not execute the user program, but executes other three tasks in each scan cycle, namely communication, housekeeping, and refreshing I/O.

2.3.2 Run and stop state

- How to change from STOP to RUN

1. Resetting the PLC

If the mode selection switch is set to ON and the PLC is reset (including system power-on reset), the system enters the RUN state automatically.

Note

If the system configuration item of the **Input point control mode** in the main module is valid, the state of the corresponding input terminal is ON. Otherwise, the system cannot enter the run state after reset.

2. Setting the mode selection switch manually

The system is changed from STOP to RUN state when you set the mode selection switch from OFF or TM to ON.

3. Startup mode of input points

When system configuration item of **Input point startup mode** in system block tab is valid, the main module is changed from STOP to RUN once the system detects that the specified input points (X0 to X17) change from OFF to ON.

Note

When you select the input point control mode, a mode selection switch should be set to ON at the same time, otherwise the PLC cannot enter the run state.

- How to change from RUN to STOP

1. Resetting the PLC

If the mode selection switch is set to OFF or TM, reset the PLC (including power-on reset),and the system enters RUN state automatically.

Note

When the mode selection switch is set to ON, the system can also enter the stop state after reset if the system configuration item of the **Input point control mode** is valid and the designated input point is in OFF.

2. Setting the mode selection switch manually

The system is change from RUN to STOP state when you set the mode selection switch from ON or TM to OFF.

3. Using the instruction control mode

The system changes from RUN to STOP after executing the STOP instruction in the user program.

4. Auto-stop upon faults

The system stops the execution of the user program when detecting that a serious error occurs (such as user program errors oruser program execution timeout).

2.3.3 Setting the state of output pointsin the stop state

You can set the output state of the output point (Y) when the PLC is stopped. There are three modes for your choice:

Disable the output mode –When the PLC is stopped, all output points are OFF.

Freeze the output mode— When the PLC is stopped, all output points are frozen at the last state.

Configuration output mode— You can set the state of output points as needed when the PLC is stopped.

You can operate the above settings in the Output Table of the system block tab. For details, refer to the output table settings in section 2.2.1 "System block".

2.4 System commissioning

2.4.1 Program download and upload

- Download

The download function is used to download a system block, data block, and user program generated by the Auto Station software to a PLC through serial ports, and the PLC is required to stop running when downloading.

When downloading, if the program is changed after last compilation, you are informed whether you need to recompile the program, as shown in Figure 2-17.

Figure 2-17 Tips for recompiling the program

📖 Note

Selecting **N** button means that you have no need to recompile it and the software adopts the previous edited results, but the running program downloaded to the PLC is different from the one displayed on the software interface.

When downloading a program, the software pops up a password window asking for entering a download password if it's configured with a download password and the downloading password has not been entered after starting the software. The download starts after a password entered is verified successfully, otherwise, the system prompts you to re-enter your password and click the Cancel button to exit the download.

- Upload

The upload function is used to upload a system block, data block, and user program in a PLC to PC through serial ports, and save them for new projects. When the battery backup data is valid, a file is selected and then the corresponding user assistance information file is bundled and uploaded. Figure 2-18 shows the upload dialog box.

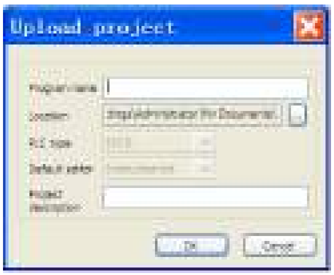


Figure 2-18 Upload the dialog box

When uploading a program, you can upload the program directly if it's not configured with a download password. The software pops up a password window asking for entering an upload password if it's configured with upload password and an upload password has not been entered after starting the software. Upload starts after a password entered is correct, otherwise, the software prompts you and returns to the upload dialog interface.

If you select the **Disable Upload** function when downloading a program, a PLC cannot upload the program later, unless you enter a correct password to remove the Disable Upload function.

2.4.2 Error reporting mechanism

A system can detect and report two types of errors: system error and user program execution error.

A system error is caused by abnormal system operation.

A user program execution error is caused by the abnormal execution of the user program.

All errors are numbered uniformly, and each error code represents an error. For details, refer to Appendix F"System error codes".

- The reporting mechanism for system error

When the system detects that a system error occurs, the system error code is written to a special data register SD3, and the special relay SM3 is set at the same time. You can obtain the system error information by assessing the error code stored in SD3.

If multiple system errors occur at the same time, the system only writes the code of the most serious error in SD3.

When a serious system error occurs, a user program stops, and a ERR indicator on the main module turns on for a long time.

- The reporting mechanism for user program execution error

When a user program execution error occurs, the system sets a special relay SM20 and writes the current error code into a special data register SD20.

When the next application instruction is executed correctly, the SM20 is reset while SD20 still keeps the previous error code.

The system records the user program execution errors sequentially in a stack. Special data registers SD20 to SD24 form an error stack that records error codes of the latest five user program execution errors.

When a user program execution error occurs and the current error code is different from one stored in SD20, the error code is added to the error stack. Figure 2-19 shows the stack push process of error codes when a user program execution error occurs.

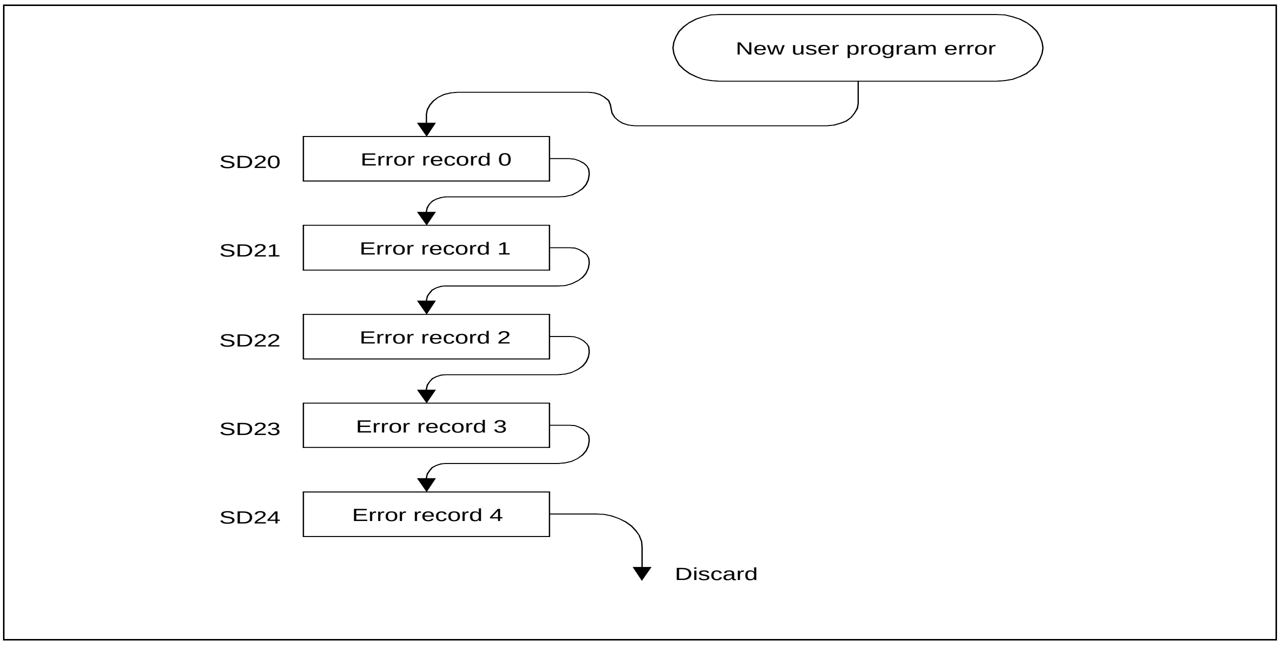


Figure 2-19 Push stack process of error codes

When a serious user program execution error occur, a user program stops and a ERR indicator on the main module turns on for a long time. In less serious cases, a ERR indicator on the main module does not turn on.

- Checking the error information online

Connecting the PLC with your PC through a serial port, and you can read various PLC state information through Auto Station, including codes and descriptions of the above-mentioned system errors and user program execution errors.

In the main interface of Auto Station, you can click **PLC**->**PLC Info** item to check the PLC information, as shown in Figure 2-20.

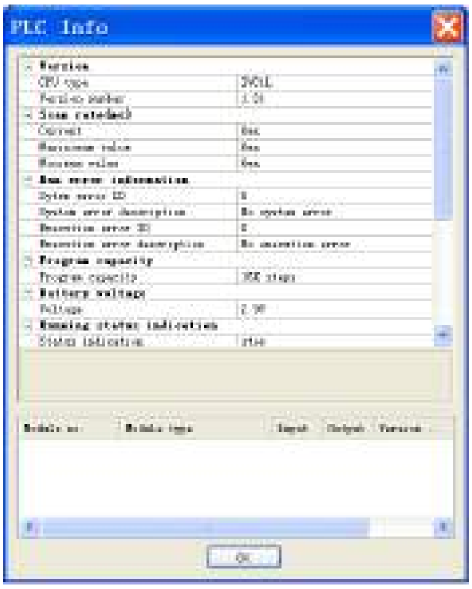


Figure 2-20 PLC information

The **System error ID** is the code of the system error stored in SD3 while the **Execution error ID** is the code of the user program execution error stored in SD20. The relevant error descriptions displayed by both are for your reference.

2.4.3 Modifying the user program online

You can use the online modification function when you need to modify the user program without stopping the PLC.

Warning

On the occasions that casualties or property loss may occur, the online program modification function should be used by professionals with sufficient protection measures.

- Operation method

  After making sure that PC-PLC communication has been set up successfully and the PLC is in the run state, you can click the **Debug** ->**Online modification** in the Auto Station main interface to enter the online modification state.

  In the online modification state, you can modify the main program, subprogram and interrupt subprogram as usual. After the modification, you click **PLC**->**Download** menu,and the modified program is compiled and downloaded to the PLC automatically. When the download completes, the PLC executes the new downloaded program.

- Limits

  1. In the online modification state, the global variable table and local variable table of any program cannot be modified, nor can add or delete any subprogram or interruption subprogram.

  2. When a program is in the online modification state, Auto Station automatically exits the online modification state if a PLC is stopped.

2.4.4 Clearing and formatting

The clearing operations include PLC element value clear, PLC program clear, and PLC data block clear.

Formatting is to clear all data and programs in PLCs.

- PLC element value clear

  The PLC element value clear function can clear all element values in the PLC. Element values should be cleared when the PLC is in the stop state.

  Clearing the element values in the PLC causes the PLC to operate improperly or lose the intermediate working data, so you need to use this function with caution. During the operation process, the software displays a confirmation window for you to choose whether to continue or cancel the current operation.

- PLC program clear

The PLC program clear function can clear the user programs in the PLC. User programs should be cleared when the PLC is in the stop state.

Clearing the user programs in the PLC leads to no-execution of the user programs by PLC, so you need to use this function with caution. During the operation process, the software displays a confirmation window for you to choose whether to continue or cancel the current operation.

- PLC data block clear

The PLC data block clear function can clear all the data block setups in the PLC. Data blocks should be cleared when the PLC is in the stop state.

Clearing the data blocks in the PLC causes the PLC to stop using the preset value of the data block for initializing D elements, so you need to use this function with caution. During the operation process, the software displays a confirmation window for you to choose whether to continue or cancel the current operation.

- PLC formatting

The PLC formatting function can format all the PLC data, including clearing the user program, restoring the default configuration, and clearing data blocks. Data blocks should be cleared when the PLC is in the stop state.

This operation clears all the downloaded and setup data, so you need to use this function with caution. During the operation process, the software displays a confirmation window for you to choose whether to continue or cancel the current operation.

2.4.5 Checking PLC information online

- PLC information

The PLC information function can be used to obtain and display various operation data and important information of PLCs. You can see the important information about the current operation of the PLC on the information display window, as shown in Figure 2-21.

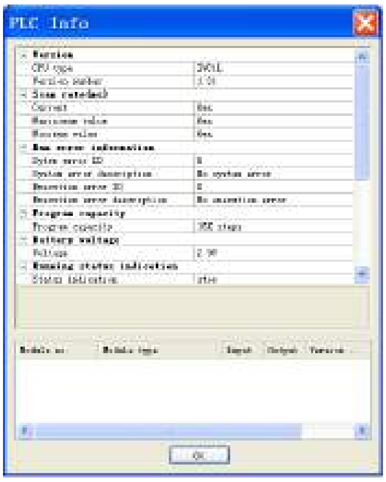


Figure 2-21 Current operation information of the PLC

- PLC time

The PLC time function can be used to display and set the current time of PLCs, as shown in Figure 2-22.

Figure 2-22 Setting the PLC time

The window displays the current date and time from the PLC. You can adjust the time setting and click the **Set Time** button to set a new time to the PLC.

2.4.6 Element value writing and forcing, and element monitoring table

- Write and forced values of the elements

  During the commissioning process, you can manually modify the value of certain soft elements to achieve some conditions, the element value write and force offer you this function. Difference between the write and force is that: the write element value is valid for once, and the value being written may change with the running of the program, but the forced element value is recorded in the PLC hardware until being cancelled by you.

  To execute the write or force function, you need to select the elements to be written or forced firstly, and right click the menu to select **Write** or **Force**. Then a corresponding dialog box pops up and lists all the soft element addresses being referred to by the selected elements. You can selectively write or force certain soft element values, after confirmation, these values are sent to the PLC hardware. When these values take effect in the hardware, you can see the change results during the subsequent commissioning process.

Figure 2-23 shows the dialog box for writing an element value.

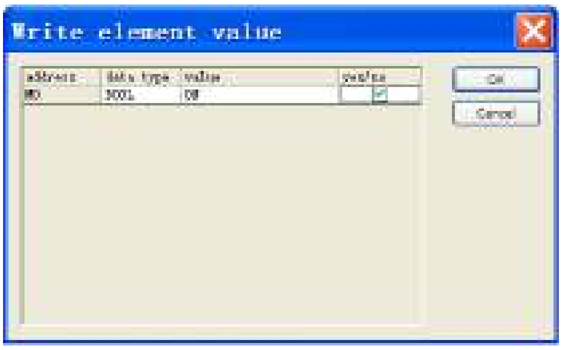


Figure 2-23 Dialog box for writing an element value

Figure 2-24 shows the dialog box for forcing an element value.

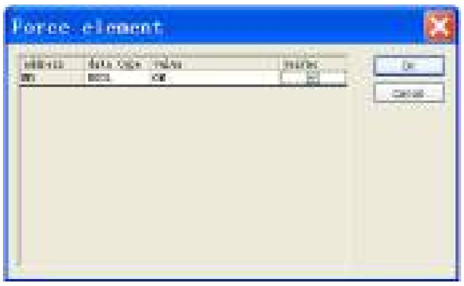


Figure 2-24 Dialog box for forcing an element value

The forced soft element carries a lock mark in LAD, as shown in Figure 2-25:

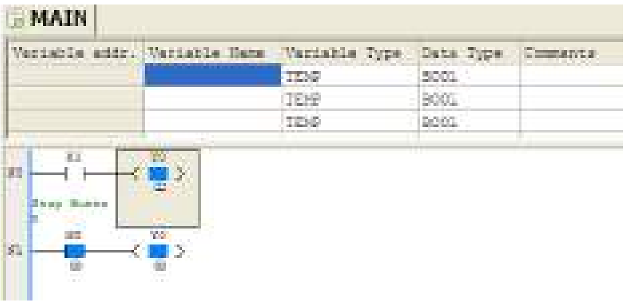


Figure 2-25 Locks mark of the forced soft element

- Element value unforcing

For element values that no long need to be forced, you can unforcing them. When you remove the force function, you need to first select target elements, right click and select **Unforce**, and then a dialog box pops up and displays all forced soft elements. You can selectively unforce certain forced soft elements, click OK button to confirm, and then these forced values are deleted from the PLC, so are the corresponding lock marks. Figure 2-26 shows the dialog box of unforce.

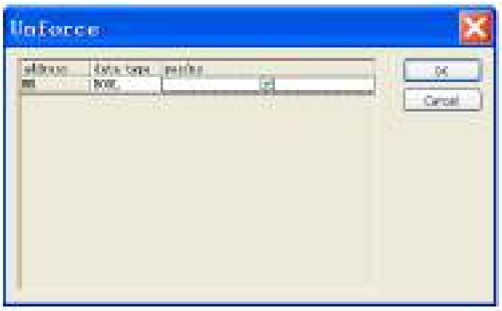


Figure 2-26 Unforce

- Element monitoring table

The element monitoring table provides element value monitoring function during commissioning. Program input and output elements, registers, and word elements can be added to the element monitoring table so that they can be tracked after the program is downloaded to PLCs.

The element monitoring table works in two modes: editing and monitoring modes. In the editing mode, all editing functions can be carried out but no monitoring function can be carried out. In the monitoring mode, both of the monitoring and editing functions can be carried out.

In the monitoring mode, displayed element values are refreshed automatically, and then modified or forced element values can be updated timely.

The element state monitoring table provides the functions includes editing, sequencing, searching, auto-refresh and display of the current values of designated elements, writing element values, forcing element/variable values, and unforcing values.

Figure 2-27 shows the element monitoring table.

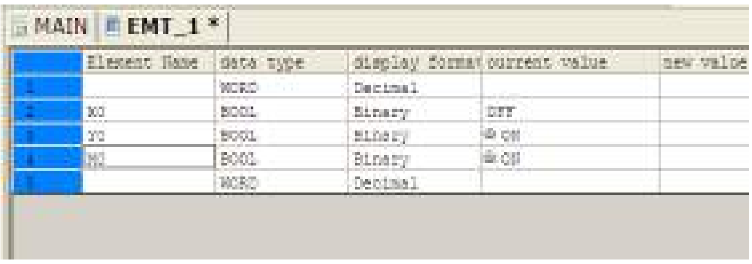


Figure 2-27 Element state monitoring table

2.4.7 Generating data blocks from RAM

The data values of 500 D registers (at most) read from PLC consecutively are displayed, and the results can be merged to the data block or overlay the original data block.

Open the window for generating data blocks from RAM, as shown in Figure 2-28.

Figure 2-27Window for generating data blocks from RAM

Entering the range of data blocks to be read, click the **Read from RAM** button, and data is read into the list after the instruction is correctly executed.

You can select hex, decimal, octal, or binary in the field of **display typ**e to display the data.

After reading the data successfully, the **Merge to datablock** and **Overwrite datablock** button become available. Click the **Merge to datablock** button to add the generated results to the rear part of the current data blocks and click the **Overwrite datablock** button to replace the existed contents in the data block with the generated results. After exiting the register value reading window, the software prompts that the data block has been changed and the data block window is open automatically.

Chapter 3 Soft element and data

This chapter detailedly describes the definition, classification and functions of the soft elements of IVC series PLCs.

3.1 Type and function of soft elements

3.1.1 Summary of soft elements

PLCs are configured with a variety of virtual elements in the system design to replace the real generic relays, time relays, and other devices in the relay control circuits.These virtual elements are collectively referred to as soft elements. PLCs adopt soft elements for program operation and system function configuration to implement all operation and control functions. Due to their virtual nature, the soft elements can be used repeatedly in the program without quantity limit the oretically (actually related to the program capacity). Besides, the soft elements have no mechanical or electric problems of the real devices. Such features make PLCs much more reliable than relay control circuits. In addition, it is easier to program and modify the logic.

The types and functions of IVC series PLC soft elements are shown in the following figure.

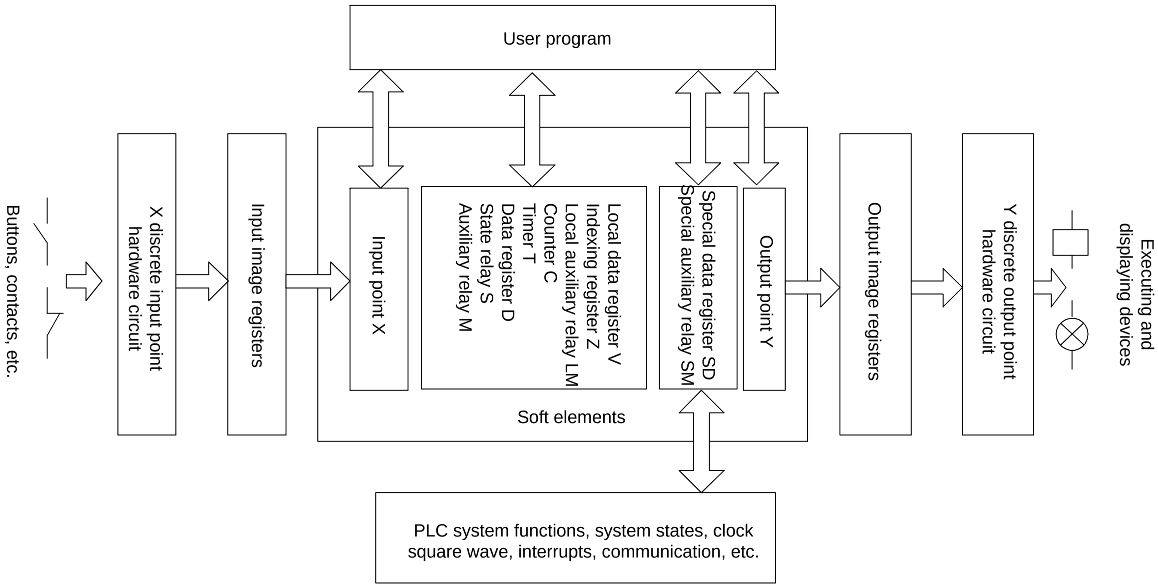


Figure 3-1 Types and functions of soft elements

In this manual, soft elements are named according to their types. For example:

- Input point X, or "X element" for short
- Output point Y, or "Y element" for short
- Auxiliary relay M, or "M element" for short
- Data register D, or "D element" for short
- State relay S, or "S element" for short

3.1.2 Soft elementlist

The soft elements of IVC series micro-PLCs are compiled and classified according to their functions. Different elements execute different functions and are easily accessible.

Figure 3-1 shows the list of IVC series micro-PLC soft elements.

Figure 3-1 IVC series PLC soft element list

		IVC1 series	IVC1L series	IVC2L series	IVC3 series	Numbered in
Soft element resources[note 4]	I/O points	128 inputs/128 outputs (inputs: X0–X177; outputs: Y0–Y177)[note 1]	128 inputs/128 outputs (inputs: X0–X177; outputs: Y0–Y177)[note 1]	256 inputs/256 outputs (inputs: X0–X377; outputs: Y0–Y377)[note 1]	512 inputs/512 outputs (inputs: X0–X777; outputs: Y0–Y777)[note 1]	Octal

	Auxiliary relay	2048 (M0–M2047)	2048 (M0–M2047	2000 (M0–M1999)	10240 (M0–M10239)	Decimal
	Local auxiliary relay[note 5]	64 (LM0–LM63)	64 (LM0–LM63)	64 (LM0–LM63)	64 (LM0–LM63)	Decimal
	Special auxiliary relay	256 (SM0–SM255)	512 (SM0–SM511)	512 (SM0–SM511)	1024 (SM0–SM1023)	Decimal
	State relay	1024 (S0–S1023)	1024 (S0–S1023)	992 (S0–S991)	4096 (S0–S4095)	Decimal
	Timer	256 (T0–T255)[note 2]	256 (T0–T255)[note 2]	256 (T0–T255)[note 2]	512 (T0–T511)[note 2]	Decimal
	Counter	256 (C0–C255)[note 3]	256 (C0–C255)[note 3]	256 (C0–C255)[note 3]	307 (C0–C256)[note 3]	Decimal
	Data register	8000 (D0–D7999)	8000 (D0–D7999)	8000 (D0–D7999)	8000 (D0–D7999)	Decimal
	Data register R				32768 (R0–R32767)	Decimal
	Local data register[note 5]	64 (V0–V63)	64 (V0–V63)	64 (V0–V63)	64 (V0–V63)	Decimal
	Indexing register	16 (Z0–Z15)	16 (Z0–Z15)	16 (Z0–Z15)	16 (Z0–Z15)	Decimal
	Special data register	256 (SD0–SD255)	512 (SD0–SD511)	512 (SD0–SD511)	1024 (SD0–SD1024)	Decimal

Notes:

Note 1: The addresses of X and Y elements are numbered in octal, for example, address X10 represents the 8th input point. The max. value of I/O points here is the system capacity. The actual extendable hardware points should depend on the PLC system configuration (including the available extension module types and points, power capacity limit, etc.).

Note 2: T element address is classified into three categories based on timing precision:

- 100 ms precision: T0–T209
- 10 ms precision: T210–T251
- 1 ms precision: T252–T255

(IVC3)

- 100 ms precision: T0–T209
- 10 ms precision: T210–T479
- 1 ms precision: T480–T511

Note 3: C element address is classified into three categories based on the width and function of count values:

- 16-bit increment counter: C0–C199
- 32-bit increment and decrement counter: C200–C235
- 32-bit high-speed counter: C236–C255

(IVC3)

- 16-bit increment counter: C0–C199
- 32-bit increment and decrement counter: C200–C235
- 32-bit high-speed counter: C236–C255, C301-C307, C256-C300 reserved

Note 4: Some internal soft element resources of PLCs have been reserved for internal use only. You should not use these elements on the user program, if possible. For details, refer to Appendix C "Reserved elements".

Note 5: These two types of soft elements are local variables which cannot be defined in the global variable table. When the user program calls subprograms or returns to the main program, they are rest to zero obtain the parameter values or states according to the interface parameter transfer function.

3.1.3 Input and output points

- Element mnemonics
  - X elements (discrete input points)
  - Y elements(discrete output points)
- Function

  The X and Y elements represent respectively the input state of the hardware X terminal and output state of hardware Y terminal.

  The state of X elements is obtained through the inputimage register, while the state of Y elements is output through the output circuit driven by the output image register. The two operations are carried out in the I/O refresh phase of PLC scan cycle mode, as shown in Figure3-2. For details, refer tosection 2.1.2 "PLC operating mechanism (Scan cycle model)". It is obvious that thereis a brief delay in PLC's response to the I/O. The delayis related to the input filter, communication, housekeeping, and scan cycle.

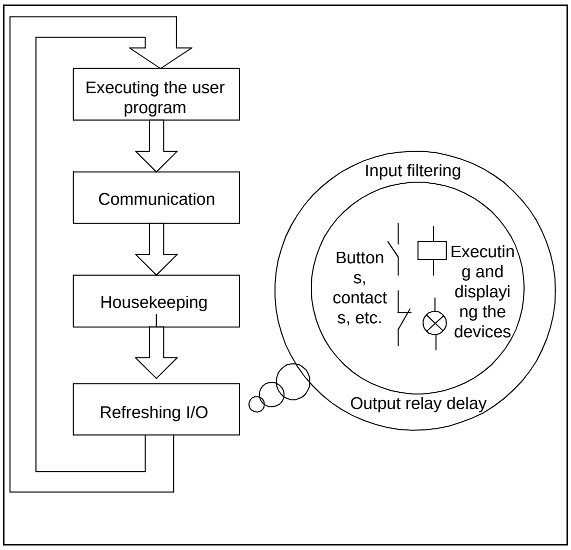


Figure 3-2 Schematic diagram of I/O refresh

- Classification

  Output channel corresponding to X elements: X0 to X17 are configured withthe digital filtering function. You can set the filtering time through the system block. Other X input points use the hardware filtering. X0 to X7 can be used as counting input points for soft elements of high-speed counters. Besides, X0 to X7 can also be used as input terminals for external interrupt, pulse capture, and the SPD instruction.

  Y elements are used as high-speed output terminals or general output terminals.
- Elements addressing mode

  Octal,started with 0. X and Y elements of the main module and I/O extension modules are addressed contiguously. X elements are addressed contiguously in X0 – X7, X10 – X17, X20 – X27, and so on. Y elements are addressed contiguously in Y0–Y7, Y10–Y17, Y20–Y27, and so on.
- Data type

  Both the X and Y elements are Boolean (element values are ON or OFF).
- Available forms

  You can adopt NO and NC contacts of X elements during programming (depending on two instructions). NO and NC contacts have opposite state values.They are sometimes referred to as "a contact" and "bcontact".

  You can also adopt NO and NC contacts of Y elements during programming.
- Value assignment

  1. X elements accept only hardware input state and forced operation state values which cannot be changed through output or setting instructions, nor be set during the system commissioning.

  2. You can assign state values to Y elements with the OUTinstruction, set the state values of Y elements, oreven force or write state values of Y elements during the system commissioning.

  3. Through the system block, you can set the output states of Y elements in STOP state.

3.1.4 Auxiliary Relays

- Element mnemonic

  M elements
- Function

  The system provides a type of discrete state elements to you, which is similar to an intermediate relay in a real electrical control circuit. You can use them to store various intermediate states in the user program.
- Elements addressing mode

  Octal, started with 0.
- Data type

M elements are Boolean (the element values are ON or OFF).

- Available forms
  NO and NC contacts
- Value assignment
  1. Through the instruction operations. 2. Force or write the state values during the system commissioning.
- Saving dataat power outage

State	M elements for which data is saved at power outage	M elements for which data is not saved at power outage
Power outage	Data saved	Date deleted
RUN → STOP	Data saved	Date deleted
STOP → RUN	Unchanged	Cleared
Note: The storage range at power outage is set through the system block. For details, refer to section 2.2.1 "System block".		

Note

When using the N:N protocol, some M elementsare called by the system. You need to pay attention to it when programming and modifying the program.

3.1.5 State relay

- Element mnemonic
  S elements
- Alias
  Step flag
- Function
  As the step flag, S elements are used in the SFC. For details, refer to Chapter 7 "SFC user guide".
- Classification
  S0–S19: initial step flag
  Others: general step flag
- Elements addressing mode
  Decimal, starting with 0
- Data type
  Boolean elements (element values are ON or OFF).
- Available forms
  1. Representing the step state (when the STL instruction is programmed through SFC)
  2. NO and NC contacts (when the STL instruction is not programmed for SFC). Similar to characteristics of M elements, NO and NC contacts of S elements can be used for programming.
- Value assignment
  1. Through the instruction operations. 2. Force or write the state values during the system commissioning.
- Uninterrupted output after power outage

State	S elements for which data is saved at power outage	S elements for which data is not saved at power outage
Power outage	Unchanged	Cleared
RUN → STOP	Unchanged	Unchanged
STOP → RUN	Unchanged	Cleared
Note: The storage range at power outage is set through the system block. For details, refer to 2.2.1 "System block".		

3.1.6 Timer

- Element mnemonic
  T elements
- Function
  T element is a composite soft element that contains a word element (2 bytes) and a bit element. The T word element records a 16-bit timing value that can be used as a value in the program. The T bit element reflects the state of the timer coil, which can be used for logic control.

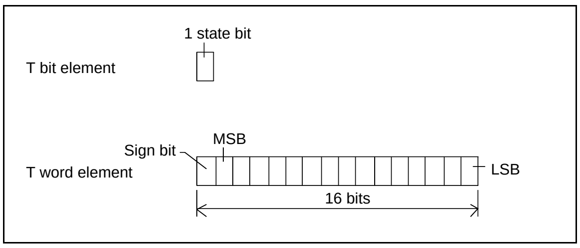


Figure 3-3 T elements

- Classification
  According to the timing precision, T elements are classified into three types.The following table describes the T elements and corresponding timing precisions in different address segments.You need to pay attention to it when using.

T element	Timing precision
T0–T209	100 ms precision
T210–T251	10 ms precision
T252–T255	1 ms precision

  For T elements with timing precision of 1 ms, they areactivated by interrupt trigger and unrelated to the PLC scancycle. Therefore, their action time is the most accurate. For T elements with timing precision of 10 ms and 100 ms, refresh and action time of their timing values are related to the PLC scan cycle.
- Elements addressing mode
  Decimal, starting with 0
- Data type

Boolean (element values are ON or OFF), and word.

- Available forms
  The timing and action modes of T elements are determined by timing instructions. There are four types of timing instructions, including TON, TOF, TONR, and TMON instructions. For details, refer to Chapter 5 "Basic instructions".
- Value assignment
  1. Through the instruction operations. 2. Force or write the state values during the system commissioning.
- Uninterrupted output after power outage

State	T elements for which data is saved at power outage (for IVC2L series only)	T elements for which data is not saved at power outage
Power outage	Unchanged	Cleared
RUN → STOP	Unchanged	Unchanged
STOP → RUN	Unchanged	Cleared
Note: The storage range at power outage is set through the system block. For details, refer to section 2.2.1 "System block".		

📖 Note

The maximum timing value of T element is 32767. The preset value is -32768 to -32767. Because T element acts only when the count value reaches or exceeds the preset value. Therefore, it is actually meaning less to set the preset value as a negative number.

3.1.7 Counter

- Element mnemonic
  C elements
- Function
  A C element is a composite soft element that contains a bit element, and a word or adouble word element (2 or 4 bytes). The C word element records a 16-bit or 32-bit timing value that can be used as a value in the program while C bit element reflects the timer coil state for logic control.

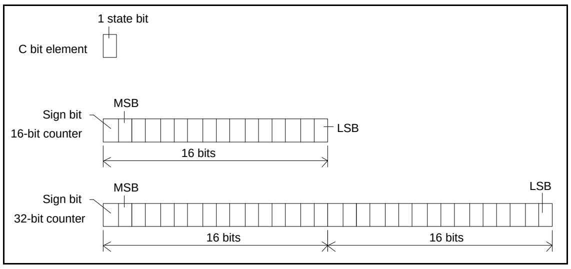


Figure 3-4 C elements

- Classification
  Two types: 16-bit counter and 32-bit counter
- Elements addressing mode
  Decimal, starting with 0
- Data type
  Boolean(element values are ON or OFF), single word or double word.
- Available forms
  The counting instructions that call C elements are classified into 4 types, namely 16-bit increment counting instruction, 16-bit cycle counting instruction, 32-bit increment and decrement counting instruction, and high-speed I/O instruction. For details, refer to Chapter 5 "Basic instructions" and Chapter 6 "Application instructions".
  The classification of C elements is shown in the table below:

C elements	Counting function	Applicable instruction type
C0–C199	16-bit increment counter	16-bit increment counting instruction 16-bit cyclic counting instruction
C200–C235	32-bit increment and decrement counter	32-bit increment and decrement counting instruction
C236–C255	32-bit high-speed counter	High-speed I/O instruction

- Value assignment
  1. Through the instruction operations. 2. Force or write the state values during the system commissioning.
- Uninterrupted output after power outage

State	C elements for which data is saved at power outage	C elements for which data is not saved at power outage
Power outage	Unchanged	Cleared
RUN → STOP	Unchanged	Unchanged
STOP → RUN	Unchanged	Cleared
Note: The storage range at power outage is set through the system block. For details, refer to section 2.2.1 "System block".		

3.1.8 Data register

- Element mnemonic
  D and R elements
- Function
  As data elements, D or R elements are used as operandsin manycalculation and control instructions.
- Elements addressing mode
  Decimal, starting with 0
- Data type
  Each D or R element is a 16-bit register that can store 16-bit data, such as 16-bit integer.

Two D or R elements can form a double-word element that can store32-bit data, such as long integer data or floating-point data.

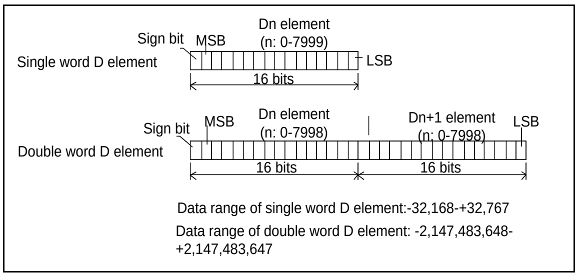


Figure 3-5 D or R elements

Note

In a double-word D or R element, the higher 16-bit is in the first D or R element, and the lower 16-bit is in the second D or R element.

- Available forms
  D or R elements are used as operands in many calculation and control instructions.
- Value assignment
  1. Data block initialization. 2. Through the instruction operations. 3. Force or write the state values during the system commissioning.
- Uninterrupted output after power outage

State	D elements for which data is saved at power outage	D elements for which data is not saved at power outage
Power outage	Unchanged	Cleared
RUN → STOP	Unchanged	Unchanged
STOP → RUN	Unchanged	Cleared
Note: The storage range at power outage is set through the system block. For details, refer to section 2.2.1 "System block". *R elements cannot be saved at power outage*		

Note

Some D elements are called by the system when inverter instruction or N:N protocol is adopted. You need to pay attention to it when programming and modifying the program.

3.1.9 Special auxiliary relay

- Element mnemonic
  SMelements
- Function
  SM elements are closely related to the PLCsystem function, reflectingthe system function and state of the PLCs. For details, refer to Appendix A "Specialauxiliary relay" in this manual.
- Classification
  The frequently used SM elements include:
  - SM0: PLC operation monitor bit. It is ON whenthe PLC is in RUN state.
  - SM1: initial operation pulse bit. It is ON in the firstscan cycle of PLC operation.
  - SM3: system error. It is ON if any system error is detected after the PLC is powered on or when the PLC changes from STOP to RUN.
  - SM10–SM12: respectively the clock square-wave cycled at 10 ms, 100 ms, and 1 s (flipping-overtwice in a cycle).

  In addition, you can use, control, or change the PLC system function by adjusting states of certain SM elements.

  Such elements include:
  - SM40 – SM68: Interrupt control flag bit. Setting theseSM elements enables the corresponding interrupt functions.
  - SM80 or 81: Y0 or Y1 high-speed pulse output stopinstruction.
  - SM110–SM114: The monitoring bit of free port 0.
  - SM135 and 136: Modbus communication flag bit.
  - SM172–SM178: Enable flag for the integrated analog channel (valid only for IVC1-1614MAR1)
- Elements addressing mode
  Decimal, started with 0.
- Data type
  Boolean elements (element values are ON or OFF).
- Available forms
  NO and NC contacts
- Value assignment
  1. Through the instruction operations. 2. Force or write the state values during the system commissioning.

Note

You cannot assign values to read-only SD elements.

3.1.10 Special data register

- Element mnemonic
  SD elements
- Function
  SD elements are closely related to the PLC system function, reflecting the system function parameters, state code values, and instruction operation data of PLCs. For details, refer to Appendix B "Special data register" in this manual.
- Elements addressing mode
  Decimal, started with 0.

- Data type
  Word, and doubleword (integer)elements.
- Available forms
  Storage and calculation of integers
- Value assignment
  1. Through the instruction operations. 2. Force or write state values during system commissioning.

Note

You cannot assign values to read-only SD elements.

3.1.11 Indexing register

- Element mnemonic
  Z elements
- Function
  16-bit register elements can be used to store symbolic integer data.For details about indexing, refer to section 3.1.15 "Indexing mode (Z addressing Mode)".
- Elements addressing mode
  Decimal, started with 0.
- Data type
  Word elements
- Available forms
  Z elements are used for indexing. You need to write the address offset in Z elements before using them.
- Value assignment
  1. Through the instruction operations. 2. Force or write state values during system commissioning.

3.1.12 Local auxiliary relay

- Element mnemonic
  LM elements
- Function
  LM elements are local variants that can be applied in the main program and subprograms. They are variable elements that are locally valid in each individual program body (main program, subprogram, and interrupt program). Therefore, it is not possible to directly share the state of any LM element between different program bodies. When the system leaves a program body in the execution of the user program, the LM element is redefined. When returning to the main program or calling a subroutine, the value of the redefined LM element is deleted or the corresponding state needs to be obtained according to the interface parameter transfer function.
  LM elements can be used to define the interface parameters of subprograms to realize the interface parameter transfer function. For details, refer to section 4.4 "Subprogram".
- Elements addressing mode
  Decimal, started with 0.
- Data type
  Boolean(element values are ON or OFF).
- Available forms
  NO and NC contacts
- Value assignment
  1. Through the instruction operations.

3.1.13 Local data register

- Element mnemonic
  V elements
- Function
  V elements are local variants that can be applied in the main program and subprograms. They are variable elements that are locally valid in each individual program body (main program and subprogram). Therefore, it is not possible to directly share the data of any V element between different program bodies. When the system leaves a program body in the execution of the user program, the V element is redefined. When returning to the main program or calling a subroutine, the value of the redefined V element is deleted or the corresponding data needs to be obtained according to the interface parameter transfer function.
  V elements can be used to define the interface parameters of subprograms to realize interface parameter transfer function. For details, refer to section 4.4 "Subprogram".
- Elements addressing mode
  Decimal, started with 0.
- Data type
  Boolean elements (element values are ON or OFF).
- Available forms
  Word element, which can be used to store numeric information
- Value assignment
  1. Through the instruction operations.

3.1.14 Bit-string combined addressing mode (Kn addressing mode)

- Concept

The bit-string combined addressing mode,or Kn addressing mode, combines bit element strings into words or long words.

- Method

The format is: "K (n) (U)", where the "n" is an integer among one to eight, indicating that the length of the bit string is n×4 bits. U indicates the start bit element address of the element string.

Application instance:

1. K1X0 indicates a word made up of a bit string of 4 bit elements (X0, X1, X2 and X3) starting from X0.

2. K3Y0 indicates a word made up of a bit string of 12 bit elements (Y0, Y01, Y02 and Y03), (Y04, Y05, Y06 and Y07), and (Y10, Y11, Y12 and Y13) starting from Y0.

3. K4M0 indicates a word made up of a bit string of 16 bit elements: M0, M1, M2, M3… and M15.

4. K8M0 indicates a double word made up of a bit string of 32 bit elements: M0, M1, M2, M3… and M31.

- Data storage format in the Kn addressing mode

The following example describes how data is stored in the Kn addressing mode:

MOV 2#10001001 K2M0 (which is the equivalent of MOV 16#89 K2M0, or MOV 137 K2M0). After the instruction is executed, K2M0 is stored. The following table describes the specific storage format of K2M0.

Data	MSB	Intermediate bit						LSB
K2M0	M7	M6	M5	M4	M3	M2	M1	M0
16#89	1	0	0	0	1	0	0	1

- Note of the bit-string combined addressing mode

If the destination operand of the instruction uses the Kn addressing mode, while the width of the data to be stored in the destination operand is greater than the width defined in the Kn addressing mode, the system keeps the LSB and discards the MSB of the data.

For example:

Executing the instruction "DBITS 16# FFFFFFF0 K1M0".

After the instruction is executed, the operand 2 (K1M0) needs to store the calculation result 16#1c (28). However, the K1M0 is only 4-bit wide, which is not enough for 16#1c. By discarding the higher bits, the actual calculation result of operand 2 is K1M0=16#c (12).

3.1.15 Indexing mode (Z addressing mode)

- Concept of indexing

IVC2L and IVC3 series PLCs provides the indexing mode (Z addressing mode). You can use Z elements (indexing registers) to get indirect access to the target elements.

- Z addressing method:

Target address=Basic element address+Address offset stored in Z element.

For example:

In the indexing mode, for D0Z0 (in which Z0=3), the target address is D3. D0 is the basic address, and the offset address stored in element Z0 is 3.

Therefore, when Z0=3, the instruction "MOV 45 D0Z0" is equal to "MOV 45 D3" in effect, and in both cases D3 is assigned the value 45 after the instruction is executed.

- Indexing example

1. Bit element indexing example LD M01

MOV 6 Z1

SFTR X0Z1 M0 8 2

The preceding instructions are in effect equal to:

LD M01

SFTR X6 M0 8 2

The addressing process is as follows:

Z1=6

X0Z1 = X (0＋Z1) = X6

2. Word element indexing example

LD M01

MOV 30 Z20

MOV D100Z20 D0

The preceding instructions are in effect equal to:

LD M01

MOV D130 D0

The addressing process is as follows:

Z20=30

D100 Z20 = D (100＋Z20)= D130

- Notes of the indexing mode

1. In the indexing mode (Z addressing mode), Z elements store the address offset. They support signed integers, which means negative offset is supported.

For example:

MOV -30 Z20

MOV D100Z20 D0

The preceding instructions are in effect equal to:

MOV D70 D0

2. The SM and SD elements do not support the indexing mode.

3. You need to pay attentrion to the address range when using the Z addressing mode. For example, D7999Z0 (Z0=9) is outside the address range of the D elements (The max. address of D elements is 7999.).

3.1.16 Bit-string combined indexing mode

The bit-string combined addressing mode can be used in conjunction with the indexing mode. For example: K1X0Z10.

In this mode, you need to determine the start bit element address through the Z addressing, and determine the length of the bit string through the Kn addressing.

For example:

LD M1

MOV 3 Z10

MOV K1X0Z10 D0

The preceding instructions are in effect equal to:

LD M1

MOV K1X3 D0

The addressing process is as follows:

Z10 = 3

K1X0Z10=K1X (0+Z10) = K1X3

3.1.17 Storing and addressing 32-bit data in D, R, and V elements

- Storing 32-bit data in D, R, and V elements

  The DINT, DWORD, and REAL type data are 32-bit, anda D, R, or V element is only 16-bit. Two consecutive D, R, or V elements are needed to store the 32-bit data.

  IVC2L series PLCs store the 32-bit data in the Big Endian mode, that is to say,the elements with small addressesare used to store the higher bits of the 32-bit data, while the elements with big addresses are used to store the lower bits of the 32-bit data.

  For Application instance, unsigned long integer "16# FEA8_67DA" is stored in the element (D0, D1). The actual storage format is as follows:

D0	0xFEA8
D1	0x 67DA

- Addressing 32-bit data in D, R, and V elements

  You can use a D or V element to locate a 16-bit data, such as an INT or WORD type data, or a 32-bit data, such as a DINT or DWORD data.

  If a D, R, or V element address is used in an instruction operand, the operand data type determines whether the data is 16-bit or 32-bit.

  For example:

  In the instruction "MOV 16#34 D0", the address D0 indicates a single D0 element, because operand 2 of the MOV instruction is of the WORD data type.

  In the instruction "DMOV 16# FEA867DA D0", the address D0 indicates two consecutive words: D0 and D1,because operand 2 of the DMOV instruction is of the DWORD data type.

3.2 Data

3.2.1 Data type

All instruction operands are of a certain data type. There are altogether six data types, as listed in the following table.

Table 3-2 Operand data types

Data type	Type description	Data width	Range
BOOL	Bit	1	ON, OFF (1, 0)
INT	Signed integer	16	－32768–+32767
DINT	Signed long integer	32	－2147483648–+2147483647
WORD	Word	16	0–65535 (16#0–16#FFFF)
DWORD	Double word	32	0–4294967295 (16#0–16#FFFFFFFF)
REAL	Floating-point number	32	±1.175494E -38–±3.402823E ＋38

3.2.2 Correlation between elements and data types

The element types used by instruction operands needs to match their data types. The correlations between the applicable elements and data types are listed in the following table.

Table 3-3 Mapping between elements and data types

Data type	Soft elements													
BOOL										C	T			
	X	Y	M	S	LM	SM								
INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R

Data type	Soft elements													
DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R
WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R
DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V		R
REAL	Constant							D				V		R

If an instruction uses an operand with an unsuitable data type, the instruction is deemed illegal. For example,instruction "MOV 10 X0" is illegal because operand 2 of the MOV instruction is of signed integer type, while the X0 element can store only Boolean data.

📖 Note

1. When the operand is of INT or WORD type, the applicable elements include KnX, KnY, KnM, KnS, KnLM, and KnSM, where $1 \leqslant n \leqslant 4$.
2. When the operand is of DINT or DWORD type, the applicable elements include KnX, KnY, KnM, KnS, KnLM, and KnSM, where $5 \leqslant n \leqslant 8$.
3. When the operand is of INT or WORD type, the applicable C elements are C0–C199.
4. When the operand is of DINT or DWORD type, the applicable C elements are C200–C255, and C301–C306.

3.2.3 Constant

You can use constants as the instruction operands. IVC2 series PLCs support a variety of constant mode inputs. Expressions of constants are shown in the following table.

Table 3-4 Expressions of constants

Constant type	Application instance	Valid range	Remarks
Decimal constant (16-bit signed integer)	−8949	−32768–+32767	
Decimal constant (16-bit unsigned integer)	65326	0–65535	
Decimal constant (32-bit signed integer)	−2147483646	−2147483648–+2147483647	
Decimal constant (32-bit unsigned integer)	4294967295	0–4294967295	
Hex constant (16-bit)	16#1FE9	16#0–16#FFFF	Hex, octal, and binary constants have no positive or negative meaning. If you select hex, octal or binary constants used as operands, positive and negative natures of these constants, and their sizes are determined by data types of these operands.
Hex constant (32-bit)	16#FD1EAFE9	16#0–16#FFFFFFFF	
Octal constant (16-bit)	8#7173	8#0–8#177777	
Octal constant (32-bit)	8#71732	8#0–8#37777777777	
Binary constant (16-bit)	2#10111001	2#0–2#1111111111111111	
Binary constant (32-bit)	2#101110011111	2#0–2#1111111111111111 1111111111111111	
Single-precision floating-point constants	−3.1415E-16 3.1415E+3 0.016	±1.175494E -38–±3.402823E +38	Compliance with the IEEE-754 standard. The programming software can display and input floating-point constants with 7-bit effective precision.

Chapter 4 Programming concept

This chapter detailedly describes the contents of IVC series micro-PLC programming, including the programming language and programelements. The programming and usage of subprograms arealso introduced, and finally, it describes some general explanation of instructions.

4.1 Introduction to programming languages

There are three programming languages: LAD, IL, and SFC.

4.1.1 LAD

- Concepts

  LAD is a widely-used graphical PLC programming language similar to the electrical (relay) control diagram. Its main features include:

  1. It is configured with the left bus, and the right bus is omitted.

  2. All control output elements (coils) and function blocks (application instructions) share the same energy flow input terminal.

  The electrical control diagram and LAD are equivalent to a certain degree, as shown in the following figure.

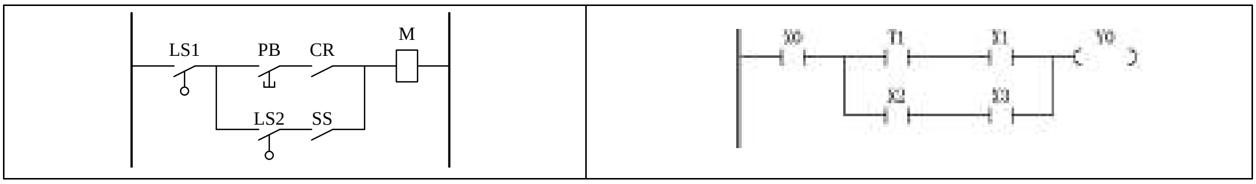


Figure 4-1 The equivalence between electrical control diagram and LAD

- LAD basic programming elements

  According to the principles in the electrical (relay) control diagram, several basic programming elements are abstracted for theLAD:

  1. Left bus: It corresponds to the control bus in electrical control diagram, and provides control power for the control circuit.

  2. Connecting line ( ): It represents the electrical connection in the electrical control diagram, and is used to conduct other inter-connected elements.

  3. Contact ( ): It represents the input contact in the electrical control diagram, it controls the ON/OFF and direction of control currents in the circuit. The parallel and serial connection of the contacts essentially represents the operational relationship of the input logic of the control circuit, and controls the transmission of the energy flow.

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

  5. Function block ( ): It is also known as the application instruction, and corresponds to the actuator or functional device that completes the special functions in the electrical control diagram. It can perform the specific control functions or control calculation functions (such as data transmission, data calculation, timer, counter, etc.).

- Energy flow

  Being an important concept in LAD, the energy flow is used to drive coil elements and application instructions, which is similar tothe control current output by the driving coil, and executed by the execution unit in the electrical control diagram.

  In LAD, the coils or the front end of the application instruction need to be connected to the energy flow. The coil element can be output and the application instructions can be effectively executed only when the energy flow is valid.

  The following figure demonstrates the energy flow transferin LAD and the drive of the energy flow to the coils or function blocks.

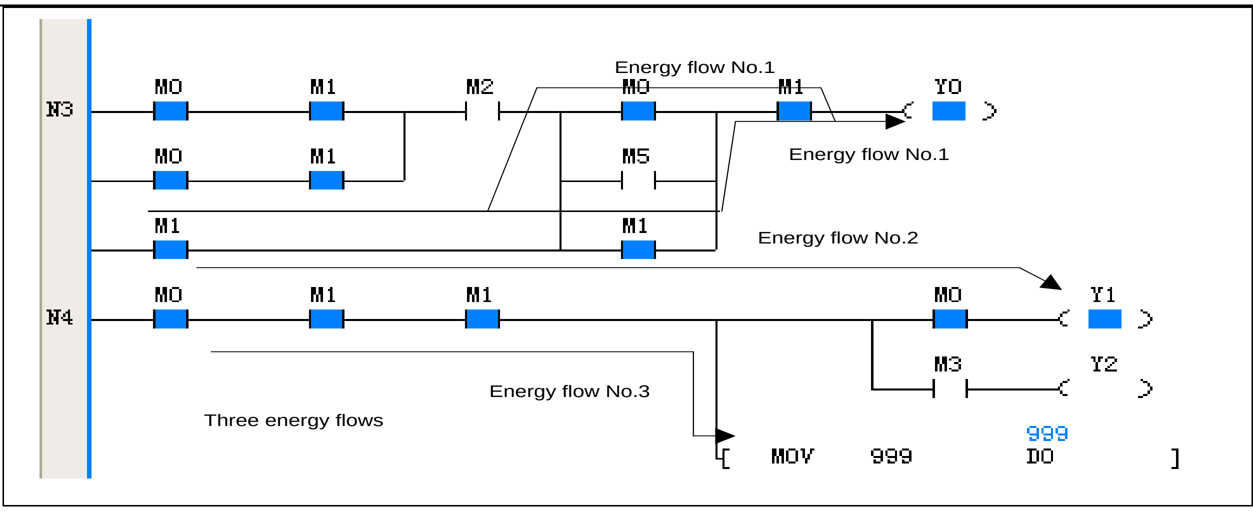


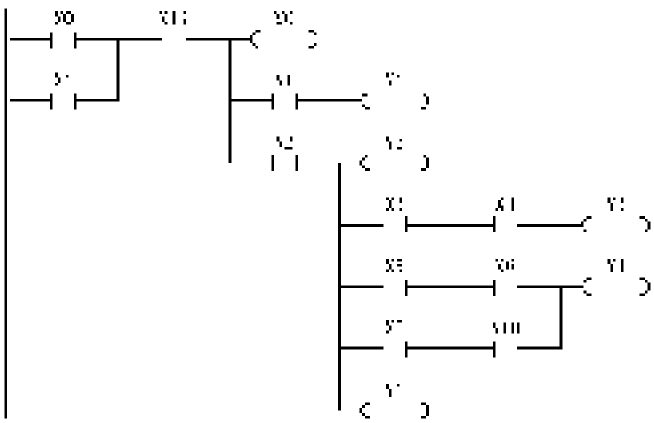
Figure 4-2 Energy flow and its driving function

4.1.2 IL

The IL is a textual user program, or a set of the instruction sequences written by the users.

The user program stored in the PLC main module for execution is actually the instruction sequence recognizable by the main module, and 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 application instance of converting a LAD into an IL.

LAD	1L	
	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 SFC

The SFC is a graphical design language for the user program framework that is commonly used to implement sequential control functions.

Sequence control is a control process that can be divided into multiple procedures (processing steps) and proceed them according to the certain working sequence.

The user program designed with SFC is relatively straight forward and clear because its program structure is consistent with the actual sequence control process.

The following figure is an application instance of a simple SFC.

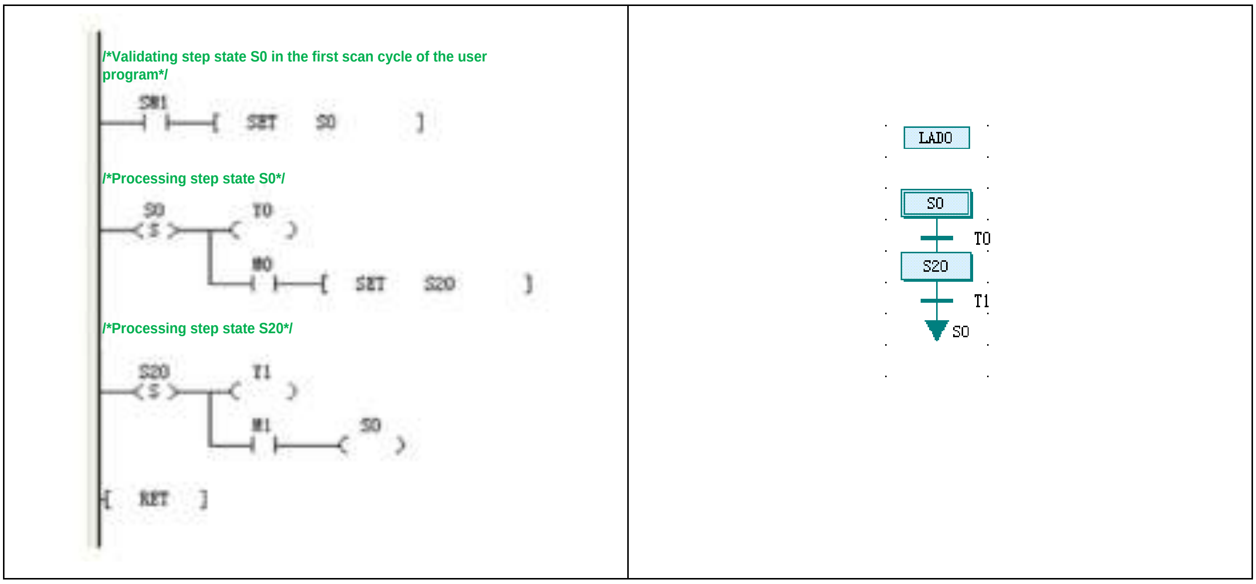


Figure 4-3 Application instance of SFC

4.2 Program Elements

The program elements include user program, system block and data block. You can change these program elements by programming.

4.2.1 User Program

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

The user program consists of three program organization units (POU): main program (MAIN), subprogram (SBR) and interrupt program (INT).

- MAIN

  The main program is the main body and framework of the user program. When the system is in RUN, the main program is executed cyclically.

  One user program has only one main program.

- SBR

  A subprogram is a structurally and functionally independent user program that can be called by other program bodies. Subprograms generally have call operand interfaces that are executed only when being called.

  A user program can have random number of subprograms, or no subprogram at all.

- INT

  An interrupt program is a user program segment that handles a specific interrupt event. A specific interrupt event always corresponds to aspecific interrupt program.

  Upon the occurrence of an interrupt event, a normal scan cycle is interrupted, and the user program flow automatically jumps to the execution of the interrupt program until the interrupt return instruction system returns to the normal scan cycle.

  A user program can have random number of interrupt programs, or no interrupt program at all.

4.2.2 System block

The system block contains multiple system configuration options. You can modify, compile and download the system block to configure the operation mode of the main module.

For details about how to use the system configuration items, refer tosection 2.2.1 "System block"in the manual or the related description in *Auto Station Programming Software User Manual*.

4.2.3 Data block

The data block contains the set value of D or R element. When the data block is downloaded to the controller, the designated D or R element is assigned a set value, thereby achieving the purpose of batch setting the D or R element value.

If the "Datablock enabled" in the advanced settings tab of the system block, the D or R elements are initialized by the data block before the user program is in RUN.

4.3 Block comment and variable comment of the program

4.3.1 Block comment

When programming, you can add the block comments to the program. Block comments textually describe the relevant programs. Each block comment takes up an entire line of space.

In the program, right click and select **Insert Row** to insert a row above the current row. You can use an empty row toseparate two program blocks.

To make a block comment, you need to first select an empty row, then right click and select **Insert Block Comment**, as shown below.

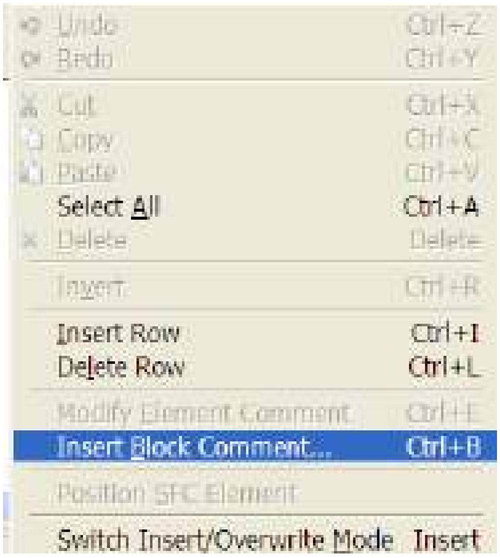


Figure 4-4 Adding block comment

Entering your comment into the block comment dialogue box that pops out and clicking the OK button.

Figure 4-5 Block comment dialogue box

The software automatically adds "/*" and "*/" to both sides of the entered text, and displays them in green color, as shown below:

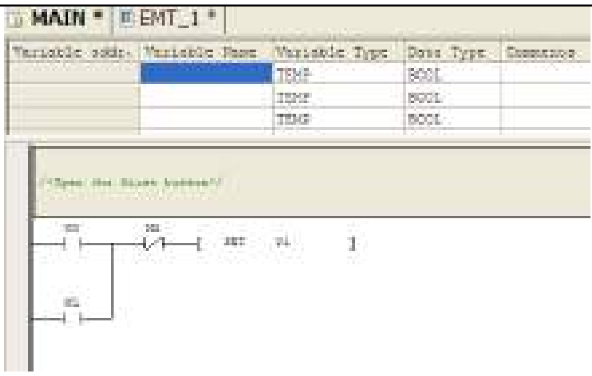


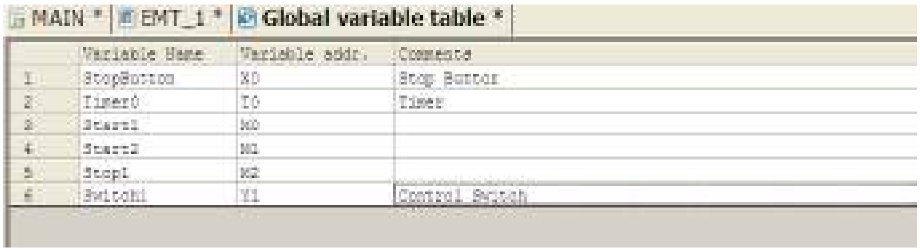
Figure 4-6 The clock comment in the program

A block comment occupies a whole row. You cannot add a block comment to an occupied row, nor can a row occupiedby a comment be used for other purposes.

4.3.2 Variable Comment

You can define the variables in the global variable table and local variable table. (For details, refer to section 2.2.3 "Global variable table" and section 4.4.3 "Definition of the SBR variable table"), and correctly defined variables can be used in the LAD. A variable can stand for a

certain address to make the program more readable. Figure 4-7 shows some variables defined in a global variable table.



	Variable Name	Variable addr.	Comments
1	StopButton	X0	Stop Button
2	Timer0	T0	Timer
3	Start1	M0	
4	Start2	M1	
5	Stop1	M2	
6	Switch1	Y1	Control Switch

Figure 4-7 Variables defined in the global variable table

- Symbol addressing

When the defined variables are used, you can switch between the variable name and the element address by selecting **Symbol addressing** menu.

The following figure shows the same LAD in both display modes.

The following figure shows the state of the LAD when the Symbol Addressing is not selected.

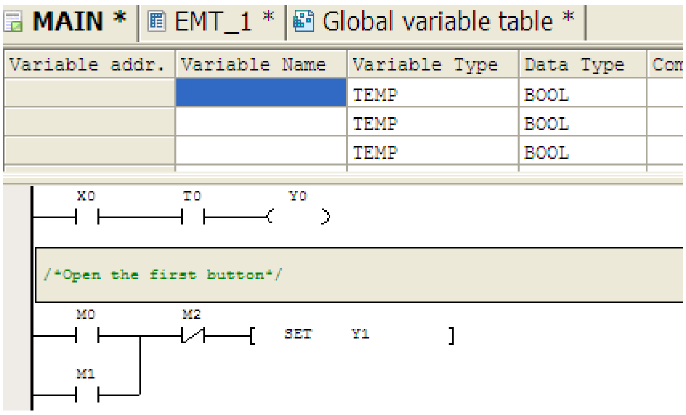


Figure 4-8 The state of symbol addressing is not selected

The following figure shows the state of the LAD when the Symbol Addressing is selected.

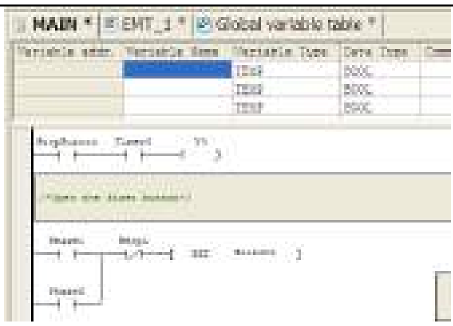


Figure 4-9 The state of symbol addressing is selected

- Elementcomment

You can control whether the element comments are displayed in the LAD by selecting the **Element comment** menu, as shown in Figure4-10.

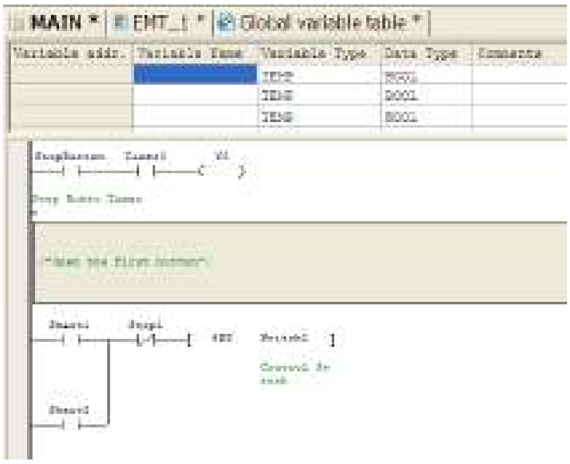


Figure 4-10 The LAD program displaying the element comments

Note

For the IVC2L and IVC3 series products, global variable table and local variable table can be downloaded to the PLCs for saving after compilation. Saving these comments needs to be supported by the backup battery.The battery failure may cause comment loss, failure to upload the comments, and the system reports a user information file error, but the user program can still run normally.

4.4 Subprogram

4.4.1 Concept

Being an optional part of the user program, a subprogram (SBR) is an independent program organization unit (POU) that can be called by the main program or other SBRs.

You can use SBRs in your user program to:

1. Reduce the size of the user program. You can write a repeated program section as a SBR and call it whenever necessary.

2. Clarify the program structure, particularly the structure of the main program.

3. Make the user program more transplantable.

4.4.2 Notestouse SBRs

When writing or calling a SBR, You need to note the following:

1. The PLCs support a maximum of 6 levels of SBR nesting calling.

The following is an fine application instance of 6-level of SBR nesting calling:

MAIN→SBR1→SBR2→SBR3→SBR4→SBR5→SBR6

(where the "→" represents calling the corresponding SBRs with the CALL instruction)

2. The PLCs do not support recursive call and cyclic call of SBRs.

The following two application instances show two illegal SBR callings.

- MAIN→SBR0→SBR0 (recursive call, illegal)
- MAIN→SBR0→SBR1→SBR0 (cyclic call, illegal)

3. A maximum of 64 SBRs can be defined in a user program.

4. A maximum of 16 bit and 16 word types of variables can be defined in the variable table of a SBR.

5. When calling a SBR, you need to note that the operand type of the CALL instruction needs to match the variable type defined in the variable table of theSBR, and the compiler checks whether the match is correct.

6. The interrupt programs are not allowed to call SBRs

4.4.3 Definition of the SBR variable table

- SBR variabletable
  The SBR variable table is used to displays all SBR interface parameters and local variables (collectively referred to as variables) and specify their properties.
- SBR variable properties
  The SBR variables (including interface parameters and local variables) have the following properties:

  1. Variable address

  Based on the variable data type, the software automatically assign a fixed LM or V element address to each SBR variable in sequence.

  2. Variable name

  You can give each SBR variable a name (alias). You can use a variable in the program by quoting its name.

  3. Variable type

  The SBR variables are classified into the four types:IN, OUT, IN_OUT, and TEMP.

  - IN: The IN type variable is used to transfer the input value of the SBR when a SBR is being called.
  - OUT: The OUT type variable is used to call the return value for transferring the SBR when a SBR returns.
  - IN_OUT: The IN type variable is used to transfer the input value of the SBR when a SBR is being called, or call the return value for transferring the SBR when a SBR returns.
  - The TEMP type variables are only used as local variables that are valid only within scope of the SBR.

4. Variable data type

The properties of the variable data types specify the data widthand range of the variables. The variable data types are listed in the following table.

Table 4-1 Variable data types

Data type	Description	Occupied LM/V element address
BOOL	Bit type	One LM element address
INT	Signed integer type	One V element address
DINT	Signed double integer type	Two consecutive V element addresses
WORD	Word type	One V element address
DWORD	Double word type	Two consecutive V element addresses
REAL	Floating-point type	Two consecutive V element addresses

4.4.4 SBR parameter transfer

If local input or output variables are defined in the SBR when a SBR is called in the main program, you need to input the corresponding variable values, global or temporary variable elements into the SBR interface parameters. You need to note that the local variable need to be of the same data type with the interface parameter.

4.4.5 SBR application instance

Here is an application instance of how to write and call a SBR.

- Introduction to the function of the application instance
  Calling SBR_1 in the main program to perform the addition calculation (3=+2) of two integer constants, and assigning the operation result 5 to D0.
- Operation procedures of the application instance
  Step 1: Creating a SBR in the project and naming it as SBR_1.

  Step 2: Writing the SBR_1.

  1. Creating the SBR calling operand interface through the variable table of the SBR_1.

  1) Defining variable 1: Naming it as IN1 (variable type: IN). It is used as the INT type data and sequentially assigned a V elementaddress V0.

  2) Defining variable 2: Naming it as IN2 (variable type: IN). It is used as the INT type data and sequentially assigned a V elementaddress V1.

  3) Defining variable 3: Naming it as OUT1 (variable type: OUT). It is used as the INT type data and sequentially assigned a V elementaddress V2.

  2. Writing the SBR_1 as:

  LD SM0

  ADD #IN1 #IN2 #OUT1

  The writing process of the SBR_1 is shown in the following figure.

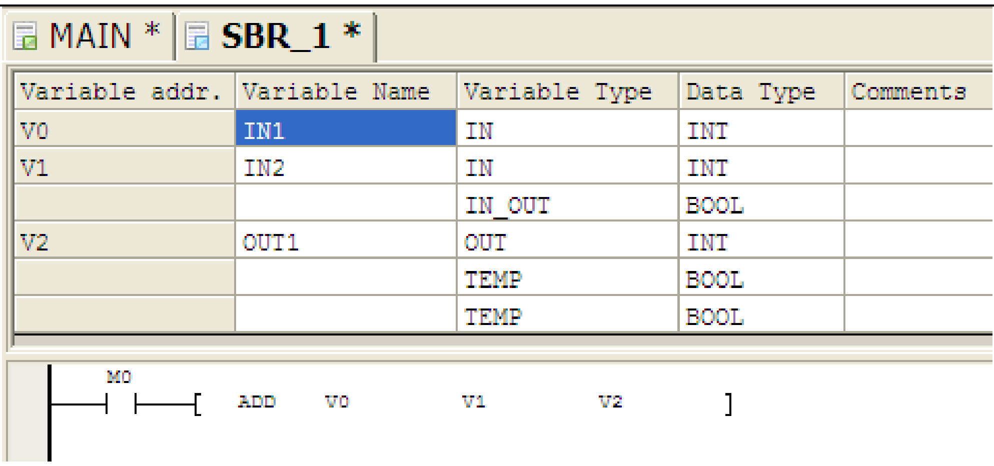


Figure 4-11 Writing process of the SBR_1

Step 3: Writing the main program and calling the SBR

Using the CALL instruction in the main program to call the SBR_1.

The corresponding main program is as follows:

LD M0

CALL SBR_1 3 2 D0

You can use the parameters to transfer the corresponding relationship table and filling in the parameters that are brought in or returned when the SBR is called.

- Parameter IN1 is brought in to transfer the constant integer 3
- Parameter IN2 is brought in to transfer the constant integer 2
- The return value OUT1 is stored in D0

The above program is shown in the following figure.

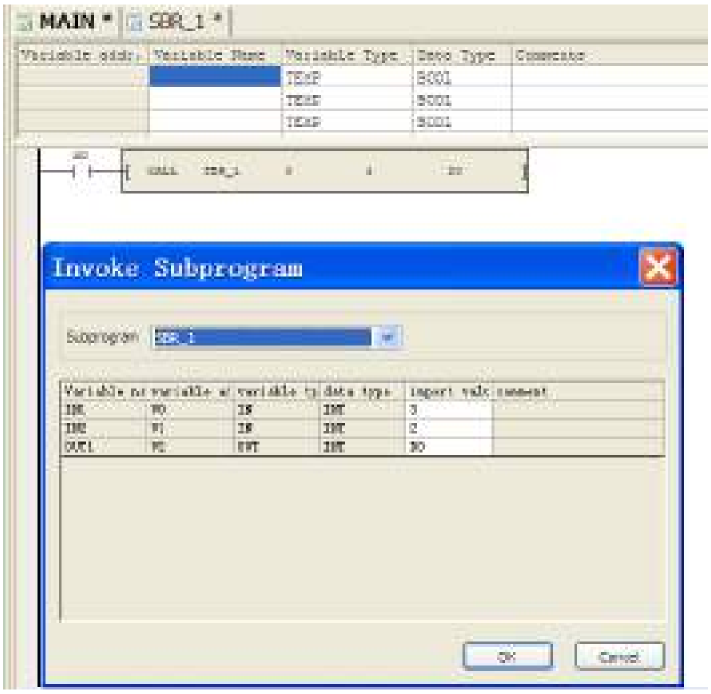


Figure 4-12 Calling the SBR

Step 4: Compiling, downloading and running the user program to verify the logic of the subprogram.

- Execution result of the application instance

When M0 is ON, SBR_1 is called. Values 2 and 3 are transferred to the operands IN1 and IN2 to carry out the addition operation. The result 5 is then returned to the main program, that is, D0 is 5.

4.5 General information of the instructions

4.5.1 Instruction operands

The instruction operands can be classified into the following two types:

- Source operands: Or S (or S1, S2, S3 … when there are more than one of them in the same instruction). The instruction reads values from source operands for calculation.
- Destination operands: or D (or D1, D2, D3 … when there are more than one of them in the same instruction). The instruction controls or outputs values to the destination operands.

The operands could be bit elements, word elements, double word elements, or constants. For details about the instruction description, refer to Chapter 5 and Chapter 6.

4.5.2 Flag bit

The instruction operations may affect the following three kinds of flags bits.

- Zero flag SM180
  Setting the zero flag when the instruction operation result is zero.
- Carry flag SM181
  Setting the carry flagwhen the instruction operation result is a carry.
- Borrow flag SM182
  Setting the borrow flag when the instruction operation result is a borrow.

4.5.3 Restrictions on the use of instructions

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

- Exclusive hardware resources
  Some instructions requires hardware resources. When a specific hardware is being used by a certain instruction, the access to the hardware is denied to other instructions, because the occupation of the resource is exclusive.

  Taking the high-speed counting instructions and SPD instruction for application instance. Any of these instructions can occupy certain input points among X0–X7. The limited resources make it impossible to execute these instructions at the same time.

- Exclusive time
  The execution of certain instructions may take a period of time. Therefore, when using these instructions, you need to ensure that the instructions have enough time to complete the function. Only one can be executed at a certain time when the system is running.

  Taking the XMT instruction for application instance. Due to the timeliness of communication, the XMT instruction is sent to the free port, and only one can be executed at the same time. Similarly, the free port can execute only one RCV instruction once. Everytime when a Modbusinstruction is being executed, the communication channel is unavailable to other instructions for a while. The same applies to other instructions such as high-speed output instruction, positioning instructions and inverter instructions.

- Application limit of the instructions
  Some instructions cannot be used in certain situations due to their limited application scope.

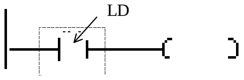
  For example, the instruction pair MC/MCR cannot be used in the step state programmed by SFC.

Chapter 5 Basic instructions

This chapter detailedly describes basic instructions of IVC series micro-PLCs, including the instruction format (form), operand,influenced flag bit, function, application instance, and sequence diagram.

5.1 Contact logic instructions

5.1.1 LD: NO contact instruction

<table>
<tr><td colspan="9" rowspan="2">LAD:
</td><td colspan="4">Applicable model</td><td colspan="3">IVC1 IVC1S IVC3 IVC2L IVC1L</td></tr>
<tr><td colspan="4">Influenced flag bit</td><td colspan="3"></td></tr>
<tr><td colspan="9">IL: LD (S)</td><td colspan="4">Step length</td><td colspan="3">1</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="13">Applicable soft element</td><td>Indexing</td></tr>
<tr><td>S</td><td>BOOL</td><td>X</td><td>Y</td><td>M</td><td>S</td><td>LM</td><td>SM</td><td></td><td>Dx.y</td><td></td><td>C</td><td>T</td><td></td><td></td><td></td></tr>
</table>

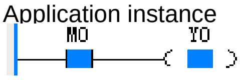
Operand description

S: Source operand

Function description

Connecting to the left bus to connect (state: ON) or disconnect (state: OFF) the energy flow.

Application instance

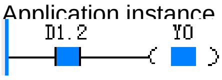


LD M0

OUT Y0

When M0 is ON, the Y0 output is ON.

Application instance



LD D1.2

OUT Y0

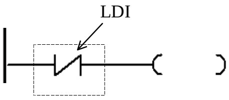
When the second bit of D1 is 1, the Y0 output is ON.

Note:

For contact logic instructions of the IVC1 series, when the operand ranges from M1536 to M2047, the step length is the step length assigned by each instruction plus one.

For contact logic instructions of the IVC3 series, when the operand ranges from M1536 to M10240, C256 to C511, T256 to T511, or S0 to S4096, the step length is the step length assigned by each instruction plus one. When you operate Dx.y, the step length is 4.

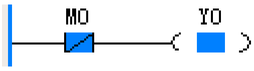
5.1.2 LDI: NC contact instruction

<table>
<tr><td colspan="9" rowspan="2">LAD:
</td><td colspan="4">Applicable model</td><td colspan="3">IVC1 IVC1S IVC3 IVC2L IVC1L</td></tr>
<tr><td colspan="4">Influenced flag bit</td><td colspan="3"></td></tr>
<tr><td colspan="9">IL: LDI (S)</td><td colspan="4">Step length</td><td colspan="3">1</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="13">Applicable soft element</td><td>Indexing</td></tr>
<tr><td>S</td><td>BOOL</td><td>X</td><td>Y</td><td>M</td><td>S</td><td>LM</td><td>SM</td><td></td><td>Dx.y</td><td></td><td>C</td><td>T</td><td></td><td></td><td></td></tr>
</table>

Operand description

S: Source operand

Application instance



LDI M0

OUT Y0

Function description

Connecting to the left bus to connect (state: ON) or disconnect (state: OFF) the energy flow.

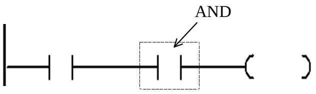
When M0 is OFF, the Y0 output is ON.

Note:

For the contact logic instructions of the IVC1 series, when the operand ranges from M1536 to M2047, the step length is the step length assigned by each instruction plus one.

For contact logic instructions of the IVC3 series, when the operand ranges from M1536 to M10240, C256 to C511, T256 to T511, or S0 to S4096, the step length is the step length assigned by each instruction plus one. When you operate Dx.y, the step length is 4.

5.1.3 AND: NO contact and instruction

LAD:	Applicable model	IVC1 IVC1S IVC2L VC3 IVC1L
	Influenced flag bit	
IL: AND *(S)*	Step length	1

Operand	Type	Applicable soft element													Indexing
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

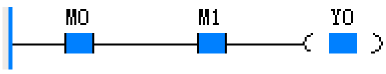
Operand description

S: Source operand

Function description

Performing the "AND" operation on the ON/OFF state of the designated contact (***S***) and the current energy flow, and assigning the obtained result to the current energy flow.

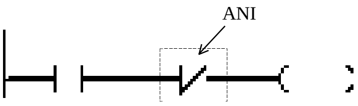
Application instance




```
LD   M0
AND  M1
OUT  Y0
```

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

5.1.4 ANI: NC contact and instruction

LAD:	Applicable model	IVC1 IVC1S IVC2L IVC3 IVC1L
	Influenced flag bit	
IL: ANI *(S)*	Step length	1

Operand	Type	Applicable soft element													Indexing
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

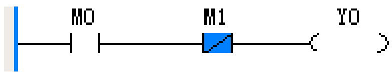
Operand description

S: Source operand

Function description

Performing the "NOT" operation on the ON/OFF state of the designated contact (S), and then performing the "AND" operation on the obtained result and the current energy flow value, and assigning the obtained result to the current energy flow.

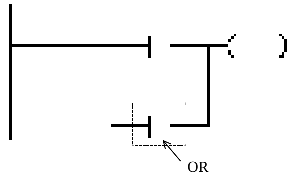
Application instance



LD M0
ANI M1
OUT Y0

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

5.1.5 OR: NO contact or instruction

LAD: 	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: OR *(S)*	**Step length**	**1**

Operand	Type	Applicable soft element													Indexing
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

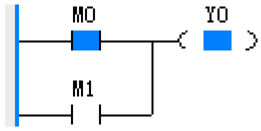
Operand description

S: Source operand

Function description

Performing the "OR" operation on the ON/OFF state of the designated contact (S) and the current energy flow, and assigning the obtained result to thecurrent energy flow.

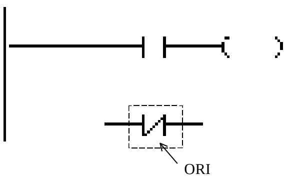
Application instance



LD M0
OR M1
OUT Y0

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

5.1.6 ORI: NO contact or instruction

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: ORI *(S)*	**Step length**	**1**

Operand	Type	Applicable soft element													Indexing
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T			

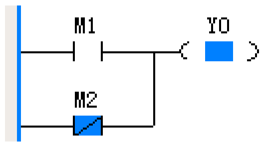
Operand description

S: Source operand

Function description

Performing the "NOT" operation on the ON/OFF

Application instance

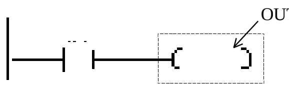


LD M1
ORI M2
OUT Y0

state of the designated contact (***S***), and then performing the "OR" operation on the obtained result and the current energy flow value, and assigning the obtained result to the current energy flow.

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

5.1.7 OUT: Coil output instruction

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	**Influenced flag bit**	
IL: OUT *(S)*	**Step length**	**1**

Operand	Type	Applicable soft element												Indexing
S	BOOL	X	Y	M	S	LM	SM		Dx.y		C	T		

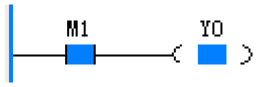
Operand description

S: Source operand

Function description

Assigning the value of the current energy flow to the designated coil (***D***).

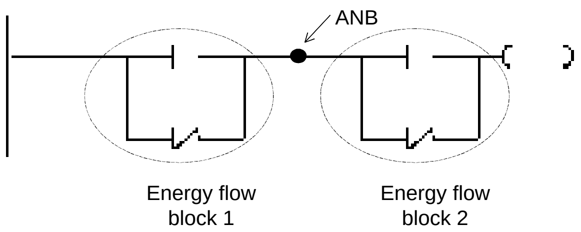
Application instance



LD M1
OUT Y0

When M1 is ON, the Y0 output is ON.

5.1.8 ANB: Energy flow block and instruction

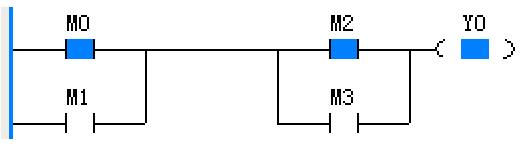
LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	**Influenced flag bit**	
IL: ANB	**Step length**	**1**

Operand description

Function description

Conducting "AND"operation on the energy flow values of two energy flow blocks, and then assigning the obtained result tothe current energy flow.

Application instance

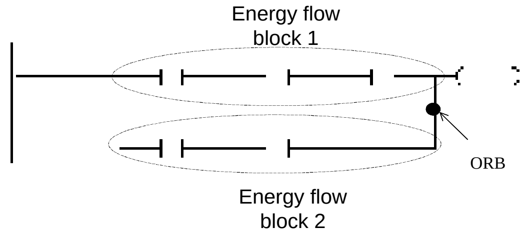


LD M0
OR M1
LD M2
OR M3
ANB
OUT Y0

When either M0 or M1 is on, and either M2 or M3 is ON, the Y0 output is ON.

5.1.9 ORB: Energy flow block or instruction

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**

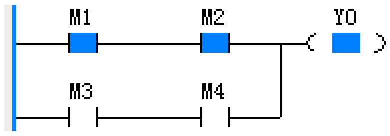
	**Influenced flag bit**	
IL: ORB	**Step length**	**1**

Operand description

Function description

Conducting "OR" operation on the energy flow values of two energy flow blocks, and then assigning the obtained result to the current energy flow.

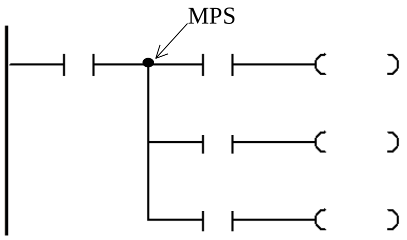
Application instance



LD M1
AND M2
LD M3
AND M4
ORB
OUT Y0

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

5.1.10 MPS: Output energy flow push instruction

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: MPS	**Step length**	**1**

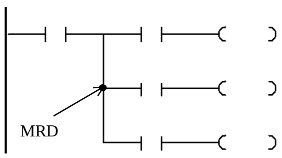
Function description

Pushing and storing the current energy values for the energy flow calculation of subsequent output branches.

Note:

It is not allowed to use MPS instructions(without MPP instructions among them) for over 8 consecutive times in only one ladder diagram network,otherwise the energy flow output stack may overflow.

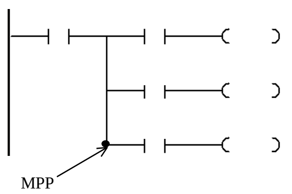
5.1.11 MRD: Instruction for reading output energy flow stack top value

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: MRD	**Step length**	**1**

Function description

Assigning the top value of the energy flow output stack to the current energy flow.

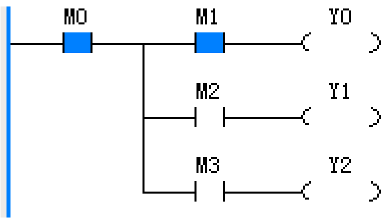
5.1.12 MPP: Output energy flow stack pop instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: MPP	Step length	1

Function description

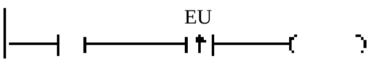
Performing the Pop operation on the energy flow output stack, andassigning the popped value to the current energy flow.

Application instance



LDM0
MPS
AND M1
OUT Y0
MRD
AND M2
OUT Y1
MPP
AND M3
OUT Y2

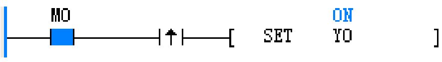
5.1.13 EU: Rising edge detection instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: EU	Step length	2

Function description

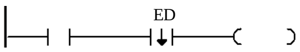
Comparing the state of the input energy flow in the current scan with that in the last scan. If the energy flow rises (OFF→ON), the output in the current scan cycle is valid.

Application instance



LD M0
EU
SET Y0

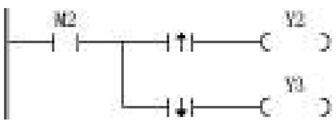
5.1.14 ED: Falling edge detection instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: ED	Step length	2

Function description

Comparing the state of the input energy flow in the current scan with that in the last scan. If the energy flow falls (OFF→ON), the output in the current scan cycleis valid.

Application instance

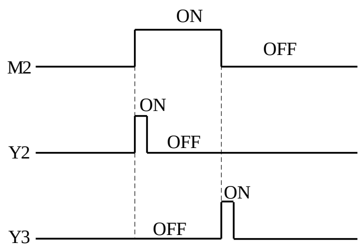


LD M2
MPS
EU
OUT Y2
MPP
ED
OUT Y3

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

2. In two consecutive scan cycles, the states of M2 contact are ON and OFF respectively, and the ED instruction detects a falling edge edge, so that Y3 outputsan ON state with a scan cycle width.

Sequence chart of the application instance

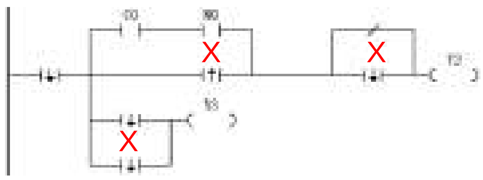


Note

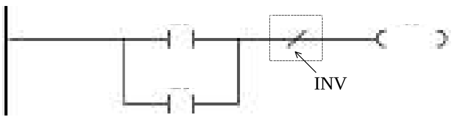
In LAD, the rising edge contact andfalling edge contact instructions need to be used in series rather than in parallel with other contact elements.

In LAD, the rising edge contact and falling edge contact instructions cannot be connected to the left energy flow bus directly.

The following figure shows an example of incorrect use of EU and ED instructions in LAD.



5.1.15 INV: Energy flow negation instruction

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: INV	**Step length**	**1**

Function description

Performing the "INV" operation on the current energy flow value and then assigning the obtained result to the current energy flow.

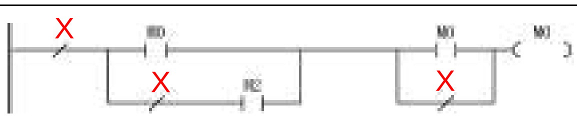
Note

In LAD, the INV instruction needs to be used in series rather than in parallel with other contact elements.

INV cannot be used as the first instruction in the parallel input branches.

In LAD, the INV instruction cannot be directly connected to the left energy flow bus.

The following figure shows an example of incorrect use of the INV instruction in LAD.



5.1.16 SET: Coil set instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: SET *(S)*	Step length	1

Operand	Type	Applicable soft element													Indexing
S	BOOL		Y	M	S	LM	SM		Dx.y		C	T			

Operand description

S: Source operand

Function description

Setting the bit element assigned by ***D*** when the energy flow is valid.

Application instance

LD M0
SET M1

5.1.17 RST: Coil reset instruction

LAD: RST (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: RST *(S)*	Step length	1

Operand	Type	Applicable soft element													Indexing
S	BOOL		Y	M	S	LM	SM		Dx.y		C	T			

Operand description

S: Source operand

Function description

Resetting the specified bit element (***D***) to zero when the energy flow is valid.

Application instance

LD M0
RST M1

Note

If D is a C element, the corresponding count value is deleted. If D is a T element, the corresponding timing value is deleted.

5.1.18 NOP:No operation instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: NOP	Step length	1

Function description

Performing no operation.

Note

In LAD, this instruction cannot be directly connected to the left energy flow bus.

5.2 Main control instructions

5.2.1 MC: Main control instruction

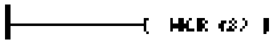
LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: MC *(S)*	Step length	3

Operand	Type	Applicable soft element													Indexing
S	INT	Constant													

Operand description

S: Source operand

5.2.2 MCR: Main control reset instruction

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **MCR *(S)***	**Step length**	**1**

Operand	Type	Applicable soft element													Indexing
S	INT	Constant													

Operand description

S: Source operand

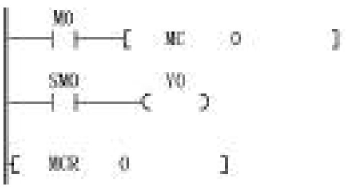
Function description

1. Matching the MC and MCR instructions to form an MC-MCR structure. The MC instruction represents the beginning of an MC-MCR structure, and its operand Sis the label of the MC-MCR structure. The value of operand S is a constantranging from 0 to 7. MCR represents the end of an MC-MCR structure.

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

3. Skipping over the instruction in middle of the MC-MCR structure when the energy flow before the MC instruction is invalid, and executing it and deleting the destination operands corresponding to OUT, TON, TOF, PWM,HCNT, PLSY,PLSR, DHSCS, SPD, DHSCI, DHSCR, DHSZ,DHST, DHSP and BOUT in the structure after the program directly jumps to the structure

Application instance




```
LD   M0
MC   0
LD   SM0
OUT  Y0
MCR  0
```

When M0 = ON, instructions in the MC 0–MCR 0 structure are executed, and Y0 = ON. When M0 = OFF,instructions in the MC 0–MCR 0 structure are not be executed,and the bit element Y0 designated by the destination operandof the OUT instruction in the structure is deleted, Y0 = OFF.

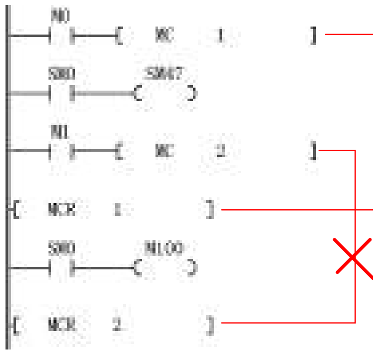
Note

1. In LAD, the MCR instruction must be directly connected to the left energy flow bus.

2. In LAD, the MCR instruction cannot connect in parallel or in series with other instructions.

3. Multiple different numbered MC-MCR structures can be used in the nested mode, but thenumber of nest levels cannot exceed 8. MC-MCR structures with the same number cannotbe used in the nested mode.

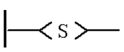
4. Two MC-MCR structures cannot be used in the cross manner. The use method shown in the following figure is illegal.



Note: It cannot be used in SFC programming.

5.3 SFC instructions

5.3.1 STL: SFC state loadinginstruction

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **STL *(S)***	**Step length**	**3**

Operand	Type	Applicable soft element													Indexing
S	BOOL						S								

Operand description

S: Source operand

Function description

1. Indicating the beginning of the process of a step state (***S***).
2. Executing thebuilt-in instructions of a step when the state of the step is valid (ON).
3. If the state of a step changes from ON to OFF (falling edge change), not executing the built-in instructions of the step, and deleting the destination operands corresponding to OUT, TON, TOF, PWM, HCNT,PLSY, PLSR, DHSCS, SPD, DHSCI, DHSCR,DHSZ, DHST, DHSP and BOUT.
4. Not executing the built-in instructions of a step when the state of the step is invalid (OFF).
5. Defining a parallel merging structure by consecutive STL instructions (serial connectionof STL elements), where the STL instructions can be used in a maximum of 16 consecutive times, that is, the maximum number of branches in the parallel merging structure is 16.

5.3.2 SET Sxx: SFC state transition instruction

LAD: |—< S >—| |—[SET (D)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **SET *(D)***	**Step length**	**3**

Operand	Type	Applicable soft element													Indexing
D	BOOL						S								

Operand description

D:Destination operand

Function description

Setting the state of the specified step (***D***) to be valid when the energy flow is valid, and setting the current valid step to be invalid to complete the state transition.

5.3.3 OUT Sxx: SFC state jumpinstruction

LAD: |—< S >—| |—( )	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **OUT *(D)***	**Step length**	**3**

Operand	Type	Applicable soft element													Indexing
D	BOOL						S								

Operand description

D:Destination operand

Function description

Setting the state of the specified step (***D***) to be valid when the energy flow is valid, and setting the current valid step to be invalid to complete the state jump.

5.3.4 RST Sxx: SFC state reset instruction

LAD: |—< >—| |—[RST (D)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **RST *(D)***	**Step length**	**3**

Operand	Type	Applicable soft element													Indexing
D	BOOL						S								

Operand description

D: Destination operand

Function description

Setting the state of the specified step (***D***) to be invalid when the energy flow is valid.

5.3.5 RET: SFC program segment end instruction

LAD: |[RET]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	

IL: RET	Step length	1

Function description

Indicating the end of a segment of the SFC program.

Note

It can be used only in the main program.

5.4 Timer instructions

5.4.1 TON: Turn-on delay timing instruction

LAD: TON (D) (S)	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: TON (D) (S)	Step length	5

Operand	Type	Applicable soft element														Indexing
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

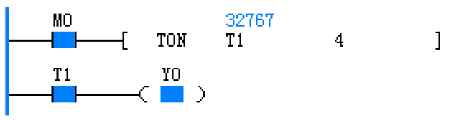
1. Timing the specified T element (D) when the energy flow is valid and the timing value is less than 32,767 (the timing value increases with the time), and remaining in 32,767 after it is reached.

2. Setting the timing coil output of the designated Telement to ON when the timing value is more than or equal to the preset value (S).

3. Stopping timing, resetting the timing value to zero, and setting the timing coil output to OFF when the energy flow is OFF.

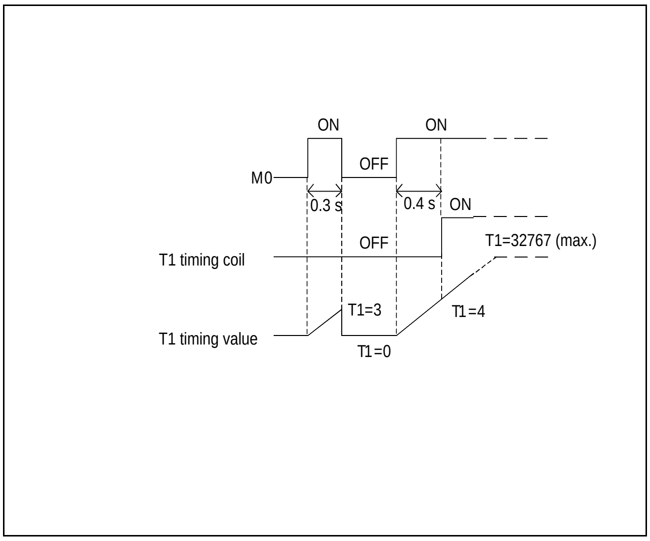
4. Setting the timing coil value of the designated T element to OFF and resetting the timing value to zero when the system executes the instruction for the first time.

Application instance




```
LD   M0
TON  T1  4
LD   T1
OUT  Y0
```

Sequence chart of the application instance



5.4.2 TONR: Memory-type turn-on delay timing instruction

LAD: TONR (D) (S)	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: TONR (D) (S)	Step length	5

Operand	Type	Applicable soft element														Indexing
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

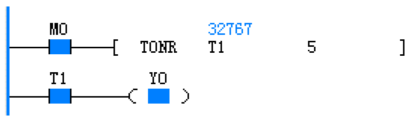
Function description

1. Timing the specified T element (***D***) when the energy flow is valid and the timing value is less than 32,767 (the timing value increases with the time), and remaining in 32,767 after it is reached.

2. Setting the timing coil output of the designated T element to ON when the timing value is more than or equal to the preset value (***S***).

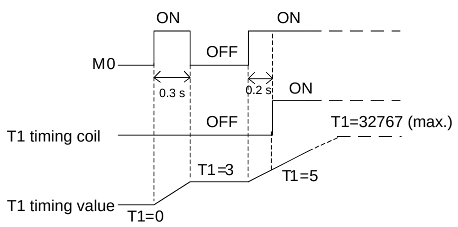
3. Stopping timing, and remaining the current timing coil and timing value when the energy flow is OFF.

Application instance



LD M0
TONR T1 5
LD T1
OUT Y0

Sequence chart of the application instance



5.4.3 TOF: Turn-off delay timing instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
┣┫ ├─[TOF (D) (S)]	Influenced flag bit	
IL: TOF (D) (S)	Step length	5

Operand	Type	Applicable soft element														Indexing
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

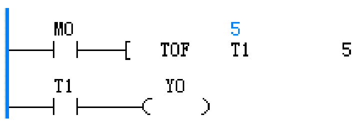
1. Starting the timing of the specified timer T (***D***) when the energy flow changes from ON to OFF (falling edge change).

2. Keeping timing when the energy flow is OFF but the specified timer T has been started, and setting the timing coil output of the T element to OFF when the timing value reaches the preset value (***S***), and then remaining in the preset value.

3. Not timing when the timing has not started but the energy flow input is OFF.

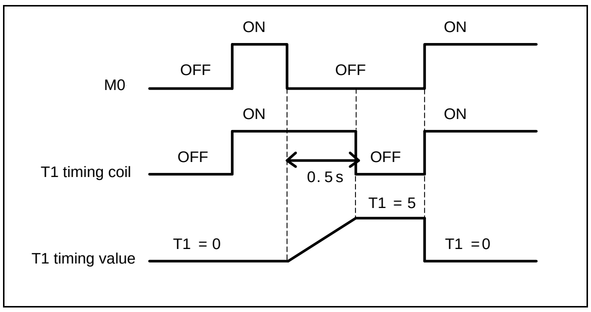
4. Stopping timing, resetting the timing value to zero, and setting the timing coil output to ON when the energy flow is ON.

Application instance



LD M0
TOF T1 5
LD T1
OUT Y0

Sequence chart of the application instance



5.4.4 TMON: Non-retriggerable monostable timing instruction

LAD:	Applicable model	IVC1IVC1SIVC3 IVC2L IVC1L

─┤ ├──[TMON (D) (S)]		**Influenced flag bit**	
IL: TMON *(D)* *(S)*		**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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

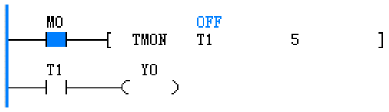
Function description

1. Starting the timing of the designated timer T (***D***) (based on the current value) when the input energy flow changes from OFF to ON (rising edge change), and the timing has not started, and keeping the timing coil output ON when in the timing state (the timing length is determined by ***S***).

2. Keeping timing and keeping the timing coil output ON no matter how the energy flow changes when it is in the timing state (the timing length is determined by ***S***).

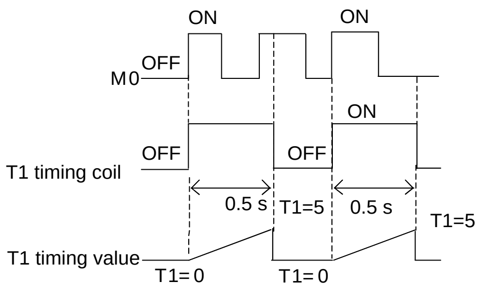
3. Stopping timing, resetting the timing value to zero, and setting the timing coil output to OFF when the timing value is reaches the preset point.

Application instance



LD M0
TMON T1 5
LD T1
OUT Y0

Sequence chart of the application instance



5.5 Counter instructions

5.5.1 CTU: 16-bit increment counterinstruction

LAD:		**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
─┤ ├──[CTU (D) (S)]		**Influenced flag bit**	
IL: CTU *(D)* *(S)*		**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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. Increasing the count value of the specified16-bit counter C (D) by 1 when the input energy flow changes from OFF to ON (rising edge change).

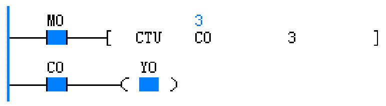
2. Remaining in the count value when it reaches 32,767.

3. Setting the counting coil to ON when the count value is larger than or equal to the preset point (***S***).

Note

Setting the address of the 16-bit counter C (***D***) to a value ranging from C0 to C199.

Application instance



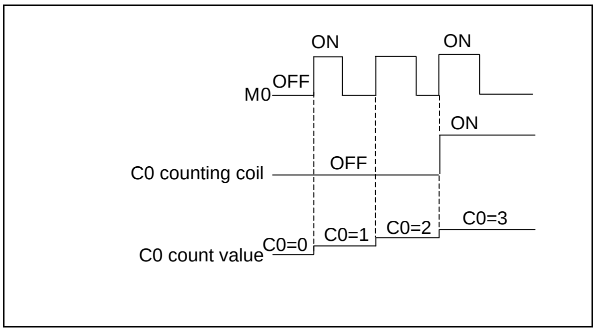
LD M0

CTU C0 3

LD C0

OUT Y0

Time sequence diagram instance



5.5.2 CTR: 16-bit cycliccountinginstruction

LAD: ⊢⊣ ⊢[CTR *(D)* *(S)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: CTR *(D)* *(S)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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. Increasing the count value of the specified 16-bit counter C (***D***) by 1 when the input energy flow changes from OFF to ON (rising edge change).

2. Setting the counting coil to ON, when the count value is equal to the preset point (***S***).

3. Setting the count value to 1, and the counting coil to OFF when the count value reaches the preset point (***S***),and the input energy flow changes from OFF to ON again (rising edge change).

Note

1. Performing no counting action when the preset count value (***S***) is less than or equal to 0.

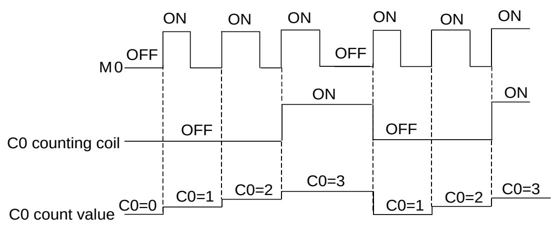
2. Setting the address of the 16-bit counter C (***D***) to a value ranging from C0 to C199.

Application instance

LD M0

CTR C0 3

Sequence diagram of the Application instance



5.5.3 DCNT:32-bit increment and decrement counting instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
├──┤ ├──[DCNT (D) (S)]	Influenced flag bit	
IL: DCNT (D) (S)	Step length	7

Operand	Type	Applicable soft element														Indexing
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. Increasing or decreasing the count value of the specified 32-bit counter C (***D)*** by 1 when the input energy flow changes from OFF to ON (rising edge change) (the increase and decrease depend on the corresponding SM flag bit).

2. For increment counters, setting the counting coil to ON when the count value is larger than or equal to the preset point (***S***).

3. For decrement counters, setting the counting coil to OFF when the count value is less than or equal to the preset point (***S***).

4. Changing the count value to –2147483648 when the count value is 2147483647 and is increased by 1.

5. Changing the count value to 2147483647 when the count value is –2147483648 and is decreased by 1.

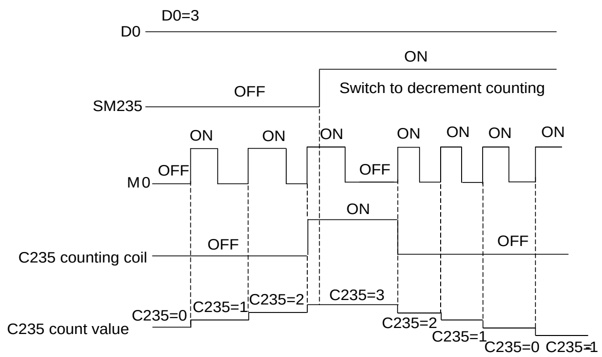
Note

The address of C element (***D***) must be within the range of C200 to C235.

Application instance

LD M0
DCNT C235 D0

Sequence diagram of the application instance



Chapter 6 Application instructions

This chapter detailedly describes the application instructions of IVC series micro-PLCs, including the instruction format (form), operand, influenced flag bit, function, application instance, and sequence diagram.

6.1 Program flow control instructions

6.1.1 FOR: Cycle instruction

LAD: FOR (S)	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: FOR (S)	Step length	3

Operand	Type	Applicable soft element														Indexing
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: Cycle return instruction

LAD: NEXT	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: NEXT	Step length	1

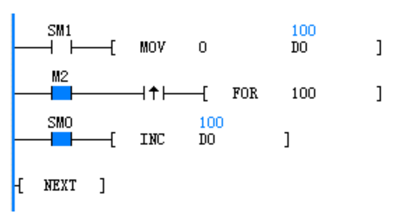
- Function description

  1. FOR and NEXT instructionsform a FOR-NEXT structure.

  2. Cyclically executing the instructions in the middle of the FOR-NEXT structure by S times continuously when the energy flow before FOR is valid and the cycle times (***S***) is larger than 0. Continuing to execute the instructions after the FOR-NEXT structure when the cyclic execution is done by S times.

  3. Not executing the instructions in the middle of the FOR-NEXT structure, directly jumping to the FOR-NEXT structure to continue the execution when the energy flow before FOR is invalid, or the cycle times (***S***) is less than or equal to zero.

- Application instance



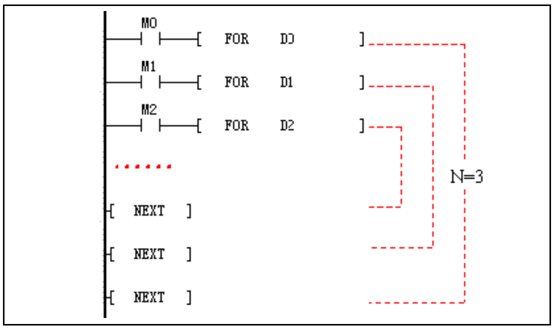

```
LD   SM1
MOV  0
D0
LD   M2
EU
FOR  100
LD   SM0
INC  D10
NEXT
```

The initial conditions of the operation are: D0=0, and M2=OFF. When M2 changes from OFF to ON, the instructions within the FOR-NEXT structure are consecutively executed for 100 times. D0 is increased by one for 100 times. When the cycle is over, D0=100.

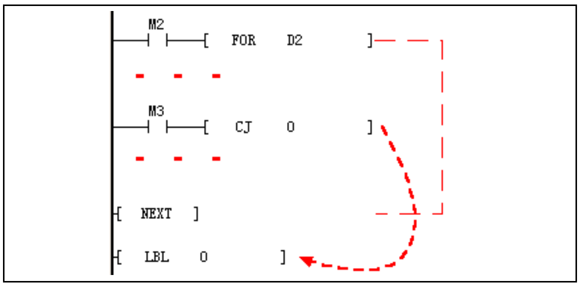
- Note

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

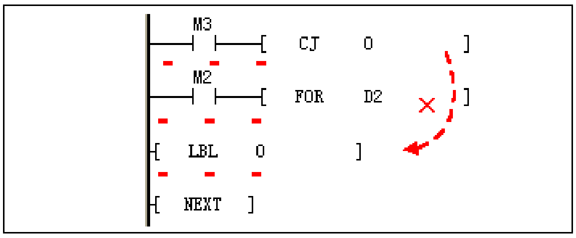
  2. Supporting the nesting of multiple FOR-NEXT structures. The CPU unit of the IVC2 series supports the nesting of a maximum of 8 FOR-NEXT structures. (The following figure shows the nesting of 3 FOR-NEXT structures.)



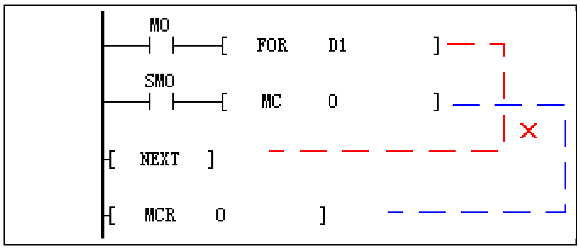
  3. You can use the conditional jump (CJ) instruction to jump out of the cycle body, so as to terminate the cycle body execution in advance, as shown in the LAD below:



4. It is prohibited to use the CJ statement to jump into the next cycle body, and the LAD below cannot be compiled correctly:



5. The intersection of the MC-MCR structure body and the FOR-NEXT structure body is prohibited, and the LAD below cannot be compiled correctly:



Note

The execution of the FOR-NEXT cycle is time-consuming. The more the cycle times are, or the more the instructions are contained in the cycle body, the longer time it takes. To prevent the operation timeout error, you need to use the WDT instruction within a time-consuming cycle body.

6.1.3 LBL: Jump label definition instruction

<table>
<tr><td colspan="8" rowspan="2">LAD:
</td><td colspan="3">Applicable model</td><td colspan="4">IVC1 IVC1S IVC3 IVC2L IVC1L</td></tr>
<tr><td colspan="3">Influenced flag bit</td><td colspan="4"></td></tr>
<tr><td colspan="8">IL: LBL (S)</td><td colspan="3">Step length</td><td colspan="4">3</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="12">Applicable soft element</td><td>Indexing</td></tr>
<tr><td>S</td><td>INT</td><td>Constant</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

- Operand description

  S:Labeling value

- Function description

  1. Defining a label whose labeling value is ***S***.

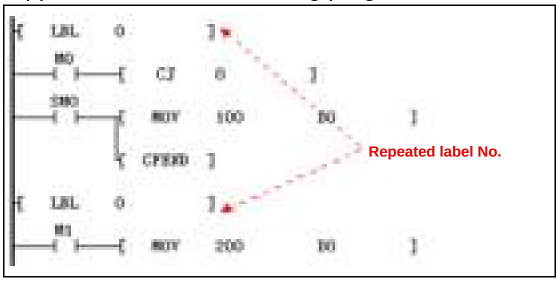
  2. Not used for actual operation. It only indicates the specific jumping position for the CJ instruction.

- Note

  1. Range of labeling value S: 0≤**S**≤127。

  2. Two labels with a repetitive definition are not allowed in one POU, otherwise the user program cannot be compiled. However, repetitive label definitionsare allowed in different POUs (for example, different subprograms).

- Application instance of wrong program



6.1.4 CJ: Conditional jump instruction

LAD:	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: CJ *(S)*	Step length	3

Operand	Type	Applicable soft element												Indexing
S	INT	Constant												

- Operand description

  S: Labeling value

- Function description

  1. When the energy flow is valid, the user program jumps to execute the instruction numbered ***S,*** a legal label.

  2. When the energy flow is invalid, there is no jump operation, and the system executes the instruction after CY in order.

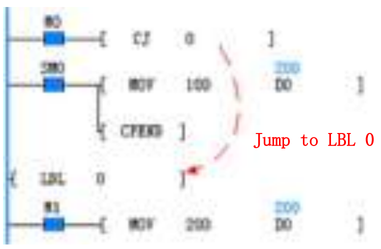
- Note

  1. The label ***S*** (0 ≤ S ≤ 127) to be jumped by the CJ instruction shall be a valid, and defined label, otherwise the user program cannot be compiled correctly.

  2. It is not allowed to use the CJ instruction to jump to a FOR-NEXT structure.

  3. It is allowable to use the CJ instruction to jump out of or jump into the MC-MCR structure or SFC state. However, such operation damages the logic of the MC-MCR structure and SFC state and makes the program complex. Thus, such operation is not recommended.

- Application instance



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

1. When the initial condition is M0=OFF and M1=ON, the CJ 0 instruction is not jumped, and D0=100. After executing the CFEND instruction, the program exits the main program in advance, instructions of LD M1 and MOV 200 D0 are not executed.

2. When M0=ON, M1=ON, the program executes the CJ 0 instruction, skips over the MOV 100 D0 and CFEND instructions. After jumping to the LBL 0 instruction, the program executes the MOV 200 D0 instruction, and now D0=200.

6.1.5 CFEND: Instruction for conditional return of user main program

LAD: [CFEND]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: CFEND	Step length	1

- Function description

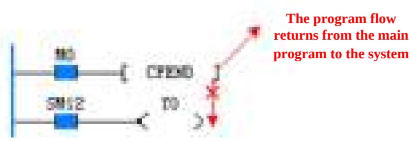
  1. When the energy flow of the instruction is valid, the main program returns to the system from the current scan cycle (the user main program is executed by the system repeatedly according to the scan cycle), and the instructionsin the subsequent main program are not executed.

  2. When the energy flow of the instruction is invalid, this instruction does not generate any action, and the subsequent instructions are executed in order.

- Note

  The CFEND instruction must be used in the user main program, otherwise the program cannot be compiled.

- Application instance



LD M0
CFEND
LD SM12
OUT Y0

When the program is running, if M0=OFF, the CFEND instruction does not generate any action, the subsequent LD SM12 and OUT Y0 instructions are executed, and you can see the Y0 blinks periodically.If M0=ON, CFEND instruction generates an action, the program flow returns to the system from the main program in advance, the subsequent LD SM12 and OUT Y0 instructions are not be executed, and the Y0 no longer blinks periodically.

6.1.6 WDT: Instruction for watchdog reset of user program

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
├──┤ ├──[WDT]	**Influenced flag bit**	
IL: **WDT**	**Step length**	**1**

- Function description

When the energy flow is valid, this instruction clears the timing value of the watchdog in the user program, and the watchdog of the system user program restarts the timing.

6.1.7 EI: Enable interrupt instruction

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
├──┤ ├──[EI]	**Influenced flag bit**	
IL: **EI**	**Step length**	**1**

- Function description

1. When the energy flow is valid, the interrupts in the current scan cycle are enabled.

2. When the EI instruction is valid, the interrupt request is allowed to be added into the interrupt request queue, waiting for the system response.

6.1.8 DI: Disable interrupt instruction

LAD: ├──┤ ├──[DI]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **DI**	**Step length**	**1**

- Function description

1. When the energy flow is valid, the global interrupt enable flag is invalid,that is, the global interrupt is disabled.

2. When the global interrupt enable flag is invalid, the interrupt events cannot generate any interrupt request.

- Note

When the interrupt disabling request instruction is valid, the system still responds to the unprocessed interrupt requests in the interrupt request queue, but the new interrupt events cannot generate the interrupt requests.

6.1.9 CIRET: Instruction for conditional return of user interrupt program

LAD: ├──┤ ├──[CIRET]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **CIRET**	**Step length**	**1**

- Function description

When the energy flow is valid, the system exits the current interrupt program in advance.

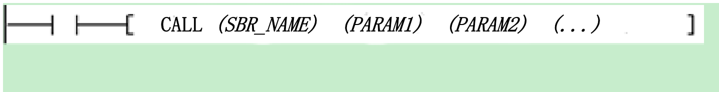
6.1.10 STOP: Instruction for stopping the user program

LAD: ├──┤ ├──[STOP]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **STOP**	**Step length**	**1**

- Function description

When the energy flow is valid, the system stops the execution of the user program immediately.

6.1.11 CALL: Instruction for calling the user subprogram

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **CALL (subprogram name) (subprogram parameter 1) (subprogram parameter 2) …**	**Step length**	**Determined by the parameters carried by the subprogram**

- Function description

  When the energy flow is valid, the system calls the designated subprogram for execution, and after execution, returns to the instructions following the CALL instruction to continue the execution.

- Note

  1. The subprogram called by the CALL instruction needs to be defined in the user program in advance. If an undefined subprogram occurs in the CALL instruction, the program cannot be compiled.

  2. The element type of the operands in the CALL instruction needs to match with the data type defined in the local variable table of the subprogram, otherwise the program cannot be compiled.

  The following application instance demonstrates some illegal matches.

  Application instance 1: In the local variable table of the SBR1 subprogram, the data type of operand 1 is DINT/DWORD.

  The following usages are illegal:

  - CALL SBR1 Z0 (The data type of Z element cannot be DINT/DWORD)
  - CALL SBR1 C199 (The data type of C0 to C199 element cannot be DINT/DOWRD)
  - CALL SBR1 K2X0 (Kn addressing 1≤n≤3, the data type cannot be DINT/DWORD)

  Application instance 2: In the local variable table of the SBR1 subprogram, the data type of operand 1 is INT/WORD.

  The following usages are illegal:

  - CALL SBR1 C200 (The data type of C200 to C255 elements cannot be INT/WORD)
  - CALL SBR1 K2X0 (Kn addressing 4≤n≤8, the data type cannot be INT/WORD)

  3. The element type of the operands in the CALL instruction needs to match with the data type defined in the local variable table of the subprogram, otherwise the program cannot be compiled.

  The following application instance demonstrates some illegal matches.

  Application instance: In the local variable table of the SBR1 subprogram, the data type of operand 1 is OUT or IN_OUT.

  The following usages are illegal:

  - CALL SBR1 321 (the constant cannot be changed, so it does not match with the OUT or IN_OUT type of the operand)
  - CALL SBR1 K4X0 (K4X0 is read-only, so it does not match with the OUT or IN_OUT type of the operand)
  - CALL SBR1 SD0 (SD0 is read only, so it does not match with the OUT or IN_OUT type of the operand)

  4. The number of the operands in the CALL instruction need to match with the local variable table of the subprogram, otherwise the program cannot be compiled.

6.1.12 CSRET: Instruction for conditional return of user subprogram

LAD: 	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **CSRET**	**Step length**	**1**

- Function description

  When the energy flow is valid, the system exits the current subprogram and return to the previous-level subprogram.

6.2 Data transmission instructions

6.2.1 MOV: Word data transmissioninstruction

LAD: ├─┤ ├─[MOV *(S)* *(D)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: MOV *(S)* *(D)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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
  Assigning the content of ***S*** to ***D*** and keeping the value of ***S*** when the energy flow is valid.
- Note
  1. MOV instruction supports two kinds of integers: one with symbol and the other without symbol. If the two operands of the instruction are both soft elements, then the data type is integer with symbol. If the source operand of the instruction is long integer with symbol, such as (－10, ＋100), then the destination operand is integer with symbol. If the source operand is long integer without symbol, such as (100, 45535), then the destination operand is also integer without symbol.
  2. The corresponding soft element C supports C0 to C199 only.
- Application instance

LD X0
MOV D0 D10

Assigning the content of D0 to D10 when X0=ON.In this case, D10=500.

6.2.2 DMOV: Double word data transmission instruction

LAD: ├─┤ ├─[DMOV *(S)* *(D)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: DMOV *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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
- Function description
  Assigning the content of ***S*** to ***D*** and keeping the value of ***S*** when the energy flow is valid.
- Note
  1. DMOV instruction supports two kinds of long integers: one with symbol and the other without symbol. If the two operands of the instruction are both soft elements, then the data type is integer with symbol. If the source operand of the instruction is long integer with symbol, such as (－10, ＋100), then the destination operand is also integer with symbol. If the source operand is long integer without symbol, such as (100, 45535), then the destination operand is also integer without symbol.
  2. The corresponding soft element C supports 32-bit C element only.
- Application instance

LD X0
DMOV D0 D10

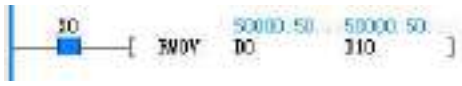
Assigning the content of (D0, D1) to (D10, D11) when X0=ON. In this case, (D10, D11)=50000.

6.2.3 RMOV: Floating-point data transmission instruction

LAD: RMOV (S) (D)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **RMOV** ***(S)*** ***(D)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  Assigning the content of ***S*** to ***D*** and keeping the value of ***S*** when the energy flow is valid.
- Application instance



LD X0
RMOV D0 D10

Assigning the content of (D0, D1) to (D10, D11) when X0=ON.In this case, (D10, D11)=50000.5.

6.2.4 BMOV: Block data transmission instruction

LAD: BMOV (S1) (D) (S2)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **BMOV** ***(S1)*** ***(D)*** ***(S2)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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
  S1: Source operand, start unit of data block
  D: Destination operand, start unit of data block
  S2: Size of the data block
- Function description
  Assigning the content of ***S2*** unitsstarting from ***S1*** to ***S2*** units starting from ***D*** and keeping the content of ***S2*** units starting from ***S1*** when the energy flow is valid.
- Application instance

LD X0
BMOV D0 D100 10

Assigning the content of 10 units starting from D0 to 10 units starting from D100 when X0=ON. In this case, D100=D0, D101=D1, ……, D109=D9

6.2.5 FMOV: Data block fill instruction

LAD: FMOV (S1) (D) (S2)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **FMOV** ***(S1)*** ***(D)*** ***(S2)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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
  S: Source operand, start unit of data block
  D: Destination operand, start unit of data block
  S2: Size of the data block
- Function description
  Fillingthe content of S1 in S2 units starting from D and keeping the content of S1 when the energy flow is valid.
- Note
  1. When ***S1, D***, and ***S2*** use C elements, the legal range is C0 to C199.
  2. ***S2*** is greater than 0.
  3. When ***S1*** and ***D*** are both Kn addressing, Kn needs to be equal.
- Application instance

LD X0
FMOV D0 D100 10

Filling the content of D0 in 10 units starting from D100 when X0=ON. In this case, D100=D101=……=D109=D0=500.

6.2.6 DFMOV: Data block double word fill instruction

LAD: DFMOV (S1) (D) (S2)	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: **DFMOV (S1) (D) (S2)**	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
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, start unit of data block
  S2: Size of the data block
- Function description
  Fillingthe content of ***S1*** in ***S2*** units starting from ***D*** and keeping the content of ***S1*** when the energy flow is valid.
- Note
  1. When ***S1, D***, and ***S2*** use C elements, only 32-bit C elements are supported.
  2. ***S2*** is greater than 0.
  3. When ***S1*** and ***D*** are both Kn addressing, Kn needs to be equal.
- Application instance

LD X0
DFMOV D0 D10 10

Filling the content of (D0,D1) in10×2 units starting from D10 when X0=ON. In this case, (D10,D11)= (D12,D13)=……= (D28,D29)= (D0,D1)=100000.

6.2.7 SWAP: MSB/LSB swop instruction

LAD: SWAP (D)	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: **SWAP (D)**	**Step length**	**3**

Operand	Type	Applicable soft element													Indexing
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description
  D: Destination operand, the word element whose most significant byte (MSB) and least significant byte (LSB) are swopped
- Function description
  Storing the value obtained after the MSB and LSB of the content of D are swopped into D when the energy flow is valid.
- Application instance

LD X0
SWAP D0

Storing the value obtained after the MSB and LSB of the content of D0=0x1027 (4135) are swapped into D0 when X0=ON. In this case, D0=0x2710 (10000).

6.2.8 XCH: Word swop instruction

LAD: XCH (D1) (D2)			
		Applicable model	**IVC1 IVC1S IVC3 IVC2L IVC1L**
		Influenced flag bit	
IL: **XCH**	***(D1)*** ***(D2)***	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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
  Exchanging the content of ***D1*** with the content of ***D2*** and then storing the obtained value into ***D1*** and ***D2*** units when the energy flow is valid.
- Note
  When using the Kn addressing mode, the Kn in *D1* and *D2* should be the same.
- Application instance

LD X0
XCH D0 D10

Exchanging the content of D0 with D10 when X0=ON. Before execution: D0=5000, D10=1000.

After execution: D0=1000, D10=5000.

6.2.9 DXCH: Double word swop instruction

LAD: DXCH (D1) (D2)			
		Applicable model	**IVC1 IVC1S IVC3 IVC2L IVC1L**
		Influenced flag bit	
IL: **DXCH**	***(D1)*** ***(D2)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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
  Exchanging the content of ***D1*** with the content of ***D2*** and then storing the obtained value into ***D1*** and ***D2*** units when the energy flow is valid.
- Note
  When using the Kn addressing mode, the Kn in *D1* and *D2* should be the same.
- Application instance

LD X0
DXCH D0 D10

Exchanging the content of (D0,D1)with(D10,D11) when X0=ON. Before execution: (D0,D1) =5000000, (D10,D11) =1000000. After execution: (D0,D1) =1000000, (D10,D11) =5000000.

6.2.10 PUSH: Data push instruction

LAD: PUSH (S1) (D) (S2)			
		Applicable model	**IVC1 IVC1S IVC3 IVC2L IVC1L**
		Influenced flag bit	
IL: **PUSH**	***(S1)*** ***(D)*** ***(S2)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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: Number of elements in the storage stack. An element No. indicates the position of a stack bottom.

S2: Size of the stack

- Function description

1. Pushing the value of ***S1*** into the stack top whose stack bottom is***D*** and increasing the value of ***D*** by 1 when the energy flow is valid. In this case, the No. of the stack top unitis No. of ***D***＋the value of ***D***.

2. Setting the operation carry flag bit (SM181) to 1, and not executing the push operation when the value of ***D*** is equal to the value of ***S2*** and there are still push instructions being executed.

- Note

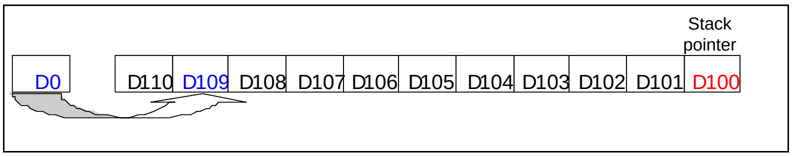
1. When the stack is defined to be illegal (for example, when the size of the stack is no more than 0, the number of elements in the stack is less than 0 or the size of the stack is beyond the limit), the system reportsan error, indicating the definition of the stack operated is invalid.

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

3. ***S2*** indicates the size of the stack and the range is greater than 0.

- Application instance

LD M0
PUSH D0 D100 10



1. Pushing the content of D0 into the stack whose stack bottom is D100 when M0=ON.

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

LAD: ├─┤ ├─[FIFO *(D1)* *(D2)* *(S)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: FIFO *(D1)* *(D2)* *(S)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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: Number of elements in the stack, and the initial element of the stack is its element number plus 1.

D2: Storage register for popped value

S: Size of the queue

- Function description

1. Assigning the initial value of the word stack starting from ***D1***(the content of the unit after ***D1***) to ***D2***, subtracting the value of ***D1*** by 1, moving the content of ***S*** units after ***D1*** forward and filling 0 in the last unit when the energy flow is valid.

2. Setting the zero flag bit (SM180) to 1 when D1 is 0, that is, the stack is empty.

- Note

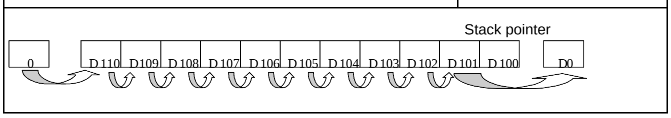
1. When the stack is defined to be illegal (for example, when the size of the stack is no more than 0, the number of elements in the stack is less than 0 or the size of the stack is beyond the limit), the system reports an error, indicating the definition of the stack operated is invalid.

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

3. ***S*** indicates the size of the stack and the range is greater than 0.

- Application instance

LD M0
FIFO D100 D0 10



1. Filling the content of D101in D0,movingthe content of D101 to D110forward, and filling 0 in D110 when M0=ON.

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

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

6.2.12 LIFO: Last-in-first-out instruction

LAD: LIFO *(D1)* *(D2)* *(S)*	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: **LIFO** ***(D1)*** ***(D2)*** ***(S)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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: Number of elements in the queue, and the initial element of the stack is its element number plus 1

  D2: Storage register for popped value

  S: Size of the queue

- Function description

  1. Assigning the content of the stack top with ***D1*** as the stack bottom to ***D2*** and subtracting the value of ***D1*** by 1 when the energy flow is valid.

  2. Setting the zero flag bit (SM180) to 1 when ***D1*** is 0, that is, the stack is empty.

- Note

  1. When the stack is defined to be illegal (for example, when the size of the stack is no more than 0, the number of elements in the stack is less than 0 or the size of the stack is beyond the limit), the system reportsan error, indicating the definition of the stack operated is invalid

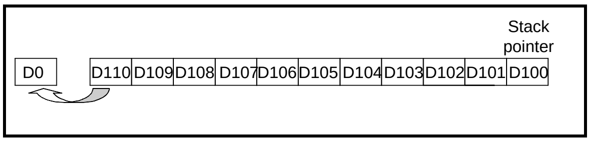
  2. The size of the stack does not include the stack bottom element(the element designated by ***D1***).

  3. ***S*** indicates the size of the stack and the range is greater than 0.

- Application instance

LD M0

LIFO D100 D0 10



1. Assigning the content of D110 to D0 and keeping the content of D101 to D110 when M0=ON.

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: Word string shift right instruction

LAD: WSFR *(S1)* *(D)* *(S2)* *(S3)*	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	**Carry flag, borrow flag**
IL: **WSFR** ***(S1)*** ***(D)*** ***(S2)*** ***(S3)***	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
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, start element of a word string

  S2: Size of destination word queue

  S3: Number of words filled when shifting a word string rightward

- Function description

  Shifting the content of ***S2*** units starting from *D*rightward by ***S3*** units in the unit of word, discarding the ***S3*** data on the rightmost side, and shifting the content of ***S3*** unitsstarting from ***S1*** to the left end of the word string when the energy flow is valid.

- Note

  1. Left and right order. The elements with small numbers are at the right, and the elements with large numbers are at the left.

2.Both of ***S2*** and ***S3*** is no less than 0.

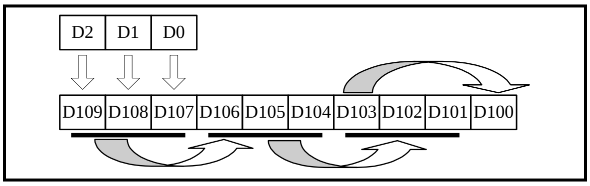
3. ***S2*** is no less than ***S3***.

4. When ***S1*** and ***D*** both use the Kn addressing, Kn needs to be the same.

- Application instance

LD X0

WSFR D0 D100 10 3



1. Shifting the content of 10 units starting from D100 rightward by three units in the unit of word, discarding the data of D102 to D100 units on the rightmost side, and shifting the content of the three units starting from D0 to the left end of the word string when M0=ON.

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: keeping the content of D0 to D2. 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: Word string shift left instruction

LAD: WSFL (S1) (D) (S2) (S3)	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: **WSFL** ***(S1) (D) (S2) (S3)***	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
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, start element of a word string

  S2: Size of destination word queue

  S3: Number of words filled when shifting a word string rightward

- Function description

  Shifting the content of ***S2*** units starting from *D* leftward by ***S3*** units in the unit of word, discarding the ***S3*** data on the leftmost side, and shifting the content of ***S3*** units starting from ***S1*** to the right end of the word string when the energy flow is valid.

- Note

  1. Left and right order. The elements with small numbers are at the right, and the elements with large numbers are at the left.

  2.Both of ***S2*** and ***S3*** is no less than 0.

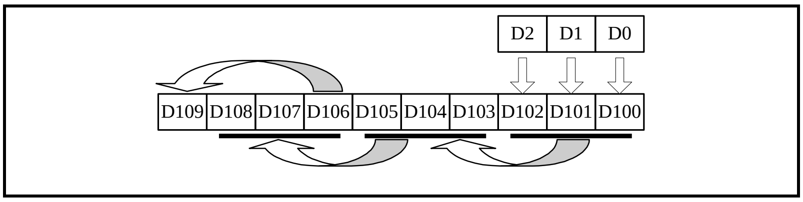
  3. ***S2*** is no less than ***S3***.

  4. When ***S1*** and ***D*** both use the Kn addressing, Kn needs to be the same.

- Application instance

LD X0

WSFL D0 D100 10 3



1. Shifting the content of 10 units starting from D100 leftward by three units in the unit of word, discarding the data of D109 to D107 units on the leftmost side, and shifting the content of the three units starting from D0 to the right end of the word string when X0=ON.

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: Keeping the content of D0 to D2. 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 operation instructions

6.3.1 ADD: Integer addition instruction

LAD: ADD *(S1)* *(S2)* *(D)*	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: **ADD *(S1)* *(S2)* *(D)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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. Increasing ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 32767. Setting the zero flag bit (SM180) when the operation result is 0. Setting the borrow flag bit (SM182) when the operation result is less than －32768.

- Application instance

LD X0

ADD D0 D1 D10

Increasing D0 (1000)by D1 (2000), and assigning the operation result toD10 when X0=ON. In this case, D10=3000.

6.3.2 SUB: Integer subtraction instruction

LAD: SUB (S1) (S2) (D)	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: SUB *(S1) (S2) (D)*	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
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. Subtracting ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.
  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 32767. Setting the zero flag bit (SM180) when the operation result is 0. Setting the borrow flag bit (SM182) when the operation result is less than －32768.
- Application instance

LD X0
SUB D0 D1 D10

SubtractingD0 (1000)by D1 (2000), and assigning the operation result toD10 when X0=ON. In this case, D10=－1000.

6.3.3 MUL: Integer multiplication instruction

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: MUL *(S1) (S2) (D)*	**Step length**	**8**

Operand	Type	Applicable soft element													Indexing
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
  Multiplying ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.
- Note
  The operation result of MUL instruction is a 32-bit data.
- Application instance

LD X0
MUL D0 D1 D10

Multiplying D0 (1000) by D1 (2000), and assigning the operation result to (D10,D11) when X0=ON. In this case, (D10,D11)=2000000.

6.3.4 DIV: Integer division instruction

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	

IL: DIV (S1) (S2) (D)										**Step length**		**7**				
Operand	Type	Applicable soft element														Indexing
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
  Dividing ***S1*** by ***S2*** and assigning the operation result to ***D*** (D includes two units, one stores the quotient value and the other stores the residue value) when the energy flow is valid.
- Note
  If ***S2***≠0, the system reportsan error, indicating that 0 cannot be used as a divisor and does not execute the division operation.
- Application instance

LD X0
DIV D0 D1 D10
Dividing D0 (2500) by D1 (1000), and assigning the operation result to (D10,D11) when X0=ON. In this case, D10=2, D11=500.

6.3.5 SQT: Instruction for extracting the square root of an integer

LAD: ├─┤ ├─[SQT (S) (D)]										**Applicable model**		**IVC1 IVC1S IVC3 IVC2L IVC1L**				
										Influenced flag bit		**Zero flag, carry flag, and borrow flag**				
IL: SQT (S) (D)										**Step length**		**5**				
Operand	Type	Applicable soft element														Indexing
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. Extracting the square root of ***S*** and assigning the operation result to ***D*** when the energy flow is valid.
  2. Setting the zero flag bit (SM180) When (***D***) is 0. Setting the carry flag bit (SM182) when the operation result is rounded off.
- Note
  If ***S*** ≥ 0, the system reports an operand error and does not execute the square root extraction.
- Application instance

LD X0
SQT D0 D10
Assigning the square root result of D0 (1000) to D10 when X0=ON. In this case, D10=31.

6.3.6 INC: Integer plus one instruction

LAD: ├─┤ ├─[INC (D)]										**Applicable model**		**IVC1 IVC1S IVC3 IVC2L IVC1L**				
										Influenced flag bit						
IL: INC (D)										**Step length**		**3**				
Operand	Type	Applicable soft element														Indexing
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description
  D: Destination operand

- Function description
  Increasing ***D*** by 1 automatically when the energy flow is valid.

- Note
  This instruction is a cyclic addition instruction, and the range is from -32768 to 32767. The supported range of C element is from C0 to C199.

- Application instance

LD X0
INC D0

Increasing D0 (1000) by 1 automatically when X0=ON. After execution, D0=1001.

6.3.7 DEC: Integer minus one instruction

LAD: DEC *(D)*								**Applicable model**		**IVC1 IVC1S IVC3 IVC2L IVC1L**					
								Influenced flag bit							
IL: DEC *(D)*								**Step length**		**3**					
Operand	Type	Applicable soft element													Indexing
D	INT			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description
  D: Destination operand

- Function description
  Decreasing D by 1 automaticallyWhen the energy flow is valid.

- Note
  This instruction is cyclic minus instruction, and the range is from -32768 to 32767.

- Application instance

LD X0
DEC D0

DecreasingD0 (1000) by 1 automatically when X0=ON. After execution, D0=999.

6.3.8 VABS: Instruction for obtaining the absolute value of an integer

LAD: VABS (S) (D)								**Applicable model**		**IVC1 IVC1S IVC3 IVC2L IVC1L**						
								Influenced flag bit								
IL: VABS *(S) (D)*								**Step length**		**5**						
Operand	Type	Applicable soft element														Indexing
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
  Taking an absolute value of S and assigning the obtained result to ***D*** when the energy flow is valid.

- Note
  The range of ***S*** needs to be from -32767 to 32767. When the value of ***S*** is -32768, the system reportsan error, indicating the value of the instruction operand is invalid, and the instruction does not generate any action.

- Application instance

LD X0
VABS D0 D10

Taking an absolute value of D0 (-1000), assigning the obtained result to D10 when X0=ON. In this case, D10=1000.

6.3.9 NEG: Integer negation instruction

LAD: NEG *(S)* *(D)*	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	
IL: NEG *(S) (D)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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
  Taking the negative of ***S*** and assigning the obtained result to ***D*** when the energy flow is valid.
- Note

The range of ***S*** needs to be from -32767 to 32767. When the value of ***S*** is -32768, the system reports an error, indicating the value of the instruction operand is invalid, and the instruction does not generate any action.

- Application instance

LD X0
NEG D0 D10

Taking the negative of D0 (1000), and assigning the obtained result to D10 when X0=ON. In this case, D10=-1000.

6.3.10 DADD: Long integer addition instruction

LAD: ─┤ ├──[DADD (S1) (S2) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DADD ***(S1) (S2) (D)***	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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. Increasing ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.
  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 2147483647. Setting the zero flag bit (SM180) when the operation result is 0. Setting the borrow flag bit (SM182) when the operation result is less than －2147483648.
- Application instance

LD X0
DADD D0 D2 D10

Increasingthe value (100000) of (D0,D1) bythe value (200000) of (D2,D3),and assigning the operation result to (D10,D11)when X0=ON. In this case, (D10,D11)=300000.

6.3.11 DSUB: Long integer subtraction instruction

LAD: ─┤ ├──[DSUB (S1) (S2) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DSUB ***(S1) (S2) (D)***	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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. Subtracting ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.

2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 2147483647. Setting the zero flag bit (SM180) when the operation result is 0. Setting the borrow flag bit (SM182) when the operation result is less than －2147483648.

- Application instance

LD X0

DSUB D0 D2 D10

Subtractingthe value (100000) of (D0,D1) bythe value (200000) of (D2,D3), and assigning the operation result to (D10,D11)when X0=ON. In this case, (D10,D11)=－100000.

6.3.12 DMUL: Long integer multiplication instruction

LAD: ├─┤ ├─[DMUL (S1) (S2) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: DMUL *(S1) (S2) (D)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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

Multiplying ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.

- Note

Please note that the operation result of DMUL instruction is a 32-bit data, which may cause an outflow.

- Application instance

LD X0

DMUL D0 D2 D10

Multiplying the value (83000) of (D0,D1) by the value (2000) of (D2,D3), and assigning the operation result to (D10,D11)when X0=ON. In this case, (D10,D11)=166000000.

6.3.13 DDIV: Long integer division instruction

LAD: ├─┤ ├─[DDIV (S1) (S2) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: DDIV *(S1) (S2) (D)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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

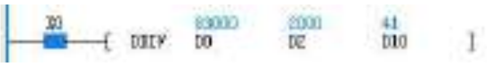
S2: Source operand 2

D: Destination operand

- Function description

Dividing***S1*** by ***S2*** and assigning the operation result to ***D*** (***D*** includes four units, the first two store the quotient value and the last two store the residue value) when the energy flow is valid.

- Note
  If *S2*≠0, the system reports an error, indicating that 0 cannot be used as a divisor and does not execute the division operation.
- Application instance



LD X0
DDIV D0 D2 D10

Dividing the value (83000) of (D0,D1) by the value (2000) of (D2,D3), and assigning the operation result to (D10,D11) and (D12,D13)when X0=ON. In this case, (D10,D11)=41 and (D12,D13)=1000.

6.3.14 DSQT: Instruction for extracting the square root of a long integer

LAD: ├─┤ ├─[DSQT (S) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: DSQT (S) (D)	Step length	7

Operand	Type	Applicable soft element													Indexing
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. Extracting the square root of ***S*** and assigning the operation result to ***D*** when the energy flow is valid.
  2. Setting the zero flag bit (SM180) When (***D***) is 0. Setting the carry flag bit (SM182) when the operation result is rounded off.
- Note
  If ***S*** ≥ 0, the system reports an operand error and does not execute the square root extraction.
- Application instance

LD X0
DSQT D0 D10

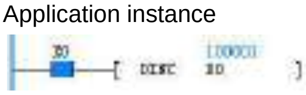
Extracting the square root of the value (83000) of (D0,D1), and assigning the square root result to (D10,D11) when X0=ON. In this case, (D10,D11)=288.

6.3.15 DINC: Long integer plus one instruction

LAD: ├─┤ ├─[DINC (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	
IL: DINC (D)	Step length	4

Operand	Type	Applicable soft element													Indexing
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description
  D: Destination operand
- Function description
  Increasing ***D*** by 1 automatically when the energy flow is valid.
- Note
  1. This instruction is a cyclic addition instruction, and the range is from -2147483648 to 2147483647.
  2. Supporting 32-bit C element only among C elements.
- Application instance



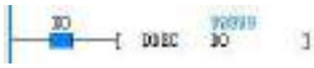
LD X0
DINC D0

Increasing the value (100000) of (D0,D1)automatically when X0=ON. After execution,(D0,D1)=100001.

6.3.16 DDEC: Long integer minus one instruction

LAD: DDEC *(D)*									**Applicable model**			**IVC1 IVC1S IVC3 IVC2L IVC1L**				
									Influenced flag bit							
IL: **DDEC** ***(D)***									**Step length**			**4**				
Operand	Type	Applicable soft element														Indexing
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description
  D: Destination operand
- Function description
  Decreasing ***D*** by 1 automatically When the energy flow is valid.
- Note
  This instruction is cyclic minus instruction, and the range is from -2147483648 to 2147483647.
- Application instance



LD X0
DDEC D0

Decreasing the value (100000) of (D0,D1) automatically when X0=ON. After execution, (D0,D1)=99999.

6.3.17 DVABS: Instruction for obtaining the absolute value of a long integer

LAD: DVABS *(S)* *(D)*									**Applicable model**			**IVC1 IVC1S IVC3 IVC2L IVC1L**				
									Influenced flag bit			**Zero flag, carry flag, and borrow flag**				
IL: **DVABS** ***(S)*** ***(D)***									**Step length**			**7**				
Operand	Type	Applicable soft element														Indexing
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
  Taking an absolute value of S and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  The range of ***S*** needs to be from -2147483647 to 2147483647. When the value of ***S*** is -2147483648, the system an error, indicating the value of the instruction operand is invalid, andthe instruction does not generate any action.
- Application instance

LD X0
DVABS D0 D10

Taking an absolute value (—100000) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11)=100000.

6.3.18 DNEG: Long integer negation instruction

LAD: DNEG *(S)* *(D)*									**Applicable model**			**IVC1 IVC1S IVC3 IVC2L IVC1L**				
									Influenced flag bit			**Zero flag, carry flag, and borrow flag**				
IL: **DNEG** ***(S)*** ***(D)***									**Step length**			**7**				
Operand	Type	Applicable soft element														Indexing
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

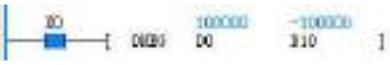
- Function description

Taking the negative of ***S*** and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

The range of ***S*** needs to be from －2147483647 to 2147483647. When the value of ***S*** is －2147483648, the system an error, indicating the value of the instruction operand is invalid, and the instruction does not generate any action.

- Application instance



LD X0
DNEG D0 D10

Taking the negative of the value (100000) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11)＝－100000.

6.3.19 SUM: Integeraccumulation instruction

LAD: ├──┤ ├──[SUM *(S1)* *(S2)* *(D)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: **SUM *(S1)* *(S2)* *(D)***	**Step length**	**8**	

Operand	Type	Applicable soft element														Indexing
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, start unit for accumulation

S2: Source operand, number of pieces of data to be accumulated

D: Destination operand, accumulation result

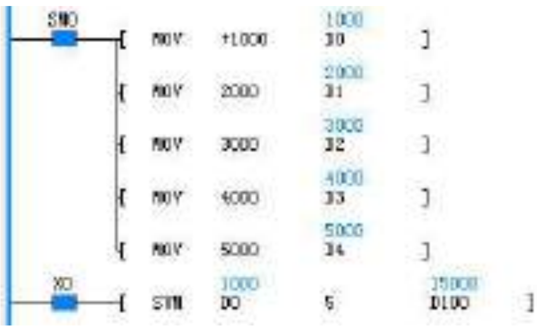
- Function description

Accumulating the content of ***S2*** units starting from the start unit (***S1***), and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

1. The operation result of the SUM instructionis 32-bit data.
2. If 0≤***S2***≤255, the system reports an operand error.
3. Since ***D*** is a 32-bit data, the carry and borrow flags are always 0, while the zero flag bit is determined by the final accumulation result.

- Application instance



LD SM0
MOV 1000 D0
MOV 2000 D1
MOV 3000 D2
MOV 4000 D3
MOV 5000 D4
LD X0
SUM D0 5 D100

Accumulating the data of 5 units starting from D0, and assigning the obtained result to (D100,D101) when X0=ON.In this case, (D100,D101)=D0＋……＋D4=15000.

6.3.20 DSUM: Long integer accumulation instruction

LAD: ├─┤ ├─[DSUM (S1) (S2) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DSUM *(S1)* *(S2)* *(D)*	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
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, start unit for accumulation

  S2: Source operand, number of pieces of data to be accumulated

  D:Destination operand, accumulation result

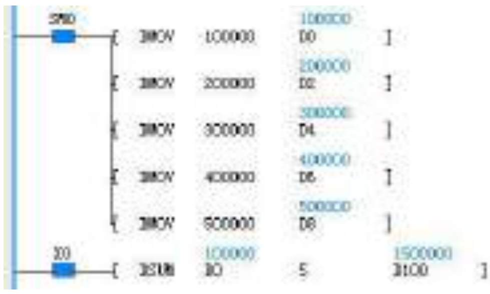
- Function description

  Accumulating the content of ***S2***×2 units starting from the start unit (***S1***), performing the DSUM instruction, and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

  If 0≤***S2***≤255, the system reports an operand error.

- Application instance



```
LD   SM0
DMOV   100000   D0
DMOV   200000   D2
DMOV   300000   D4
DMOV   400000   D6
DMOV   500000   D8
LD   X0
DSUM   D0   5   D100
```

Accumulating the data of 5×2 units starting from D0, and assigning the obtained result to (D100,D101) when X0=ON.

In this case, (D100,D101)= (D0,D1)+……+ (D8,D9)=1500000.

6.4 Floating-point arithmetic operation instructions

6.4.1 RADD: Floating-point numberaddition instruction

LAD: ├─┤ ├─[RADD (S1) (S2) (D)]	Applicable model	IVC1 IVC1S IVC3 IVC2L IVC1L
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: RADD *(S1)* *(S2)* *(D)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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. Increasing ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 1.701412e+038 or less than −1.701412e+038. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance

```
LD   X0
RADD D0 D2 D10
```

Increasingthe value (-10000.2) of (D0,D1) bythe value (2000.5) of (D2,D3),and assigning the operation result to (D10,D11)when X0=ON. In this case, (D10,D11)=-7999.7.

6.4.2 RSUB: Floating-point numbersubtraction instruction

LAD:		
├──┤ ├──[RSUB (S1) (S2) (D)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: RSUB ***(S1) (S2) (D)***	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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. Subtracting ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 1.701412e＋038 or less than －1.701412e＋038. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance

LD X0

RSUB D0 D2 D10

Subtractingthe value (－10000.2) of (D0,D1) bythe value (2000.5) of (D2,D3), and assigning the operation result to (D10,D11)when X0=ON. In this case, (D10,D11)=－12000.7.

6.4.3 RMUL: Floating-point numbermultiplication instruction

LAD:		
├──┤ ├──[RMUL (S1) (S2) (D)]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: RMUL ***(S1) (S2) (D)***	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
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. Multiplying ***S1*** by ***S2*** and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (D) is larger than 1.701412e＋038 or less than －1.701412e＋038. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance

LD X0

RMUL D0 D2 D10

Multiplying the value (－10000.2) of (D0,D1) by the value (2000.5)of (D2,D3), and assigning the operation result to (D10,D11)when X0=ON. In this case, (D10,D11)=-20005400.0 (actually the result is -20005400.1, but is rounded off according to the calculation precision).

6.4.4 RDIV: Floating-point numberdivision instruction

LAD: ├─┤ ├─[RDIV *(S1)* *(S2)* *(D)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: **RDIV *(S1)* *(S2)* *(D)***	**Step length**	**10**	

Operand	Type	Applicable soft element														Indexing
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. Dividing ***S1*** by ***S2*** and assigning the operation result to ***D***.
  2. Setting the carry flag bit (SM181) when the operation result (D) is larger than 1.701412e＋038 or less than －1.701412e＋038. Setting the zero flag bit (SM180) when the operation result is 0.
- Note
  If *S2*≠0, the system reportsan error, indicating that 0 cannot be used as a divisor and does not execute the division operation.
- Application instance

LD X0
RDIV D0 D2 D10

Dividing the value (-10000.2) of (D0,D1) by the value (2000.5) of (D2,D3), and assigning the operation result to (D10,D11)when X0=ON.In this case, (D10,D11)=-4.998850.

6.4.5 RSQT: Instruction for extracting the square root of a floating-point number

LAD: ├─┤ ├─[RSQT *(S)* *(D)*]	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: **RSQT *(S)* *(D)***	**Step length**	**7**	

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  1. Extracting the square root of ***S*** and assigning the operation result to ***D*** when the energy flow is valid.
  2. Setting the zero flag bit (SM180) When (***D***) is 0.
- Note
  If ***S***≥0, the system reports an operand error and does not execute square root extraction.
- Application instance

LD X0
RSQT D0 D10

Extracting the square root of the value (10000.2)of (D0,D1), and assigning the square root result to (D10,D11)when X0=ON. In this case, (D10,D11)=100.000999.

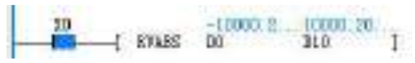
6.4.6 RVABS: Instruction for obtaining the absolute value of a floating-point number

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**

├──┤ ├──[RVABS (S) (D)]	**Influenced flag bit**	
IL: RVABS *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  Taking an absolute value of ***S*** and assigning the obtained result to ***D*** when the energy flow is valid.
- Application instance



LD X0

RVABS D0 D10

Taking an absolute value (-10000.2) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11)=10000.2.

6.4.7 RNEG: Floating-point number negation instruction

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
├──┤ ├──[RNEG (S) (D)]	**Influenced flag bit**	
IL: RNEG *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  Taking the negative of ***S*** and assigning the obtained result to ***D*** when the energy flow is valid.
- Application instance

LD X0

RNEG D0 D10

Taking the negative of the value (10000.2) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11)＝-10000.2.

6.4.8 SIN: Instruction for obtaining SIN of a floating-point number

LAD:	**Applicable model**	**IVC1 IVC1S IVC3 IVC2L IVC1L**
├──┤ ├──[SIN (S) (D)]	**Influenced flag bit**	**Zero flag, carry flag, and borrow flag**
IL: SIN *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  1. Obtaining the SIN value of ***S*** (unit: radian), and assigning the obtained result to ***D*** when the energy flow is valid.
  2. Setting the zero flag bit (SM180) when the operation result (***D***) is 0.
- Application instance

LD X0

SIN D0 D10

Taking the SIN value (1.57) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11) =1.

6.4.9 COS: Instruction for obtaining COS of a floating-point number

LAD: |—| |—[COS *(S)* *(D)*]	**Applicable model**	**IVC1 IVC3 IVC2L IVC1L IVC1S**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: COS *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant						D				V		R	√
D	REAL							D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  1. Obtaining the COS value of ***S*** (unit: radian), and assigning the obtained result to ***D*** when the energy flow is valid.
  2. Setting the zero flag bit (SM180) when the operation result (***D***) is 0.
- Application instance

LD X0
COS D0 D10

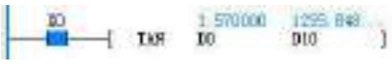
Taking the COS value (3.14) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11) =-0.999999.

6.4.10 TAN: Instruction for obtaining TAN of a floating-point number

LAD: |—| |—[TAN *(S)* *(D)*]	**Applicable model**	**IVC1 IVC3 IVC2L IVC1L IVC1S**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: TAN *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant						D				V		R	√
D	REAL							D				V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  1. Obtaining the TAN value of ***S*** (unit: radian), and assigning the obtained result to ***D*** when the energy flow is valid.
  2. Setting the carry flag bit (SM180) when the operation result (***D***) is larger than 1.701412e+038 or less than -1.701412e+038. Setting the zero flag bit (SM180) when the operation result is 0.
- Application instance



LD X0
TAN D0 D10

Taking the TAN value (1.57) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11) =1255.848398.

6.4.11 POWER: Instruction for exponentiation of a floating-point number

LAD: |—| |—[POWER *(S1)* *(S2)* *(D)*]	**Applicable model**	**IVC1 IVC3 IVC2L IVC1L IVC1S**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: POWER *(S1)* *(S2)* *(D)*	**Step length**	**10**

Operand	Type	Applicable soft element	Indexing

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. Obtaining the ***S2*** power of ***S1***, and then assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 1.701412e＋038 or less than -1.701412e＋038. Setting the zero flag bit (SM181) when the operation result is 0.

- Note

  1. If ***S1***=0 and ***S2***≤0, the system reportsan operand value error and does not execute the operation.

  2. If ***S1***＜0 and the mantissa of ***S2*** is not 0, the system reports an operand value error and does not executes the operation.

- Application instance

LD X0

POWER D0 D2 D10

Obtaining the (D2,D3) power of (D0,D1), namely (the 3.0 power of 55.0), and assigning the result to (D10,D11) when X0=ON. In this case, (D10,D11) =166375.0.

6.4.12 LN: Instruction for obtaining the natural logarithm of a floating-point number

LAD: ├──┤ ├──[LN (S) (D)]	**Applicable model**	**IVC1 IVC3 IVC2L IVC1L IVC1S**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: LN *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description

  S: Source operand

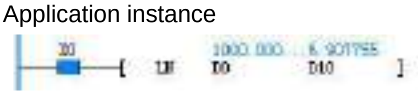
  D: Destination operand

- Function description

  1. Obtaining the LN value of ***S***, and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 1.701412e＋038 or less than -1.701412e＋038. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance



LD X0

LN D0 D10

Taking the LN value (1000.0) of (D0,D1),and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11) =6.907755.

6.4.13 EXP: Instruction for obtaining the natural number power of a floating-point number

LAD: ├──┤ ├──[EXP (S) (D)]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L IVC1S**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: EXP *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	REAL								D				V		R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

  1. Obtaining the EXP value of ***S***, and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the carry flag bit (SM181) when the operation result (***D***) is larger than 1.701412e＋038 or less than -1.701412e＋038. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance


```
LD   X0
EXP  D0  D10
```

Taking the EXP value (10.0) of (D0,D1), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11) =22026.464844.

6.4.14 RSUM: Floating-point number accumulation instruction

LAD: ├─┤ ├─[RSUM *(S1)* *(S2)* *(D)*]															**Applicable model**	**IVC2L IVC1 IVC3 IVC1L IVC1S**
															Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: **RSUM** ***(S1)*** ***(S2)*** ***(D)***															**Step length**	**9**
Operand	Type	Applicable soft element													Indexing	
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, start unit for accumulation

  S2:Source operand, number of pieces of data to be accumulated

  D:Destination operand,accumulation result

- Function description

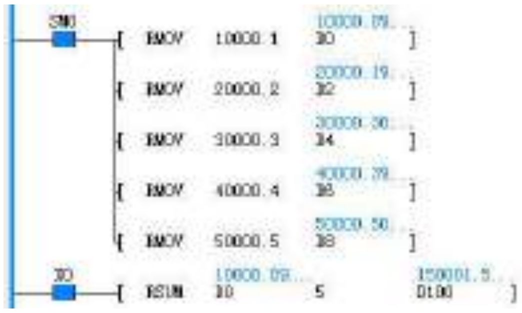
  Accumulating the content of ***S2***×2 units starting from the start unit (***S1***), performing the RSUM instruction, and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

  1. If 0≤***S2***≤255, the system reports an operand error.

  2. The system does not perform the accumulation operation once an overflow occurs.

- Application instance




```
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
```

Accumulating the data of 5×2 units starting from D0 , and assigning the obtained result to (D100,D101) when X0=ON.

In this case, (D100,D101)= (D0,D1)＋……＋ (D8,D9)=150001.5.

6.4.15 ASIN: Instruction for obtaining ASIN of a floating-point number

LAD: ├─┤ ├─[ASIN *(S)* *(D)*]													**Applicable model**	**IVC3**
													Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ASIN (*S*) (*D*)													**Step length**	**7**
Operand	Type	Applicable soft element											Indexing	
S	REAL	Constant						D				V	R	√

D	REAL								D				V	R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

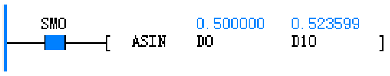
  1. Obtaining the SIN^{-1} value of ***S***, and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the zero flag bit (SM180) when the operation result (***D***) is 0.

- Note

  If S>1 or S<-1, the system reports an operand error, does not perform the conversion operation and keeps the content of ***D***.

- Application instance



LD SM0
ASIN D0 D10

Taking the SIN^{-1} value (0.500000) of (D0,D1),and assigning the obtained result to (D10,D11) when SM0=ON. In this case, (D10,D11) =0.523599.

6.4.16 ACOS: Instruction for obtaining ACOS of a floating-point number

LAD: ACOS (S) (D)	Applicable model	IVC3
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ACOS (*S*) (*D*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

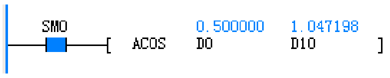
  1. Obtaining the COS^{-1} value of ***S***, and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the zero flag bit (SM180) when the operation result (***D***) is 0.

- Note

  If S>1 or S<-1, the system reports an operand error, does not perform the conversion operation and keeps the content of ***D***.

- Application instance



LD SM0
ACOS D0 D10

Taking the COS^{-1} value (0.500000) of (D0,D1), and assigning the obtained result to (D10,D11) when SM0=ON. In this case, (D10,D11) =1.047198.

6.4.17 ATAN: Instruction for obtaining ATAN of a floating-point number

LAD: ATAN (S) (D)	Applicable model	IVC3
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ATAN (*S*) (*D*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- Operand description

  S: Source operand

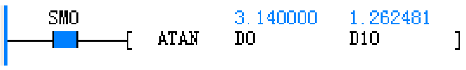
  D: Destination operand

- Function description

  1. Obtaining the TAN^{-1} value of ***S***, and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the zero flag bit (SM180) when the operation result (***D***) is 0.

- Application instance



LD SM0
ATAN D0 D10

Taking the TAN^{-1} value (3.14)of (D0,D1), and assigning the obtained result to (D10,D11) when SM0=ON. In this case, (D10,D11) =1.262481.

6.4.18 LOG: Instruction for obtaining the common logarithm of a floating-point number

LAD: ├─┤ ├─[LOG (S) (D)]	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: LOG (S) (D)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- Operand description

  S: Source operand

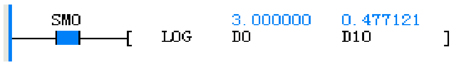
  D: Destination operand

- Function description

  1. Obtaining the LOG value of S, and assigning the operation result to D when the energy flow is valid. LOG is a common logarithm operation based on 10.

  2. Setting the carry (overflow) flag bit (SM181) when the operation result (D) overflows. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance



LD SM0

LOG D0 D10

Taking the value (3.0)of D0(D1), and assigning the obtained result to D10(D11)when SM0=ON. In this case, D10(D11) =0.477121.

6.4.19 RAD: Instruction for floating-point number angle-radian conversion

LAD: ├─┤ ├─[RAD (S) (D)]	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: RAD (S) (D)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- Operand description

  S: Source operand

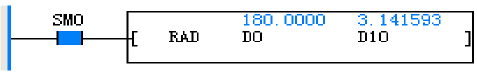
  D: Destination operand

- Function description

  1. Converting the angle value of ***S*** unit floating-point number into a radian value, and assigning the operation result to ***D*** when the energy flow is valid.

  2. Setting the zero flag bit (SM180) when the operation result is 0.

- Application instance



LD SM0

RAD D0 D10

Taking the value (180.0)of D0(D1), and assigning the obtained result to D10(D11)when SM0=ON. In this case, D10(D11) =3.141593.

6.4.20 DEG: Instruction for floating-point number radian-angle conversion

LAD: ├─┤ ├─[DEG (S) (D)]	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DEG (S) (D)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S	REAL	Constant							D				V	R	√
D	REAL								D				V	R	√

- Operand description

  S: Source operand

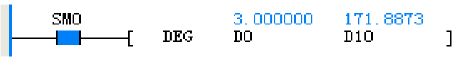
  D: Destination operand

- Function description

  1. Converting the radian value of ***S*** unit floating-point number into a angle value, and assigning the operation result to ***D*** when the energy flow is valid.

2. Setting the zero flag bit (SM180) when the operation result is 0.Setting the carry (overflow) flag bit (SM181) when the operation result (D) overflows.

- Application instance



LD SM0
DEG D0 D10

Taking the value (3.0)of D0(D1), and assigning the obtained result to D10(D11)when SM0=ON. In this case, D10(D11) =171.8873.

6.5 Value conversion instructions

6.5.1 DTI: Instruction for converting a long integer to an integer

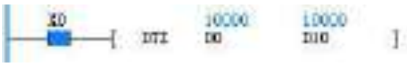
LAD: DTI (S) (D)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: DTI (S) (D)	Step length	6

Operand	Type	Applicable soft element														Indexing
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
  Converting ***S*** from a long integer to an integer, and assigning the operation result to ***D*** when the energy flow is valid.
- Note

If ***S***>32767 or ***S***<-32768, the system reports an operand error, does not execute the conversion operation and keeps the content of ***D***.

- Application instance



LD X0
DTI D0 D10

Converting (D0, D1) =10000 from a long integer to an integer, and assigning the operation result to D10 when X0=ON. In this case, D10=10000.

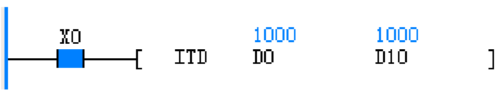
6.5.2 ITD: Instruction for converting an integer to a long integer

LAD: ITD (S) (D)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: ITD (S) (D)	Step length	6

Operand	Type	Applicable soft element														Indexing
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
  Converting ***S*** from an integer to a long integer, and assigning the operation result to ***D*** when the energy flow is valid.

- Application instance



LD X0
ITD D0 D10

Converting D0=1000 from an integer to a long integer, and assigning the operation result to D10 when X0=ON. In this case, (D10,D11)=1000.

6.5.3 FLT: Instruction for converting an integer to a floating-point number

LAD:	Applicable model	IVC2L IVC1 IVC3 IVC1L

├─┤ ├─[FLT *(S)* *(D)*]																**Influenced flag bit**	**Zero flag, carry flag, and borrow flag**
IL: FLT ***(S)*** ***(D)***																**Step length**	**6**

Operand	Type	Applicable soft element														Indexing
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

  Converting ***S*** from an integer to a floating-point number, and assigning the operation result to ***D*** when the energy flow is valid.

- Application instance

LD X0

FLT D0 D10

Converting D0=10005 from an integer to a floating-point number, and assigning the operation result to (D10,D11) when X0=ON. In this case, (D10,D11)=10005.0.

6.5.4 DFLT: Instruction for converting a long integer to a floating-point number

LAD: ├─┤ ├─[DFLT *(S)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DFLT ***(S)*** ***(D)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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

  Converting ***S*** from a long integer to a floating-point number, and assigning the operation result to ***D*** when the energy flow is valid.

- Application instance

LD X0

DFLT D0 D10

Converting (D0,D1)=100000 from a long integer to a floating-point number, and assigning the operation result to (D10,D11) when X0=ON. In this case, (D10,D11)=100000.0.

6.5.5 INT: Instruction for converting a floating-point number to an integer

LAD: ├─┤ ├─[INT *(S)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: INT ***(S)*** ***(D)***	**Step length**	**6**

Operand	Type	Applicable soft element														Indexing
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. Converting ***S*** from a floating-point number to an integer, and assigningthe operation result to ***D*** when the energy flow is valid.

2. This instruction affects zero flagand borrow flag. Setting the zero flag when the conversion result is 0. Setting the borrow flag when the decimals

of the result are rounded off. Setting the carry (overflow) flag when the result exceeds the data range of the long integer data.

- Note

When ***S***＞32767, ***D***=32767. When ***S***＜-32768, ***D***=-32768, and set the carry (overflow) flag bit.

- Application instance

LD X0
INT D0 D10

Converting (D0,D1)=10000.5from a floating-point number to an integer, and assigning the operation resultto D10 when X0=ON. In this case, (D0,D1) =10000.5.

6.5.6 DINT: Instruction for convert a floating-point number to a long integer

LAD: ├─┤ ├──[DINT *(S)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DINT *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	REAL	Constant							D				V		R	√
D	DINT			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

S: Source operand

D: Destination operand

- Function description

1. Converting ***S*** from a floating-point number to a long integer, and assigning the operation result to ***D*** when the energy flow is valid.

2. Setting the zero flag when the conversion result is 0. Setting the borrow flag when the decimals of the result are rounded off. Setting the carry (overflow) flag when the result exceeds the data range of the long integer data.

- Note

When ***S*** ＞ 2147483647, ***D***=2147483647. When ***S*** ＜ -2147483648, ***D***=-2147483648, and set the carry (overflow) flag bitsimultaneously.

- Application instance

LD X0
DINT D0 D10

Converting (D0,D1)=100000.5from a floating-point number to a long integer, and assigning the operation resultto (D10,D11) when X0=ON. In this case, (D10,D11)=100000.

6.5.7 BCD: Instruction for converting a word to a 16-bit BCD code

LAD: ├─┤ ├──[BCD *(S)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: BCD *(S)* *(D)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description

S: Source operand, ≤9999

D: Destination operand

- Function description

Converting ***S*** from an integer to a 16-bit BCD code, and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

If ***S***＞9999, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.

- Application instance

LD X0
BCD D0 D10

Converting D0=0x0D05 (3333) from an integer to a 16-bit BCD code, and assigning the operation resultto D10 when X0=ON. In this case, D10=0x3333 (13107).

6.5.8 DBCD: Instruction for converting a double word to a 32-bit BCD code

LAD: DBCD (S) (D)	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DBCD *(S)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

  S: Source operand, ≤99999999

  D: Destination operand
- Function description

  Converting ***S*** from a long integer to a 32-bit BCD code, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note

If ***S***>99999999, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.

- Application instance

LD X0

DBCD D0 D10

Converting (D0,D1)=0x3F940AA (66666666) from a long integer to a 32-bit BCD code, and assigning the operation resultto (D10,D11) when X0=ON. In this case, (D10,D11)=0x66666666 (1717986918).

6.5.9 BIN: Instruction for converting a 16-bit BCD code to a word

LAD: BIN (S) (D)	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: BIN *(S)* *(D)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description

  S: Source operand, its data format must comply with BCD code format

  D: Destination operand.
- Function description

  Converting ***S*** from a 16-bit BCD codeto an integer, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note

If the data format of ***S*** does not comply with BCD code format, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.

- Application instance

LD X0

BIN D0 D10

Converting D0=0x5555 (21845) from a 16-bit BCD code to an integer, and assigning the operation resultto D10 when X0=ON. In this case, D10=0x15B3 (5555).

6.5.10 DBIN: Instruction for converting a 32-bit BCD code to a double word

LAD: DBIN (S) (D)	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**

IL: DBIN (S) (D)									**Step length**		**7**					
Operand	Type	Applicable soft element														Indexing
S	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

  1. Converting ***S*** from a 32-bit BCD codeto a long integer, and assigning the obtained result to ***D*** when the energy flow is valid.

  2. The data format of ***S*** must comply with BCD code format.

- Note

  If the data format of ***S*** does not comply with BCD code format, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.

- Application instance

LD X0
DBIN D0 D10

Converting (D0,D1)=0x99999999 (2576980377) from a 32-bit BCD code to a long integer, and assigning the operation resultto (D10,D11) when X0=ON. In this case, (D10,D11)=0x5F5E0FF (99999999).

6.5.11 GRY: Instruction for converting a word to a 16-bit gray code

LAD: ├[GRY (S) (D)]									**Applicable model**		**IVC2L IVC1 IVC1S IVC3 IVC1L**					
									Influenced flag bit		**Zero flag, carry flag, and borrow flag**					
IL: GRY (S) (D)									**Step length**		**5**					
Operand	Type	Applicable soft element														Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

  Converting ***S*** from an integer to a 16-bit gray code, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0
GRY D0 D10

Converting D0=0xAAAA (43690) from an integer to a 16-bit gray code, and assigning the operation resultto D10 when X0=ON. In this case, D10=0xFFFF (65535).

6.5.12 DGRY: Instruction for converting a double word to a 32-bit gray code

LAD: ├─┤ ├─[DGRY (S) (D)]									**Applicable model**		**IVC2L IVC1 IVC1S IVC3 IVC1L**					
									Influenced flag bit		**Zero flag, carry flag, and borrow flag**					
IL: DGRY (S) (D)									**Step length**		**7**					
Operand	Type	Applicable soft element														Indexing
S	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

Converting ***S*** from a long integer to a 32-bit gray code, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0
DGRY D0 D10

Converting (D0,D1)=0x88888888 (2290649224) from a long integer to a 32-bit gray code, and assigning the operation resultto (D10,D11) when X0=ON. In this case, (D10,D11)=0xCCCCCCCC (3435973836).

6.5.13 GBIN: Instruction for converting a 16-bit gray code to a word

LAD: ⊢⊣ ⊢—[GBIN *(S)* *(D)*]									**Applicable model**			**IVC2L IVC1 IVC1S IVC3 IVC1L**			
									Influenced flag bit			**Zero flag, carry flag, and borrow flag**			
IL: GBIN *(S)* *(D)*									**Step length**			**5**			
Operand	Type	Applicable soft element													Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description

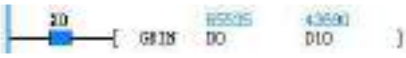
S: Source operand

D: Destination operand

- Function description

Converting ***S*** from a 16-bit gray code to an integer, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance



LD X0
GBIN D0 D10

Converting D0=0xFFFF (65535)from a 16-bit gray code to an integer, and assigning the operation resultto D10 when X0=ON. In this case, D10=0xAAAA (43690).

6.5.14 DGBIN: Instruction for converting a 32-bit gray code to a double word

LAD: ⊢⊣ ⊢—[DGBIN *(S)* *(D)*]									**Applicable model**			**IVC2L IVC1 IVC1S IVC3 IVC1L**			
									Influenced flag bit			**Zero flag, carry flag, and borrow flag**			
IL: DGBIN *(S)* *(D)*									**Step length**			**7**			
Operand	Type	Applicable soft element													Indexing
S	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

S: Source operand

D: Destination operand

- Function description

Converting S from a 32-bit gray code to a long integer, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0
DGBIN D0 D10

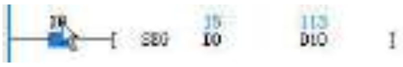
Converting (D0,D1)=0xCCCCCCCC (3435973836) from a 32-bit gray code to a long integer, and assigning the operation result to (D10,D11) when X0=ON. In this case, (D10,D11)=0x88888888 (2290649224).

6.5.15 SEG: Instruction for converting a word to a 7-segment code

LAD: ⊢⊣ ⊢—[SEG *(S)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: SEG *(S)* *(D)*	**Step length**	**5**

Operand	Type	Applicable soft element															Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description
  S: Source operand, ***S***≤15
  D: Destination operand
- Function description
  Converting ***S*** from aninteger to a7-segment code, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  If***S***＞15, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.
- Application instance



LD X0
SEG D0 D10

Converting D0=0x0F (15)from an integer to a 7-segment code, and assigning the operation result to D10 when X0=ON. In this case, D10=0x71 (113).

6.5.16 ASC: ASCII code conversion instruction

LAD: ASC (S1~S8) (D)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: **ASC *(S1–S8) (D)***	**Step length**	**19**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant														
S2	WORD	Constant														
S3	WORD	Constant														
S4	WORD	Constant														
S5	WORD	Constant														
S6	WORD	Constant														
S7	WORD	Constant														
S8	WORD	Constant														
D	WORD								D		C	T	V	Z	R	√

- Operand description
  S1 – ***S8***: Source operand (If the number of characters is less than 8, the remaining characters are filled with 0)
  Only supporting the characters whose ASCII code is 0x21 to 0x7E (input through keyboard, if the number of characters is less than 8, the remaining characters are filled with 0)
  D: Destination operand
- Function description
  Converting the character string ***S1*** – ***S8*** to ASCII code, and assigning the obtained result to the elements starting from ***D*** when the energy flow is valid. Storing two ASCII code data in the high/low byte of each ***D*** element when SM186=OFF. Storing one ASCII code data in low byte of each ***D*** element when SM186=ON.
- Application instance

LD M0
ASC 12345678 D0

Executing the ASCII conversion when M0=ON and data is stored in two modes:

- If SM186=OFF, the execution result is: D0=0x3231, D1=0x3433, D2=0x3635, D3=0x3837.

- If SM186=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: Instruction for convertinga 16-bit hex data to an ASCII code

LAD: ├─┤ ├──[ITA *(S1)* *(D)* *(S2)*]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ITA *(S1) (D) (S2)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

  S1: Source hex data to be converted

  D: Destination operand

  S2: Number of ASCII codes (1≤***S2***≤256)

- Function description

  Converting the hex data starting from ***S1*** element to S2 ASCII codes, and assigning the obtained result to the elements starting from ***D*** when the energy flow is valid. Storing two ASCII code data in the high/low byte of each ***D*** element when SM186=OFF. Storing one ASCII code data in low byte of each ***D*** element when SM186=ON.

- Note

  1. If ***S1*** and ***D*** use Kn addressing, Kn=4.

  2. If ***S2*** is not within the range of 1 to 256, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.

  3. If ***S1*** is a constant, when ***S2***≥4, the system default is ***S2***=4 and the system does not report an operand error once the default is adopted.

- Application instance

Source data: 0x9876.

LD M0

ITA 16#9876 D20 8

Executing ITA conversion operation when M0=ONand data is stored in two modes:

- If SM186=OFF, the execution result is: D20=0x3839, D21=0x3637.
- If SM186=ON, the execution result is: D20=0x39, D21=0x38, D22=0x37, D23=0x36.

6.5.18 ATI: Instruction for converting an ASCII code to a 16-bit hex data

LAD: ├─┤ ├──[ATI *(S1)* *(D)* *(S2)*]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ATI *(S1) (D) (S2)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S2	WORD	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 SM186=OFF, it is required thathigh/low bytes of ***S1*** be within this range)

  D: Destination operand

  S2: Number of ASCII codes (1≤***S2***≤256)

- Function description

  When energy flow is valid, convert the ASCII code data of the first ***S2*** of ***S1*** element to hex data, and store the result per 4 bits into the starting element of ***D***. When SM186=OFF, high/low byte of each ***D*** element stores two ASCII code data, when SM186=ON, low byte of each ***D*** element stores one ASCII code data.

Converting S2 ASCII codes starting from ***S1*** element tothe hex data, and storing the obtained result every 4 bits in the elements starting from ***D*** when the energy flow is valid. Storing two ASCII code data in the high/low byte of each ***D*** element when SM186=OFF. Storing one ASCII code data in low byte of each ***D*** element when SM186=ON.

- Note

1. If ***S1*** and ***D*** use Kn addressing, Kn=4.

2. If ***S1*** is not within the range of 0x30 to 0x39 or 0x41 to 0x46, or ***S2*** is not within the range of 1 to 256, the system reports an operand error, does not execute the conversion operation, and keeps the content of ***D***.

3. If ***S1*** is a constant, when SM186=OFF and ***S2***⩾2, the system default is ***S2***=2. When SM186=ON and S2⩾1, the system default is ***S2***=1. The system does not report an operand error once the default is adopted.

- Application instance

LD M0

ATI D10 D30 4

Source data: D10=0x3938, D11=0x3736, D12=0x3534, D13=0x3332.

Executing ITA conversion operation when M0=ONand the results generated according to the data storage mode are listed below:

- If SM186=OFF, execution result is: D30=0x8967.
- If SM186=ON, execution result is: D30=0x8642.

6.5.19 LCNV: Project conversion instruction

LAD: ├─┤ ├─[LCNV (S1) (S2) (D) (S3)]	**Applicable model**	**IVC2L IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: **LCNV *(S1) (S2) (D) (S3)***	**Step length**	**9**

Operand	Type	Applicable soft element													Indexing
S1	INT								D				V	R	
S2	INT								D				V	R	
D	INT								D				V	R	
S3	Word	Constant							D				V	R	

- Operand description

S1: The start address of the source operand to be converted

S2:The start address of conversion table

D: The start address that stores the conversion result

S3: Number of data to be converted (1≤***S3***≤64)

- Function description

When adopting the analog input module to read the external analog signal, you can use this instruction to convert the original analog read value into the corresponding project read value.

When temperature or analog modulesare used for temperature or analog measurement applications, if the temperature or project read value measured by the PLC deviates from the result measured by the standard thermometer or relevant standard instruments, this instruction can be used to make linear modification to calibrate the actual measurement.

Filling the conversion table with four parameters, namely low point measurement value V_{ML} , high point measurement value V_{MH} and corresponding low point standard value V_{SL} and high point standard value V_{SH} .

When the linear conversion is executed, the target standard value is generated by applying below formulato source data, in which S_n is the original input data, and D_n is conversion result data.

$$A=(V_{SL}-V_{SH})/(V_{ML}-V_{MH})*10000$$

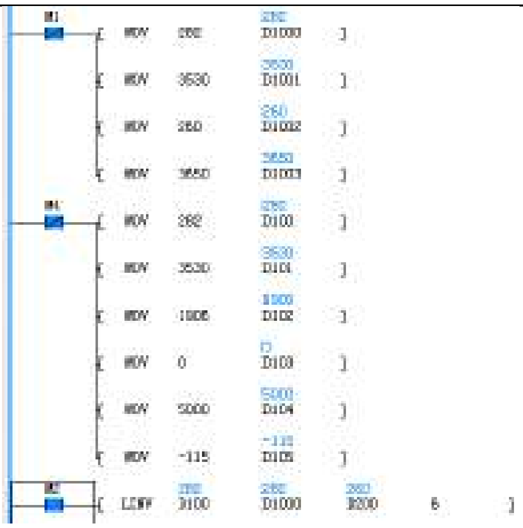
$$B=V_{SL}-(V_{ML}*A/10000)$$

$$D_n=(S_n*A/10000)+B$$

- Note

These four data in the conversion table has their actual meanings,for instance, the low point measurement value should be less than the high point measurement value. The result is inaccurate if the conversion result exceeds the integer range. If D_n is larger than 32767, it is 32767; if it is smaller than -32768, it is -32768.

- Application instance



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

Executing the LGNV instruction when M2=ON. The results generated according to data storage mode are listed below:

D200 = 260
D201 = 3650
D202 = 1955
D203 = -34
D204 = 5184
D205 = -154

6.5.20 RLCNV: Floating-point project conversion instruction

LAD: RLCNV (S1) (S2) (D) (S3)	Applicable model	IVC2L IVC3
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: RLCNV *(S1) (S2) (D) (S3)*	Step length	12

Operand	Type	Applicable soft element													Indexing
S1	REAL								D				V	R	
S2	REAL								D				V	R	
D	REAL								D				V	R	
S3	Word	Constant							D				V	R	

- Operand description

S1: The start address of the source operand to be converted

S2:The start address of conversion table

D: The start address that stores the conversion result

S3: Number ofdata to be converted (1≤***S3***≤64)

- Function description

When adopting the analog input module to read the external analog signal, you can use this instruction to convert the original analog read value into the corresponding project read value.

When temperature or analog modulesare used for temperature or analog measurement applications, if the temperature or project read value measured by the PLC deviates from the result measured by the standard thermometer or relevant standard instruments, this instruction can be used to make linear modification to calibrate the actual measurement.

Filling the conversion table with four parameters, namely low point measurement value V_{ML} , high point measurement value V_{MH} and corresponding low point standard value V_{SL} and high point standard value V_{SH} .

When the linear conversion is executed, the target standard value is generated by applying below formulato source data, in which S_n is the original input data, and D_n is conversion result data.

$$A = (V_{SL} - V_{SH})/(V_{ML} - V_{MH})*10000$$

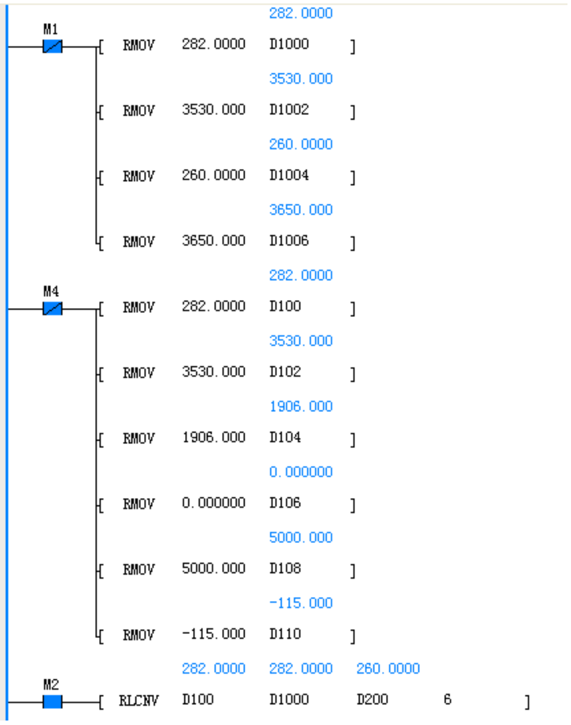
$$B = V_{SL} - (V_{ML} * A/10000)$$

$$D_n = (S_n * A/10000) + B$$

- Note

These four data in the conversion table has their actual meanings,for instance, the low point measurement value should be less than the high point measurement value. The result is inaccurate if the conversion result exceeds the integer range. If D_n is larger than 32767, it is 32767; if it is smaller than -32768, it is -32768.

- Application instance



LDI M1
RMOV 282 D1000
RMOV 3530 D1002
RMOV 260 D1004
RMOV 3650 D1006
LDI M4
RMOV 282 D100
RMOV 3530 D102
RMOV 1906 D104
RMOV 0 D106
RMOV 5000 D108
RMOV -115 D110
LD M2
RLCNV D100 D1000 D200 6

Executing the RLCNV instruction when M2=ON. The results generated according to data storage mode are listed below:

D200(D201) = 260
D202(D203) = 3650
D204(D205) = 1955
D206(D207) = -34.3288
D208(D209) = 5184.267
D210(D211) = -154.357

6.6 Word logic operation instructions

6.6.1 WAND: Word AND instruction

LAD: WAND *(S1)* *(S2)* *(D)*	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: WAND *(S1)* *(S2)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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

  Performingthe AND operationby bit on ***S1*** and ***S2***, and assigningthe obtained result to **D** when the energy flow is valid.

- Application instance

LD X0
WAND
D0 D1 D10

Performing the AND operationon D0=2#1011011010010011 (46739) and D1=2#1001001100101110 (37678) by bit, and assigning the obtained result to D10 when X0=ON. In this case, D10=2#1001001000000010 (37378).

6.6.2 WOR: Word OR instruction

LAD: ├─┤ ├─[WOR (S1) (S2) (D)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: WOR (S1) (S2) (D)	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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

  Performingthe OR operationby bit on ***S1*** and ***S2***, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0
WOR D0 D1 D10

Performing the OR operation by bit on D0=2#1011011010010011 (46739) and D1=2#1001001100101110 (37678), and assigning the obtained result to D10 when X0=ON. In this case, D10=2#1011011110111111 (47039).

6.6.3 WXOR: Word XOR instruction

LAD: ├─┤ ├─[WXOR (S1) (S2) (D)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: WXOR (S1) (S2) (D)	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description

  S1: Source operand 1

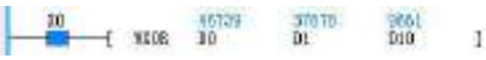
  S2: Source operand 2

  D: Destination operand

- Function description

  Performingthe XOR operationby bit on ***S1*** and ***S2***, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance



LD X0
WXOR D0 D1 D10

Performing the XOR operation by bit on D0=2#1011011010010011 (46739) and D1=2#1001001100101110 (37678), and assigning the obtained result to D10 when X0=ON. In this case, D10=2#0010010110111101 (9661).

6.6.4 WINV: Word INV instruction

LAD: ├─┤ ├─[WINV (S) (D)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: WINV *(S) (D)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√

- Operand description

  S: Source operand

  D: Destination operand

- Function description

  Performingthe INV operationby bit on ***S***, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0
WINV D0 D10

Performing the INV operation by bit on D0= (46739), and assigning the obtained result to D10 when X0=ON. In this case, D10= (18796).

6.6.5 DWAND: Double word AND instruction

LAD: ├─┤ ├─[DWAND (S1) (S2) (D)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: DWAND *(S1) (S1) (D)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

  S1: Source operand 1

  S2: Source operand 2

  D: Destination operand

- Function description

  Performingthe AND operationby bit on ***S1*** and ***S2***, and assigning the obtained result to **D** when the energy flow is valid.

- Application instance

LD X0
DWAND D0 D2 D10

Performing the AND operationon (D0,D1)=2#10110010101001101110011001010010 (2997282386) and (D2, D3)=2#00111010001110110011000100110011 (976957747) by bit, and assigning the obtained result to (D10, D11) when X0=ON. In this case, (D10, D11)=2#00110010001000100010000000010010 (841097234).

6.6.6 DWOR: Double word OR instruction

LAD: ├─┤ ├─[DWOR (S1) (S2) (D)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: DWOR *(S1) (S2) (D)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

S2	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

  S1: Source operand 1

  S2: Source operand 2

  D: Destination operand

- Function description

  Performingthe OR operationby bit on ***S1*** and ***S2***, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0

DWOR D0 D2 D10

Performing the OR operation by bit on (D0, D1)=2#10110010101001101110011001010010 (2997282386) and (D2, D3)=2#00111010001110110011000100110011 (976957747), and assigning the obtained result to (D10, D11) when X0=ON. In this case, (D10, D11)=2#10111010101111111111011101110011 (3133142899).

6.6.7 DWXOR: Double word XOR instruction

LAD: DWXOR (S1) (S2) (D)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: **DWXOR *(S1) (S2) (D)***	Step length	10

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description

  ***S1*:** Source operand 1

  S2: Source operand 2

  D: Destination operand

- Function description

  Performingthe XOR operationby bit on ***S1*** and ***S2***, and assigning the obtained result to ***D*** when the energy flow is valid.

- Application instance

LD X0

DWXOR D0 D2

D10

Performing the XOR operation by bit on (D0,D1)=2#10110010101001101110011001010010(2997282386) and (D2,D3)=2#00111010001110110011000100110011 (976957747), and assigning the obtained result to (D10, D11) when X0=ON. In this case, (D10, D11)=2#10001000100111011101011101100001 (2292045665).

6.6.8 DWINV: Double word negation instruction

LAD: DWINV (S) (D)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: **DWINV *(S) (D)***	Step length	7

Operand	Type	Applicable soft element														Indexing
S	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			KnY	KnM	KnS	KnLM		D		C		V		R	√

- Operand description
  S: Source operand
  D: Destination operand
- Function description
  Performingthe INV operationby bit on ***S***, and assigning the obtained result to ***D*** when the energy flow is valid.
- Application instance

LD X0
DWINV D0 D10

Performing the INV operation by bit on (D0,D1)=2#10110010101001101110011001010010 (2997282386), and assigning the obtained result to (D10,D11) when X0=ON. In this case, (D10,D11)=2#01001101010110010001100110101101 (1297684909).

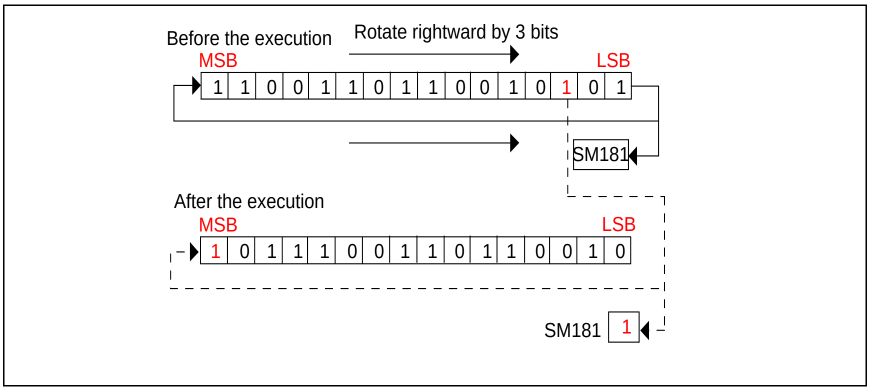
6.7 Bit shift and rotate instructions

6.7.1 ROR: 16-bit rotate right instruction

LAD: ROR (S1) (D) (S2)															
Applicable model	**IVC2L IVC1 IVC1S IVC3 IVC1L**														
Influenced flag bit	**Carry flag SM181**														
IL: ROR *(S1)* *(D)* *(S2)*	**Step length**	**7**													
Operand	Type	Applicable soft element													Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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
  Cyclically shifting the data of ***S1*** rightward by ***S2*** bits, assigning the obtained result to ***D,*** and storingthe destination bit in the carry flag bit (SM181)when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 4.
- Application instance

LD M0
ROR D0 D10 3



Cyclically shifting D0=2#1100110110010101 (52629) rightward by 3 bits, assigning the obtained result to D10, and storing the destination bit in the carry flag bit when M0=ON. In this case, D10=2#1011100110110010 (47538) and SM181=ON.

6.7.2 ROL: 16-bit rotate left instruction

LAD: ROL (S1) (D) (S2)																
Applicable model	**IVC2L IVC1 IVC1S IVC3 IVC1L**															
Influenced flag bit	**Carry flag SM181**															
IL: ROL *(S1)* *(D)* *(S2)*	**Step length**	**7**														
Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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

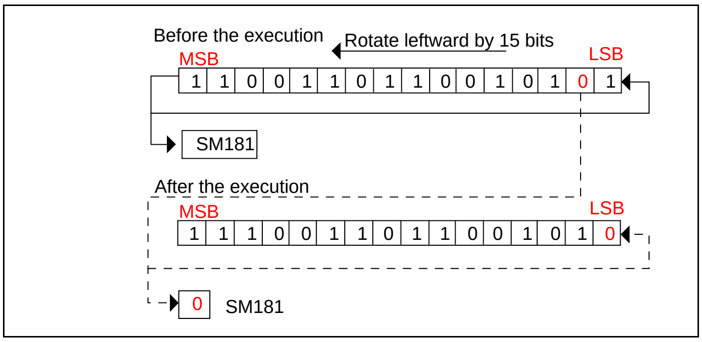
  Cyclically shifting the data of ***S1*** leftward by ***S2*** bits, assigning the obtained result to ***D,*** and storingthe destination bit in the carry flag bit (SM181)when the energy flow is valid.

- Note

  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 4.

- Application instance

LD M0
ROL D0 D10 15



Cyclically shiftingD0=2#1100110110010101 (52629) leftward by 15 bits, assigning the obtained result to D10, and storing the destination bit in the carry flag bit when M0=ON. In this case, D10=2#1110011011001010 (59082) and SM181=OFF.

6.7.3 RCR: Instruction for 16-bit rotate right with carry flag bit

LAD: RCR (S1) (D) (S2)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	Carry flag SM181
IL: RCR (S1) (D) (S2)	Step length	7

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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

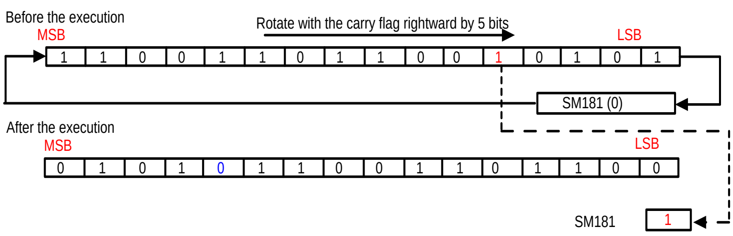
  Cyclically shifting the data of ***S1*** with the carry flag bit (SM181) rightward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 4.

- Application instance

LD M0
RCR D0 D10 5



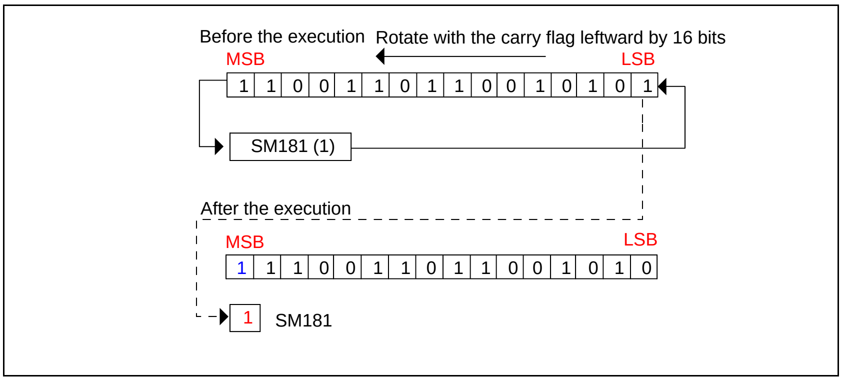
Cyclically shifting D0=2#1100110110010101 (52629) with the carry flag bit (SM181=OFF) rightward by 5 bits, and assigning the obtained result to D10 when M0=ON.In this case, D10=2#0101011001101100 (22124) and SM181=ON.

6.7.4 RCL: Instruction for 16-bit rotate left with carry flag bit

LAD: ├──┤ ├──[RCL (S1) (D) (S2)]									Applicable model		IVC2L IVC1 IVC1S IVC3 IVC1L						
									Influenced flag bit		Carry flag SM181						
IL: RCL (S1) (D) (S2)									Step length		7						
Operand	Type	Applicable soft element															Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R		√
D	WORD			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
  Cyclically shifting the data of ***S1*** with the carry flag bit (SM181) leftward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 4.
- Application instance

LD M0
RCL D0 D10 16



Cyclically shifting D0=2#1100110110010101 (52629) with the carry flag bit (SM181=ON) rightward by16 bits, and assigning the obtained result to D10 when M0=ON. In this case, D10=2#1110011011001010 (59082) and SM181=ON.

6.7.5 DROR: 32-bit rotate right instruction

LAD: ├──┤ ├──[DROR (S1) (D) (S2)]									Applicable model		IVC2L IVC1 IVC1S IVC3 IVC1L						
									Influenced flag bit		Carry flag SM181						
IL: DROR (S1) (D) (S2)									Step length		9						
Operand	Type	Applicable soft element															Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R		√
D	DWORD			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
  Cyclically shifting the data of ***S1*** rightward by ***S2*** bits, assigning the obtained result to ***D,*** and storingthe destination bit in the carry flag bit (SM181)when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 8.
- Application instance

LD M0
DROR D0 D10 7

Cyclically shifting D0 (D1)=2#10110011100110001001110010101100 (3013123244) rightward by 7 bits, assigning the obtained result to (D10, D11), and storing the destination bit in the carry flag bit when M0=ON. In this

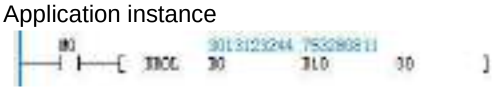
case,
(D10,D11)=2#010110010110011100 11000100111001 (1499935033) and SM181=OFF.
For details, refer to the diagram of ROR instruction.

6.7.6 DROL: 32-bit rotate left instruction

LAD: ├─┤ ├──[DROL (S1) (D) (S2)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Carry flag SM181**
IL: DROL *(S1)* *(D)* *(S2)*	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			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
  Cyclically shifting the data of ***S1*** leftward by ***S2*** bits, assigning the obtained result to ***D*,** and storingthe destination bit in the carry flag bit (SM181)when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 8.
- Application instance



LD M0
DROL D0 D10 30

Cyclically shifting (D0,D1)=2#10110011100110001001110010101100 (3013123244) rightward by 30 bits, assigning the obtained result to (D10, D11), and storing the destination bit in the carry flag bit when M0=ON. In this case, (D10,D11)=2#00101100111001100010011100101011 (753280811) and SM181=ON.
For detals, refer to the diagram of ROL instruction.

6.7.7 DRCR: Instruction for 32-bit rotate right with carry flag bit

LAD: ├─┤ ├──[DRCR (S1) (D) (S2)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	**Carry flag SM181**
IL: DRCR *(S1)* *(D)* *(S2)*	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			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
  Cyclically shifting the data of ***S1*** with the carry flag bit (SM181) rightward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 8.

- Application instance

1. Cyclically shifting (D0,D1)=2#10110011100110001001110010101100 (3013123244)with the carry flag bit (SM181=OFF) rightward by 11 bits, and assigning the obtained result to (D10,D11) when M0=ON.In this case, (D10,D11)=2#00101011000101100111001100010011 (722891539) and SM181=ON.

LD M0
DRCR D0 D10 11

2. For details, refer to the diagram of RCR instruction.

6.7.8 DRCL: Instruction for 32-bit rotate left with carry flag bit

LAD: ├─┤ ├─[DRCL (S1) (D) (S2)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	Carry flag SM181
IL: DRCL (S1) (D) (S2)	Step length	9

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			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

Cyclically shifting the data of ***S1*** with the carry flag bit (SM181) leftward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.

- Note

S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 8.

- Application instance

LD M0
DRCL D0 D10 25

1. Cyclically shifting (D0,D1)=2#10110011100110001001110010101100 (3013123244)with the carry flag bit (SM181=ON) rightward by25 bits, and assigning the obtained result to (D10,D11) when M0=ON. In this case, (D10,D11)=2#00101100010110011100110001001 1100 (1488165020) and SM181=ON.

2. For details, refer to the diagram of RCL instruction.

6.7.9 SHR: 16-bit shift right instruction

LAD: ├─┤ ├─[SHR (S1) (D) (S2)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: SHR (S1) (D) (S2)	Step length	7

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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

Shifting the data of ***S1*** rightward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.

- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 4.
- Application instance

	LD M0 SHR D0 D10 5

Before the execution Rotate rightward by 5 bits

MSB LSB

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

After the execution

MSB

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

Shifting D0=2#0111101011011100 (31452) rightward by 5 bits, and assigning the obtained result to D10 when M0=ON. In this case, D10=2#0000001111010110 (982).

6.7.10 SHL:16-bit shift left instruction

LAD: SHL (S1) (D) (S2)								Applicable model		IVC2L IVC1 IVC1S IVC3 IVC1L						
								Influenced flag bit								
IL: SHL (S1) (D) (S2)								Step length		7						
Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD			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
  Shifting the data of ***S1*** leftward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 4.
- Application instance

LD M0
SHL D0 D10 7

Before the execution Rotate leftward by 7 bits

MSB LSB

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

After the execution

MSB LSB

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

Shifting D0=2#0111101011011100 (31452) leftward by 7 bits, and assigning the obtained result to D10 when M0=ON. In this case, D10=2#0110111000000000 (28160).

6.7.11 DSHR: 32-bit shift right instruction

LAD: DSHR (S1) (D) (S2)		Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L	
		Influenced flag bit		
IL: DSHR (S1) (D) (S2)		Step length	9	
Operand	Type	Applicable soft element		Indexing

S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			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
  Shifting the data of ***S1*** rightward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 8.
- Application instance

LD M0
DSHR D0 D10 10

1. Shifting (D0,D1)=2#01110011100110001001110010101100 (1939381420) rightward by 10 bits, and assigning the obtained result to (D10,D11) when M0=ON. In this case, (D10,D11)=2#00000000000111001110011000100111 (1893927).
2. For details, refer to the diagram of SHR instruction.

6.7.12 DSHL: 32-bit shift left instruction

LAD: DSHL *(S1)* *(D)* *(S2)*		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: DSHL *(S1)* *(D)* *(S2)*		**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
S1	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	DWORD			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
  Shifting the data of ***S1*** leftward by ***S2*** bits, and assigning the obtained result to ***D*** when the energy flow is valid.
- Note
  S2 is greater than or equal to 0. When ***S1*** uses Kn addressing, Kn needs to be equal to 8.
- Application instance

LD M0
DSHL D0 D10 15

1. Shifting (D0,D1)=2#01110011100110001001110010101100 (1939381420) leftward by 15 bits, and assigning the obtained result to (D10,D11) when M0=ON. In this case, (D10,D11)=2#01001110010101100000000000000000 (1314258944).
2. For details, refer to the diagram of SHR instruction.

6.7.13 SFTR: Bit string shift right instruction

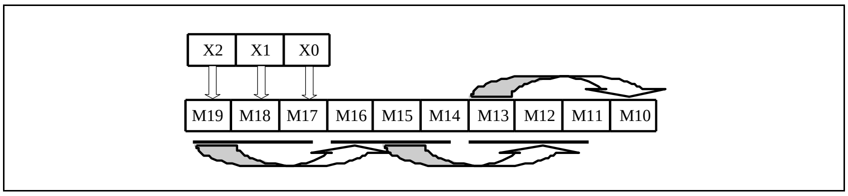
LAD: SFTR *(S1)* *(D)* *(S2)* *(S3)*		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: SFTR *(S1)* *(D)* *(S2)* *(S3)*		**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
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
  Shifting the content of the ***S2*** units starting from ***D*** rightward by ***S3*** units, discarding the ***S3*** data on the rightmost end, and shifting the content of ***S3*** units starting from ***S1*** to the left end of the word string when the energy flow is valid.
- Note
  Left right order, elements with large element No. areon the left, and elements with small element No. areon the right.
  Both ***S2*** and ***S3*** aregreater than or equal to zero.
- Application instance

LD M0
SFTR X0 M10 10 3



1. Shifting the content of 10 units starting from M10 rightward by 3 units, discarding M10 to M12 on the rightmost end, and shifting the content of 3 units starting from X0 to the left end of the bit string when M0=ON.

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, and M19=1.

3. After execution: The contents of X0 to X2 are kept. M10=0, M11=0, M12=1, M13=0, M14=0, M15=0, M16=1, M17=1, M18=0, and M19=1.

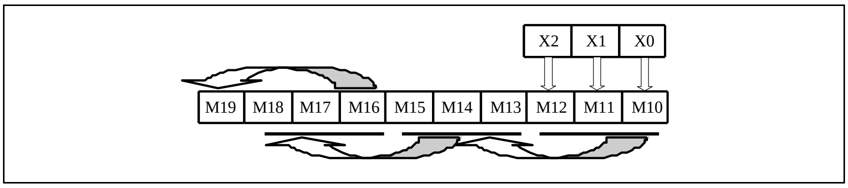
6.7.14 SFTL: Bit string shift left instruction

LAD: ├─┤ ├─[SFTL *(S1)* *(D)* *(S2)* *(S3)*	**Applicable model** **IVC2L IVC1 IVC1S IVC3**
	Influenced flag bit
IL: SFTL *(S1)* *(D)* *(S2)* *(S3)*	**Step length** **9**

Operand	Type	Applicable soft element														Indexing
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
- Note
  1. Left right order, elements with large element No. areon the left, and elements with small element No. areon the right.
  2. Both ***S2*** and ***S3*** are greater than or equal to zero.
- Function description
  Shifting the content of the ***S2*** units starting from ***D*** rightward by ***S3*** units, discarding the ***S3*** data on the leftmost end, and shifting the content of ***S3*** units starting from ***S1*** to the right end of the word string when the energy flow is valid.
- Application instance

LD M0
SFTL X0 M10 10 3



1. Shifting the content of 10 units starting from M10 leftward by 3 units, discarding M17 to M19 on the leftmost end, and shifting the content of 3 units starting from X0 to the right end of the bit string when M0=ON.

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, and M19=1。

3. After execution: The contents of X0 to X2 are kept. M10=1, M11=0, M12=1, M13=0, M14=1, M15=1, M16=0, M17=0, M18=1, and M19=0.

6.8 Peripheral instructions

6.8.1 FROM: Instruction for reading words from a special module buffer register

LAD:	Applicable model	IVC2L IVC3
─┤├─[FROM *(S1)* *(S2)* *(D)* *(S3)*]	Influenced flag bit	
IL: FROM *(S1)* *(S2)* *(D)* *(S3)*	**Step length**	**9**

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant														
S2	INT	Constant														
D	INT								D				V		R	√
S3	INT	Constant														

- Operand description

S1: SN of the special modules visited

Settable range: 0 to 7. When accessing a special module that does not exist in the system, the system reports that the special module address is invalid.

S2: The start address of the BFM buffer of the special module to be read

Settable range: 0 to 32767. When accessing an invalid BFM address, the system reports that BFM unit of the assessed special module exceeds the range.

D: The element where the data read from the special module is stored.

S3: Number of the consecutive BFM buffers (single word) to be read

Data range: 1 to 32767. When accessing an invalid address of the BFM buffer, the system reports that the BFM buffer of the assessed special module exceeds the range.

- Function description

Reading consecutively ***S3*** word data starting from ***S2*** buffer in the designated BFM of designated special module (SN: ***S1***), and consecutively storing these data in ***S3*** word elements starting from ***D***.

- Note

The execution time of the FROM instruction is relatively long, and related to the value of ***S3***.

- Application instance

LD M0
FROM 0 3 D100 2

Reading consecutively 2 data starting from the buffer 3 in the BFM of No. 0 special module, and storing these data in D100 and D101 respectively when M0 is ON.

6.8.2 DFROM: Instruction for reading double words from a special module buffer register

LAD:	Applicable model	IVC2L IVC3
─┤├─[DFROM *(S1)* *(S2)* *(D)* *(S3)*]	Influenced flag bit	
IL: DFROM *(S1)* *(S2)* *(D)* *(S3)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant														
S2	INT	Constant														
D	DINT								D				V		R	√
S3	INT	Constant														

- Operand description

  S1: SN of the special modules visited

  Settable range: 0 to 7. When accessing a special module that does not exist in the system, the system reports that the special module address is invalid.

  S2: The start address of the BFM buffer of the special module to be read

  Settable range: 0 to 32767. When accessing an invalid BFM address, the system reports that BFM unit of the assessed special module exceeds the range.

  D: The element where the data read from the special module is stored.

  S3: Number of the consecutive BFM buffers (double word) to be read

  Data range: 1 to 32767. When accessing an invalid address of the BFM buffer, the system reports that the BFM buffer of the assessed special module exceeds the range.

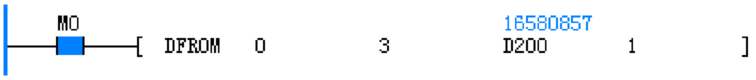
- Function description

  Reading consecutively ***S3*** word data starting from ***S2*** buffer in the designated BFM of designated special module (SN: ***S1***), and consecutively storing these data in ***S3*** double word elements starting from ***D***.

- Note

  The execution time of the DFROM instruction is relatively long, and related to the value of ***S3***.

- Application instance



LD M0

DFROM 0 3 D200 1

Reading1 double data starting from the buffer 3 in the BFM of No. 0 special module, and storing these double data in D200 and D201 respectively when M0 is ON.

6.8.3 TO: Instruction for writing words from a special module buffer register

LAD: ─┤├─[TO (S1) (S2) (S3) (S4)]	Applicable model	IVC2L IVC3
	Influenced flag bit	
IL: TO (S1) (S2) (S3) (S4)	Step length	9

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant														
S2	INT	Constant														
S3	INT	Constant							D				V		R	√
S4	INT	Constant														

- Operand description

  S1: SN of the special modules visited

  Settable range: 0 to 7. When accessing a special module that does not exist in the system, the system reports that the special module address is invalid.

  S2: The start address of the BFM buffer of the special module to be written

  Settable range: 0 to 32767. When accessing an invalid BFM address, the system reports that BFM unit of the assessed special module exceeds the range.

  S3: Data to be written into the special modules

  S4: Number of the consecutive BFM buffers (single word) to be written.

  Data range: 1 to 32767. When accessing an invalid address of the BFM buffer, the system reports that the BFM buffer of the assessed special module exceeds the range.

- Function description

  Writing the data from consecutive ***S4*** word units starting from ***S3*** into ***S4*** word elements starting fromthe designated BFM buffer (address is ***S2***)of the designated special module (SN: ***S1***).

- Note

  The execution time of the TO instruction is relatively long, and related to the value of ***S4***.

- Application instance

LD SM0

TO 0 8 1000 2

Writing 1000 respectively into BFM #8 and BFM #9 buffers of No. 0 special module when the PLC is running.

6.8.4 DTO: Instruction for writing double words from a special module buffer register

LAD: DTO (S1) (S2) (S3) (S4)		**Applicable model**	**IVC2L IVC3**
		Influenced flag bit	
IL: DTO *(S1)* *(S2)* *(S3)* *(S4)*		**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant														
S2	INT	Constant														
S3	DINT	Constant							D				V		R	√
S4	INT	Constant														

- Operand description

  S1: SN of the special modules visited

  Settable range: 0 to 7. When accessing a special module that does not exist in the system, the system reports that the special module address is invalid.

  S2: The start address of the BFM buffer of the special module to be written

  Settable range: 0 to 32767. When accessing an invalid BFM address, the system reports that BFM unit of the assessed special module exceeds the range.

  S3: Data to be written into the special modules

  S4: Number of the consecutive BFM buffers (double word) to be written.

  Data range: 1 to 32767. When accessing an invalid address of the BFM buffer, the system reports that the BFM buffer of the assessed special module exceeds the range.

- Function description

  Writingthe data from consecutive***S4*** double word units starting from ***S3*** into ***S4*** double word elements starting fromthe designated BFM buffer (address is ***S2***)of the designated special module (SN: ***S1***).

- Note

  The execution time of the DTO instruction is relatively long, and related to the value of ***S4***.

- Application instance

  SM0 DTO 0 8 16711935 1

  LD SM0
  DTO 0 8 16711935 1

  Writing one double word data (16711935)into BFM #8 and BFM #9 buffers(they can constitute one double word) of No. 0 special module when the PLC is running.

6.8.5 VRRD: Instruction for readingthe value of an analog potentiometer

LAD: VRRD (S) (D)		**Applicable model**	**IVC2L IVC1**
		Influenced flag bit	
IL: VRRD *(S)* *(D)*		**Step length**	**5**

Operand	Type	Applicable soft element													Indexing
S	WORD	Constant													
D	WORD								D				V		√

- Operand description

  S: Designated analog potentiometer No.

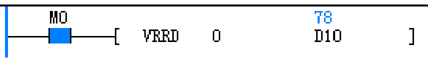
  Settable range:: 0 to 255. If ***S*** is set outside this range, the system reports an operand error.

  D: The element where the read analog potentiometer value is stored. Range:0﹣255.

- Function description

  Reading the value of the designated analog potentiometer, and storing it into the designated element.

- Application instance



Reading the value of No. 0 analog potentiometer in the system, and putting the read value into D10 when M0 is ON.

LD M0

VRRD 0 D10

6.8.6 REFF: Instruction for setting input filtering constant

LAD: ⊢⊣ ⊢[REFF *(S)*]										**Applicable model**		**IVC2L IVC1 IVC1S IVC1L**				
										Influenced flag bit						
IL: REFF *(S)*										**Step length**		**3**				
Operand	Type	Applicable soft element														Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

  S: Input filtering constant
  - IVC2L: Setting range: 0－64ms. Any data larger than 64 is processed as 64.
  - IVC1: The actual valid data is 0, 8, 16, 32 and 64. Any parameter less than 8 is processed as 0 and less than 16 is processed as 8. The parameter less than 32 is processed as 16 and less than 64 is processed as 32, and other data are processed as 64.

- Function description

  Setting the input filtering constant of X0－X7.

- Note

  The input filtering constant is valid only for the port used as normal input, and it is invalid for the port used as high-speed input.

- Application instance

LD M0

REFF 30

Setting the input filtering constant time to 30ms when X10 is ON.

6.8.7 REF: Instruction for immediately refreshing I/O

LAD: ⊢⊣ ⊢[REF *(D)* *(S)*									**Applicable model**		**IVC2 IVC1 IVC1S IVC3 IVC1L**				
									Influenced flag bit						
IL: REF *(D)* *(S)*									**Step length**		**5**				
Operand	Type	Applicable soft element													Indexing
D	BOOL		X	Y											
S	INT	Constant													

- Operand description

  D: The start X or Y element to be refreshed

  Designating the start soft element No. as an integral multiple of 8. For example, X0, X10, X20…or Y0, Y10, Y20…. The min. bit is 0.

  S: Number of I/O ports to be refreshed

  It should always be a multiple of 8, for example, 8, 16, ……, 256, and so on. Other values(except a multiple of 8) are wrong.

- Function description

  Generally, the PLC does not refresh its I/O points before the user program ends. However, if it is required to read the latest input state or immediately refresh the output state during the operation process, you can use the REF instruction.

- Note

  1. The subscript number of input port (Xn, Yn) should be an integer multiple of 8.

  2. Number of the refreshed (port) should also be an integer multiple of 8.

  3. Generally, the REF instruction is used to refresh I/O immediately between the FOR-NEXT instruction and the CJ instruction.

  4. You can also use the REF instruction to obtain the latest input information and output the operation result without delay during the execution of the interrupt subprogram with I/O.

5. To refresh arelay output, you need to consider the response time.

- Application instance

LD M0
REF Y0 8

Immediately outputting the statesat Y0 to Y7 regardless of the scan cycle when M0 is ON.

6.8.8 EROMWR: EEPROM write instruction

LAD: ┣━┫ ┣━[EROMWR (S1) (S2)]	Applicable model	IVC2L IVC1
	Influenced flag bit	
IL: EROMWR *(S1)* *(S2)*	**Step length**	**6**

Operand	Type	Applicable soft element														Indexing
S1	WORD								D						R	
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

S1: The start address of the elements being written(Range:D6000﹣D6999)

S2: Number of the elements being written (S2＜16, and S1＋S2＜D7000)

- Note

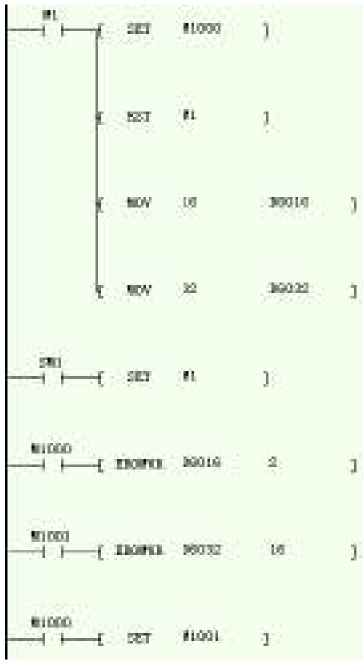
One EEPROM instruction operation makes the scan cycle 2-5 ms longer. It is recommended to set ***S1*** to 6000 plus an integer multiple of 16, such as D6000, D6016 and D6032.

- Function description

1. Generally, partial PLC data can be stored throughusing the function for saving data at power outage. However, during the calculation, you can save the intermediate data into EEPROM with the ERPROM instruction.

2. This instruction is executed upon the rising edge.

- Application instance



LD	M1
SET	M1000
RST	M1
MOV	16 D6016
MOV	32 D6032
LD	SM1
SET	M1
LD	M1000
EROMWR	D6016 2
LD	M1001
EROMWR	D6032 16
LD	M1000
SET	M1001

In the preceding example, two sets of D elements are stored in the EEPROM.

SM1 and M1 makes M1000 generate a rising edge in the second scan cycle, and triggers the execution of the first EROMWR instruction. M1001 makes the second rising edge, and triggers the execution of the second EROMWR instruction.

6.8.9 PR: Printing instruction

LAD: ┣━┫ ┣━[PR (S) (D)]	Applicable model	IVC2L IVC3
	Influenced flag bit	
IL: PR *(S)* *(D)*	**Step length**	**5**

Operand	Type	Applicable soft element													Indexing
S	WORD								D		C	T		R	√
D	BOOL			Y											

- Operand description

  S: The start number of the soft elements that save data.

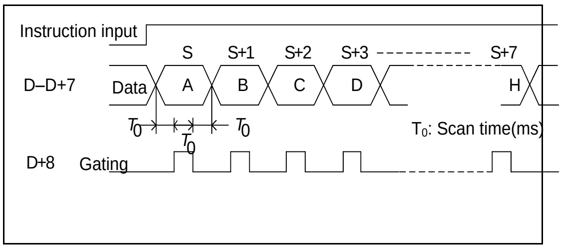
  D: The start number of Y elements that output data.

- Function description

  1. Outputting the data saved in LSB of S–S+7 (1 byte) to D–D+7 in mode of hours and minutes, and fixing the enabling signal to Y0.

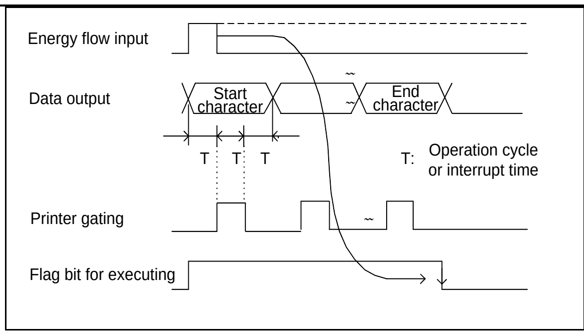
  2. SM71 is a flag that indicates the printing instruction is being executed. SM71 is switched on during the printing process, and reset after the printing is completed.

  The sequence diagram is shown below.



  3. When the special register SM70 is OFF, the serial output is fixed at 8 bytes; when it is on, the serial output may be 1 to 16 characters; the code H00 (NUL code) indicates that the previous character is the last character of a string.

  When SM70 is ON, the sequence diagram is as follows.



Resetting SM71 when the energy flow is invalid.

Note

1. Printing only once when the energy flow is valid.
2. SM71 (which indicates that the printing instruction is being executed) can be used to control the ON/OFF of the printing instruction.

- Application instance

- Note

  1. This instruction is applicable only to transistor output modules.

  2. This instruction is executed with the scan cycle.

  3. Only one instruction can be executed at a time. When the printing is done, SM71 is reset.

6.8.10 TKY: Numeric key input instruction

LAD: TKY (S) (D1) (D2)	Applicable model	IVC3
	Influenced flag bit	
IL: TKY (S) (D1) (D2)	Step length	7

Operand	Type	Applicable soft element																		Indexing
S	BOOL	X	Y	M	S	SM	LM													
D1	INT							KnY	KnM	KnLM	D	SD	C	T	V	Z		R		√
D2	BOOL		Y	M	S	SM	LM													

- Operand description

  S: Start bit of the data entered through numeric key(occupying 10 bits).

  D1: Data storage unit.

  D2: Number of the soft element corresponding to the ON/OFF state of the input key (occupying 11 points).

- Function description

  1. S to S+9 are used for keypad input, and the input data is stored in D1; D2 to D2+9 output the information entered through keypad; D2+10 detects keypad input, that is, when any one input is ON, this bit is set.

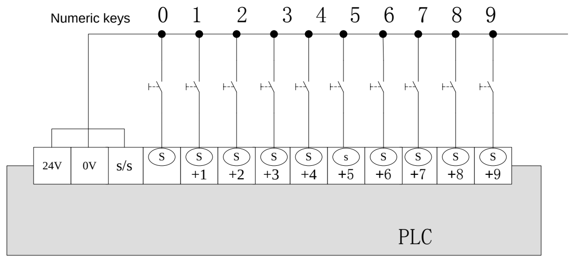
  1) Inputting the value of D1

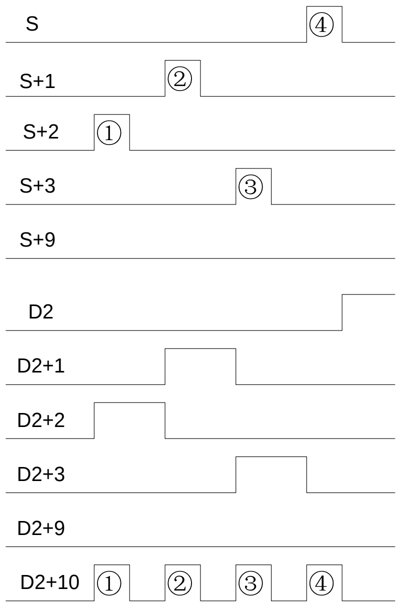
  In the figure below, the value 2130 is stored in D1 after you press the numeric keys in the order of 1, 2, 3, and 4.

  2) Key information of D2-D2+10

  Whether the key information of D2-D2+9 is ON or OFF depends on the key you press.

  When any one of the keys 0-9 is pressed, the keypad detection output of D2+10 is ON.





M1000

When you press X2, X1, X3, and X0 in sequence, the content of D7999 becomes 2130. After X2 is pressed, M1002 is set to ON until another key is pressed,and so is the case when other keys are pressed. After any one of the keys is pressed, the keypad detection output M1010 is ON only during the pressing time.

- Note

1. When multiple keys are pressed simultaneously, only the information input by the key pressed first is valid.
2. When the energy flow is OFF, the content of D1 does not change, but D2 to D2+10 are set to OFF.
3. If the inputted data exceeds 9999, the MSB overflows first.
4. After an input key is pressed, the keypad detection output D2 corresponding to the key is set to ON until another input key.
5. Only one TKY instruction can be used in the program, and but you can use it multiple times through indexing.

- Application instance

LD M0
TKY X0 D7999

6.9 Real-time clock instructions

6.9.1 TRD: Real-time clock read instruction

<table>
<tr><td colspan="9" rowspan="2">LAD:
├──┤ ├──[TRD (D)]</td><td colspan="3">Applicable model</td><td colspan="5">IVC2L IVC1 IVC3 IVC1L</td></tr>
<tr><td colspan="3">Influenced flag bit</td><td colspan="5"></td></tr>
<tr><td colspan="9">IL: TRD (D)</td><td colspan="3">Step length</td><td colspan="5">3</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="14">Applicable soft element</td><td>Indexing</td></tr>
<tr><td>D</td><td>WORD</td><td></td><td></td><td></td><td></td><td></td><td></td><td>D</td><td></td><td></td><td></td><td>V</td><td></td><td>R</td><td></td><td>√</td></tr>
</table>

- Operand description

  D: Reading the start unit that storesthe system time, and storing the read time in 7 consecutive units starting from the unit assigned by ***D***.

- Function description

  Reading the system time and storing it in the storage unit assigned by ***D***.

- Note

  The TRD instruction fails if there are something wrong in the system clock setting.

- Application instance

LD M0
TRD D10

Sending the system time to 7 units starting from D10 respectively when M0 is ON.

The execution results of the instruction are listed below:

	Element	Project	Clock data		Element	Project
Special data register for the real-time clock	SD100	Year	2000–2099	———————→	D10	Year
	SD101	Month	1–12	———————→	D11	Month
	SD102	Day	1–31	———————→	D12	Day
	SD103	Hour	0–23	———————→	D13	Hour
	SD104	Minute	0–59	———————→	D14	Minute
	SD105	Second	0–59	———————→	D15	Second
	SD106	Week	0–6	———————→	D16	Week

6.9.2 TWR: Real-time clock write instruction

LAD: TWR (S)														**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
														Influenced flag bit	
IL: TWR *(S)*														**Step length**	**3**
Operand	Type	Applicable soft element													Indexing
S	WORD							D				V		R	√

- Operand description

S: The soft element where the system time is to be written

	Element	Project	Clock data		Element	Project
Data for clock setting	D10	Year	2000–2099	→	SD100	Year
	D11	Month	1–12	→	SD101	Month
	D12	Day	1–31	→	SD102	Day
	D13	Hour	0–23	→	SD103	Hour
	D14	Minute	0–59	→	SD104	Minute
	D15	Second	0–59	→	SD105	Second
	D16	Week	0–6	→	SD106	Week

- Function description

You can use the TWR instruction to correct the system time when the system time is different from the actual time.

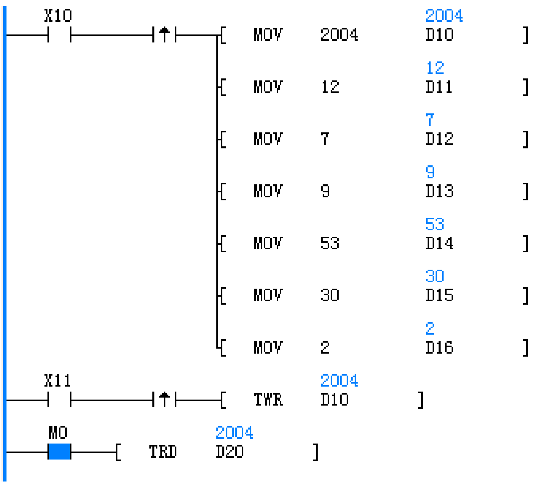
- Note

1. The written time data needs touse the solar calendar, otherwise the instruction cannot be executed successfully.

2. It is suggested touse the edge trigger as the execution condition of this instruction.

- Application instance

Changing the system time with the TWR instruction, as shown in the following figure.




```
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. Writing the time set values into 7 consecutive units starting from D10 upon detecting the rising edge of X10.

2. Writing the values of 7 consecutive units starting from D10 into the system time upon detecting the rising edge of X11.

3. Reading the system time and storing it in D20 when M0 is ON.

6.9.3 TADD: Clock addition instruction

LAD: TADD (S1) (S2) (D)]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	**Zero flag SM180, carry flag SM181**
IL: TADD *(S1)* *(S2)* *(D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD								D	SD			V		R	√
S2	WORD								D	SD			V		R	√
D	WORD								D				V		R	√

- Operand description

  S1: Clock data 1

  Storing the time data in the three storage units assigned by ***S1***. If the data is not compliant with the time format, the system reports an error, indicating that the value of the instruction operand is invalid.

  S2: Clock data 2

  Storing another time data in the three storage units assigned by ***S2***. If the data is not compliant with the time format, the system reports an error, indicating that the value of the instruction operand is invalid.

  D: Storage unit of time result

  Data obtained after the time addition operation is stored in the three storage units assigned by ***D***. The result affects the carry flag SM181 and the zero flag SM180.

- Function description

  Performing the addition operation on the data with the time format, and executing the operation rule in the time format.

- Note

  The time data in the operation need to conform to the time format:

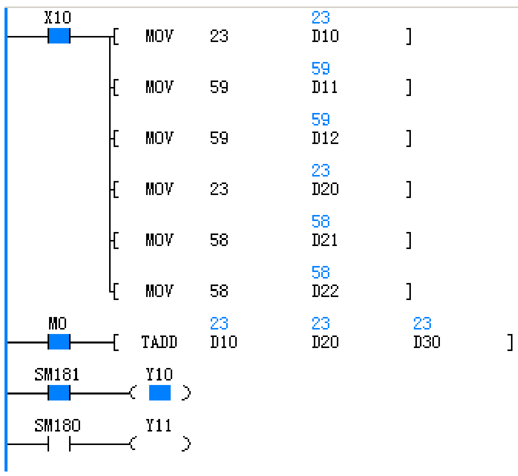
  Setting range of"Hour": 0﹣23.

  Setting range of"Minute": 0﹣59

  Setting range of"Second": 0﹣59

- Application instance

S1			S2			D	
D10	23 hours	+	D20	23 hours	=	D30	23 hours
D11	59 minutes		D21	58 minutes		D31	58 minutes
D12	59 seconds		D22	58 seconds		D32	57 seconds




```
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    SM181
OUT   Y10
LD    SM180
OUT   Y11
```

1. Sending the time data to three storage units starting from D10 and three storage units starting from D20 when X10 is ON.
2. Increasing the three storage units starting from D10 by the three storage units starting from D20, and storing the obtained result in the three storage units starting from D30 when M0 is ON.
3. Setting the carry flag (SM181) to ON and the zero flag (SM180) to OFF.

6.9.4 TSUB: Clock subtraction instruction

LAD: TSUB (S1) (S2) (D)]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	**Zero flag SM180, carry flag SM182**
IL: TADD *(S1) (S2) (D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD								D	SD			V		R	√
S2	WORD								D	SD			V		R	√
D	WORD								D				V		R	√

- Operand description

S1: Clock data 1

Storing the time data in the three storage units assigned by ***S1***. If the data is not compliant with the time format, the system reports an error, indicating that the value of the instruction operand is invalid.

S2: Clock data 2

Storing another time data in the three storage units assigned by ***S2***. If the data is not compliant with the time format, the system reports an error, indicating that the value of the instruction operand is invalid.

D: Storage unit of time result

Data obtained after the time addition operation is stored in the three storage units assigned by ***D***. The result affects the carry flag SM181 and the zero flag SM180.

- Function description

Performing the subtraction operation on the data with the time format, and executing the operation rule in the time format.

- Note

The time data in the operation need toconform to the time format:

Setting range of "Hour": 0 - 23

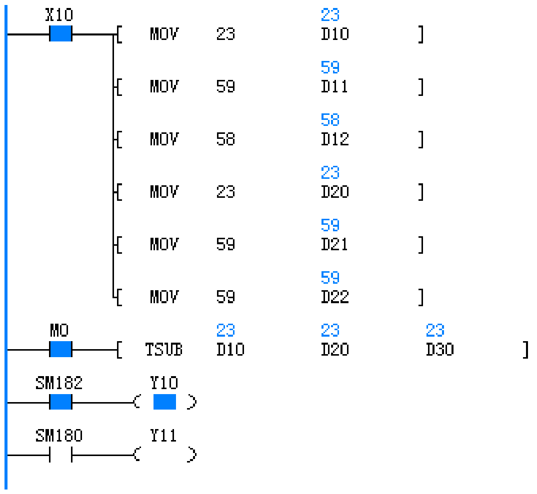
Setting range of "Minute": 0 - 59

Setting range of "Second": 0 - 59

- Application instance

S1			S2			D	
D10	23 hour	—	D20	23 hour	=	D30	23 hour
D11	59 minute		D21	59 minute		D31	59 minute
D12	58 second		D22	59 second		D32	59 second

- Application instance




```
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   SM182
OUT  Y10
LD   SM180
OUT  Y11
```

1. Sending the time data to the three storage units starting from D10 and the three storage units starting from D20 when X10 is ON.

2. Subtracting the three storage units starting from D10 by the three storage units starting from D20, and storing the obtained result in the three storage units starting from D30 when M0 is ON.

3. Setting the carry flag (SM182) to ON and the zero flag (SM180) to OFF.

6.9.5 HOUR: Chronograph instruction

LAD: HOUR (S) (D1) (D2)		**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
		Influenced flag bit	
IL: HOUR *(S) (D1) (D2)*		**Step length**	**8**

Operand	Type	Applicable soft element														Indexing
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: Hour comparison data. Data range: 0－32767

D1: Time storage unit

The data unit of ***D1*** stores hours while the data unit of ***D1***＋1 stores seconds.

D2: Alarm output address

When the data of ***D1*** is greater than or equal to the data designated by ***S***, the alarm point is changed to ON output.

- Function description

Making the judgment on the time when the input contact is in the ON state (unit: hour).

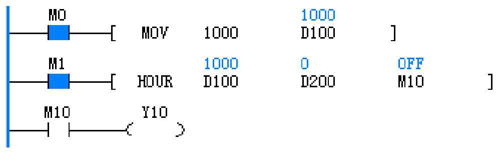
- Note

1. Designate ***D1*** as the soft element unit for power-off storage to ensure present data can be used after PLC power is cut off. If normal soft element unit is used, current data will be cleared when PLC power is cut off or RUN → STOP operation is being carried out.

2. Even if the alarm output ***D2*** is ON, it can continue counting.

3. This instruction hour is 16-bit integer data. When the hour data is larger than 32767, starts from 0 again.

- Application instance



LD M0
MOV 1000 D100
LD M1
HOUR D100 D200 M10
LD M10
OUT Y10

1. Setting the comparison data of the HOUR instruction when M0 is ON.

2. HOUR performs the time accumulation operation on the input contacts when M1 is ON.

3. When the accumulation time, that is, time of keeping M1 in the ON state, is greater than or equal to 1000, M10 is in the ON state.

6.9.6 DCMP: (=、<、>、<>、>=、<=)Date comparison instruction

LAD: ├─┤ ├─[DCMP= *(S1)* *(S2)* *(D)*] ├─┤ ├─[DCMP< *(S1)* *(S2)* *(D)*] ├─┤ ├─[DCMP> *(S1)* *(S2)* *(D)*] ├─┤ ├─[DCMP<> *(S1)* *(S2)* *(D)*] ├─┤ ├─[DCMP>= *(S1)* *(S2)* *(D)*] ├─┤ ├─[DCMP<= *(S1)* *(S2)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	
IL: **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 length**	**7**

Operand	Type	Applicable soft element														Indexing
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, which occupies the start three word units designated by ***S1***. Data of these three units need to comply with the format of solar calendar, otherwise the system reports an operand error.

  S2: Date comparison data 2, which occupies the start three word units designated by ***S2***. Data of these three units need to comply with the format of solar calendar, otherwise the system reports an operand error.

  D: Comparison state output. When the data comply with the comparison condition, setting D to ON, otherwise, it is OFF.

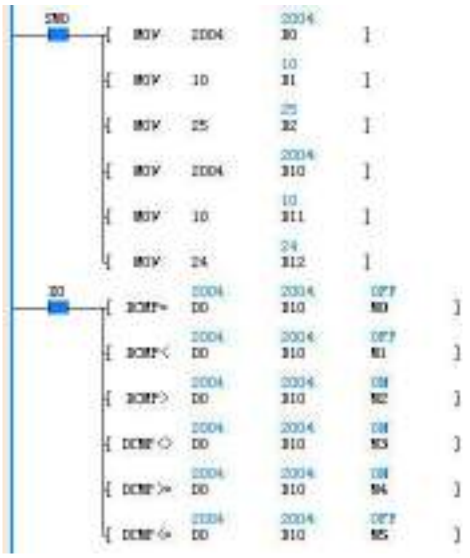
- Function description

  Conducting the BIN comparison on the date data starting from ***S1*** and ***S2*** respectively, and assigning the comparisonresult to ***D***.

- Note

  The date data starting from ***S1*** and ***S2*** need to comply with the format of the solar calendar, otherwise,the system reports an operand error(eg: 2004, 9, 31 and 2003, 2,29 are illegal).

- Application instance



```
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
```

Conducting the BIN comparison on the date data starting from D0 and D10 respectively, and assigning the comparison result to the destination data (M0, etc.).

6.9.7 TCMP: (=、<、>、<>、>=、<=)Time comparison instruction

LAD:		
├──┤ ├──[TCMP= (S1) (S2) (D)] ├──┤ ├──[TCMP< (S1) (S2) (D)] ├──┤ ├──[TCMP> (S1) (S2) (D)] ├──┤ ├──[TCMP<> (S1) (S2) (D)] ├──┤ ├──[TCMP>= (S1) (S2) (D)] ├──┤ ├──[TCMP<= (S1) (S2) (D)]	**Applicable model** **Influenced flag bit**	**IVC2L IVC1IVC3 IVC1L**
IL: **TCMP＝ (*S1*) (*S2*) (*D*)** **TCMP＜ (*S1*) (*S2*) (*D*)** **TCMP＞ (*S1*) (*S2*) (*D*)** **TCMP＜＞ (*S1*) (*S2*) (*D*)** **TCMP＞＝ (*S1*) (*S2*) (*D*)** **TCMP＜＝ (*S1*) (*S2*) (*D*)**	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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, which occupies the start three word units designated by ***S1***. Data of these three units need to comply with the 24-hour time format, otherwise the system reports an operand error.

  S2: Time comparison data 2, which occupies the start three word units designated by ***S2***. Data of these three units need to comply with the 24-hour time format, otherwise the system reports an operand error.

  D: Comparison state output. When the data comply with the comparison condition, setting D to ON, otherwise, it is OFF.

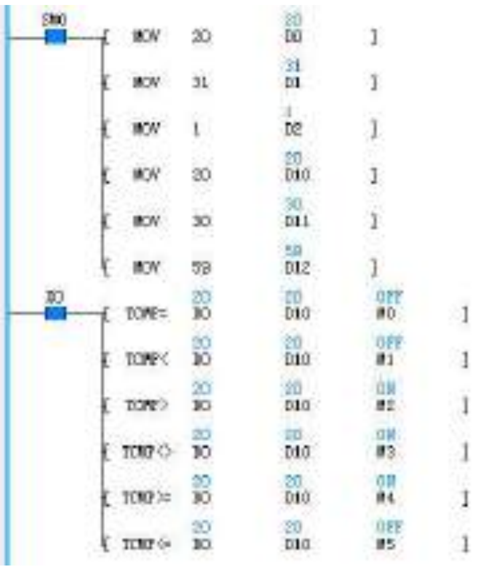
- Function description

  Conducting the BIN comparison on the time data starting from***S1*** and ***S2*** respectively, and assigning the comparisonresult to ***D***.

- Note

  The time data starting from***S1*** and ***S2*** need to comply with the 24-hour time format, otherwise,the system reports an operand error(eg: 24, 10, 31 and 13, 59, 60 are illegal).

- Application instance



```
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
```

Conducting the BIN comparison on the time data starting from D0 and D10 respectively, and assigning the comparison result to the destination data (M0, etc.).

6.9.8 HTOS: Instruction for converting hour-minute-second data to seconds

LAD: ─┤ ├─[HTOS (S) (D)]														
Applicable model	IVC3													
Influenced flag bit														
IL: HTOS (S) (D)														
Step length	5													
Operand	Type	Applicable soft element												Indexing
S	WORD		KnX	KnY	KnM	KnS	T	C	D	SD			R	√
D	WORD			KnY	KnM	KnS	T	C	D	SD			R	√

- Operand description

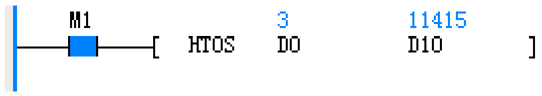
  S: Number of the start soft element that stores the time data to be converted.

  D: Number of the soft element that stores the converted time data.

- Function description

  Converting the time data (hours, minutes, and seconds) of S-S+2 into seconds, and storing the obtained result in D.

- Application instance



LD M1
HTOS D0 D10

1. Converting the time data (hours, minutes, and seconds) starting from D0 into seconds, and storing the obtained resultin D10 when M1=ON.D10=11415 when D0=3, D1=10 and D2=15.

6.9.9 STOH: Instruction for converting seconds to hour-minute-second data

LAD: ─┤ ├─[STOH (S) (D)]														
Applicable model	IVC3													
Influenced flag bit														
IL: STOH (S) (D)														
Step length	5													
Operand	Type	Applicable soft element												Indexing
S	WORD		KnX	KnY	KnM	KnS	T	C	D	SD			R	√
D	WORD			KnY	KnM	KnS	T	C	D	SD			R	√

- Operand description

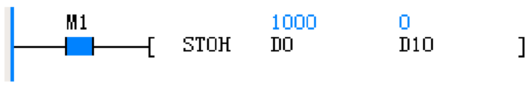
  S: Number of the soft element that stores the time data to be converted.

  D: Number of the start soft elementthat stores the converted time data.

- Function description

  Converting the second data of S into hours, minutes, and seconds, and storing the obtained result in D, D+1, and D+2 respectively.

- Note
- Application instance



LD M1
STOH D0 D10

1. Converting the second data of D0 into the time data (hours, minutes, and seconds), and storing the obtained result in three units starting from D10 when M1=ON.D10=0, D11=16, and D12=40 when D0=1000.

6.10 High-speed I/O instructions

6.10.1 HCNT: Instruction for driving the high-speed counter

LAD: ├─┤ ├──[HCNT (D) (S)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: **HCNT** ***(D)*** ***(S)***	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
D	DINT										C					
S	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√

- Operand description

D: Assigning the counter number. Settable range: C236－C255, C300－C306.

S: Assigning the comparison constant, a 32-bit data with symbol. The data range: -2147483648 － +2147483647.

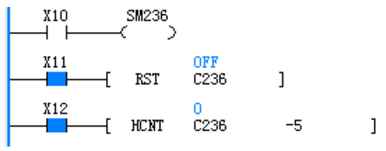
- Function description

Driving the designated hardware high-speed counter. Performing the hardware high-speed counting on all high-speed counters only when it is continuously driven, and judging the action of NO contacts of the high-speed counter according to ***S***.

- Note

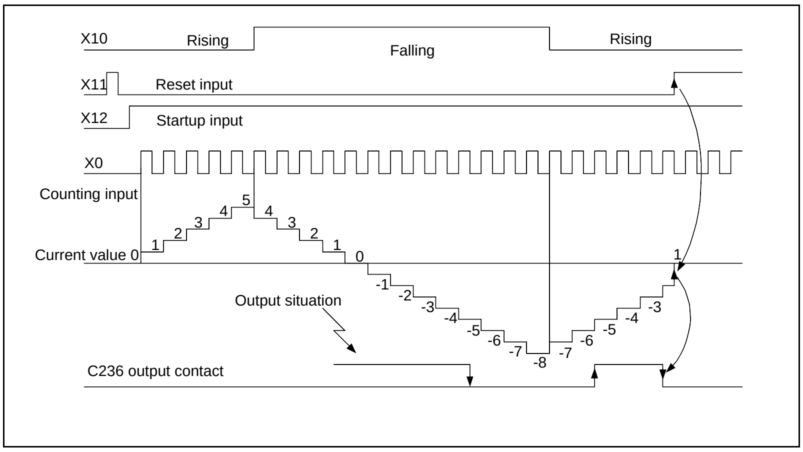
There are hardware conflicts between the HCNT instruction, SPD instruction, external input interrupt, and pulse capture. You need to note that the use conditions of all high-speed I/O. Refer to Chapter 8"Operating guide for high-speed input function" for use.

- Application instance




```
LD    X10
OUT   SM236
LD    X11
RST   C236
LD    X12
HCNT  C236  -5
```

The time sequence operation of the program operation is shown below:



1. Initializing the hardware counter corresponding to C236 when X12 changes from OFF to ON. X0 is the pulse input end of C236 while C236 counts the external pulse of X0. X0 is a general input point when X12 is OFF, and C236 cannot count the external pulse of X0.

2. Action on the contacts: Setting the contact of C236 when the current value of the counter C236 is increased from -6 to -5. Resetting the contact of C236 when the current value of counter C236 is decreased from -5 to -6.

3. Executing the RST instruction when X11 is ON, clearing the C236 data, and disconnecting the C236 contact.

4. When a power failure occurs, you can set the data and contact state of the high-speed counter in the system block of the upper software.

6.10.2 DHSCS: Instruction of setting the high-speed count comparison

LAD: ├─┤ ├──[DHSCS (S1) (S2) (D)]	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: **DHSCS** ***(S1)*** ***(S2)*** ***(D)***	**Step length**	**10**

Operand	Type	Applicable soft element	Indexing

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: Data to be compared by the high-speed counter, 32-bit DINT data, ranging from -2147483648 to +2147483647.

S2: High-speed counter. The applicable range: C236－C255 and C300－306.

D: Bit element output object, immediately setting the output of Y, M, and S regardless of the scan period.

- Note

1. The DHSCS instruction needs to work together with the HCNT instruction. DHSCS can be executed correctly only when the high-speed counter is driven by HCNT.

2. The result compared by the DHSCS instruction takes effect when external pulses of the high-speed counter are inputted. Therefore, the DHSCS instruction does not generate any action even if the high-speed counter value is changed through the DMOV or MOV instruction.

3. Like general instructions, multiple DHSCS (DHSCI, DHSCR, DHSZ, DHSP, and DHST) instructions can be used simultaneously, but the total number of these instructions cannot exceed 6. The first 6 instructions are executed in order, and the 7th or later instructions are not effective and therefore not executed.

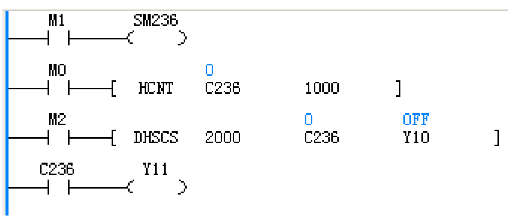
4. The max. frequency supported by the high-speed counter of the PLCs. If you use the DHSCS, DHSCI, DHSCR, DHSZ, DHSP, or the DHST instruction, it is limited by the max. response frequency and comprehensive frequency. For details, refer to Chapter 8"Operating guide for high-speed input function".

- Function description

1. A high-speed counter counts in interrupt mode only when it is driven by the HCNT instruction and the counting input changes from OFF to ON. When the value of the high-speed counter is equal to ***S1*** in the DHSCS instruction, the bit element designated by ***D*** is set immediately, or in the case of a Y element, the Y element is outputted immediately.

2. This instruction can be used to immediately output the comparison result based on the comparison setting for the current value of the high-speed counter.

- Application instance




```
LD     M1
OUT    SM236
LD     M0
HCNT C236 1000
LD     M2
DHSCS 2000 C236 Y10
LD     C236
OUT    Y11
```

1. When M1 is ON, C236 counts in the interrupt mode when X0 changes from OFF to ON (for the input frequency of X0, refer to the instruction for high-speed I/O). When C236 changes from 999 to 1000, the C236 contact is set, and when C236 changes from 1001 to 1000, the C236 contact is reset. When Y11 is driven by C236, the execution of Y11 is determined by the scan cycle of the user program.

2. When M2 is ON and the DHSCS high-speed instruction meets the requirements stated in the preceding "Note", Y10 is outputted immediately if C236 reaches 2000, regardless of the scan cycle.

3. When M0 is ON, SM236 is driven, and the C236 counter counts down. When M0 is OFF, SM236 is not driven, and the C236 counter counts up.

6.10.3 DHSCI: Instruction for triggering interrupt based on comparison of high-speed count

LAD: DHSCI (S1) (S2) (S3)		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: DHSCI *(S1)* *(S2)* *(S3)*		**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT										C					
S3	WORD	Constant														

- Operand description

S1: Data to be compared by the high-speed counter, 32-bit DINT data, ranging from -2147483648 to +2147483647.

S2: High-speed counter. The applicable range: C236－C255 and C300－306.

S3: Interrupt number. Range of the interrupt number: 20－25, 40, and 41.

- Function description

A high-speed counter can count in interrupt mode only when it is driven by the HCNT instruction and the counting input changes from OFF to ON. When the value of the high-speed counter is equal to ***S1*** in the DHSCS instruction, it enters into the interrupt subprogram designated by ***S3***. You can write a program to be executed immediately in the interrupt subprogram.

- Note

1. The DHSCI instruction needs to worktogether with the HCNT instruction. DHSCI can be executed correctly only when the high-speed counter is driven by HCNT.

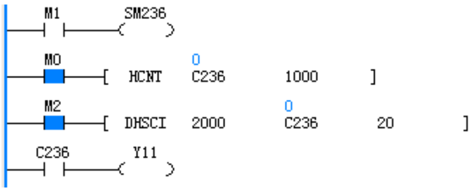
2. The result compared by the DHSCI instruction takes effect when external pulses of the high-speed counter are inputted. Therefore, the DHSCI instruction does not generate any action even if the high-speed counter value is changed through the DMOV or MOV instruction.

3. Like general instructions, multiple DHSCI (DHSCS, DHSCR, DHSZ, DHSP, and DHST) instructions can be used simultaneously, but the total number of these instructions cannot exceed 6.The first 6 instructions are executed in order, and the 7th or later instructions are not effective and therefore not executed.

4. The max. frequency supported by the high-speed counter of the PLCs. If you use the DHSCS, DHSCI, DHSCR, DHSZ, DHSP, or the DHST instruction, it is limited by the max. response frequency and comprehensive frequency. For details, refer to Chapter 8"Operating guide for high-speed input function".

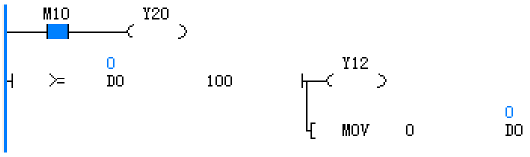
- Application instance

The User main program is shown as below:




```
LD      M1
OUT     SM236
LD      M0
DHSCI   2000   C236   20
LD      C236
OUT     Y11
```

The interrupt program whose user interrupt No. is 20 is shown as below:




```
LD      M10
OUT     Y20
LD>=    D0   100
OUT     Y12
MOV     0    D0
```

1. When M1 is ON, C236 counts in the interrupt mode when X0 changes from OFF to ON (for the input frequency of X0, refer to the instruction for high-speed I/O). When C236 changes from 999 to 1000, the C236 contact is set, and when C236 changes from 1001 to 1000, the C236 contact is reset. When Y11 is driven by C236, the execution of Y11 is determined by the scan cycle of the user program.

2. When M2 is ON and the DHSCI high-speed instruction meets the requirements stated in the preceding "Note", the interrupt subprogram whose interrupt No. is 20 responds immediately and executes the user program in the interrupt program when C236 reaches 2000.

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. Entering into the No.20 interrupt program when C236 has the pulse input and C236 is 2000. Driving Y20 when M10 is ON. But the output of Y20 is related to the scan cycle of the user program. Meanwhile, driving Y12 and clearing the data of D0 when D0 is detected to be larger than 100.

6.10.4 DHSPI: Instruction for triggering interrupt based on comparison of absolute high-speed output positions

LAD: DHSPI *(S1)* *(S2)* *(S3)*	**Applicable model**	**IVC3**
	Influenced flag bit	
IL: DHSPI *(S1)* *(S2)* *(S3)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DINT									SD						
S3	WORD	Constant														

- Operand description

  S1: Data to be compared by the high-speed output position element, 32-bit DINT data, ranging from -2147483648 to +2147483647.

  S2: High-speed output position element. The applicable range: SD200,SD210,SD320,SD330,SD340,SD350,SD360, and SD370.

  S3: Interrupt number. Range of the interrupt number: 53,54,55,56,57,58,59, and 60.

- Function description

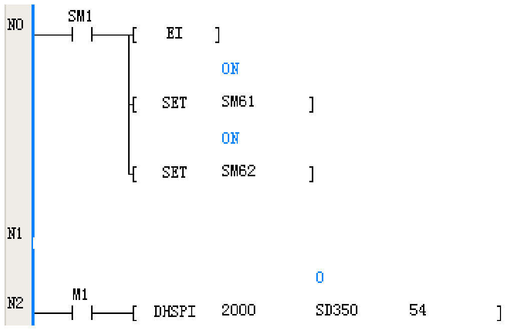
  When the value of the high-speed output position element is equal to ***S1*** in the DHSPI instruction, it enters into the interrupt subprogram designated by ***S3***. You can write a program to be executed immediately in the interrupt subprogram.

- Note

  1. When data is written in an SD element, no position-based interrupt is triggered. After the data is written, position-based interrupt is triggered by a specified position.

- Application instance

  The User main program is shown as below:



The interrupt number of the interrupt subprogram can be set to 54 or other position-based interrupt source of high-speed output, and then the program to be executed when passing the position can be written in the interrupt subprogram.

6.10.5 DHSCR: Instruction for resetting the high-speed count comparison

LAD: DHSCR (S1) (S2) (D)	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: DHSCR (S1) (S2) (D)	Step length	10

Operand	Type	Applicable soft element														Indexing
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: Data to be compared by the high-speed counter, 32-bit DINT data, ranging from -2147483648 to +2147483647.

  S2: High-speed counter. The applicable range: C236－C255.

  D: Bit element output object, immediately resetting the output of Y, M, S, and C regardless of the scan period. If D is a C element, it shall be ***S2***.

- Function description

  A high-speed counter can count in interrupt mode only when it is driven by the HCNT instruction and the counting input changes from OFF to ON. When the value of the high-speed counter is equal to ***S1*** in the DHSCR instruction, the bit element designated by ***D*** is reset immediately, or in the case of a Y element, the Y element is outputted immediately.2. This instruction can be used to immediately output the comparison result based on the comparison setting for the current value of the high-speed counter.

- Note

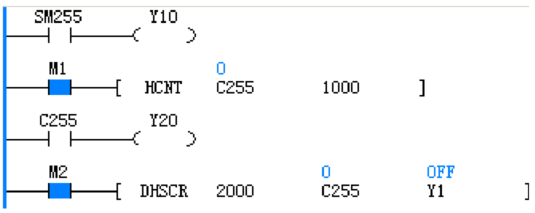
  1. The DHSCR instruction needs to work together with the HCNT instruction. DHSCR can be executed correctly only when the high-speed counter is driven by HCNT.

  2. The result compared by the DHSCR instruction takes effect when external pulses of the high-speed counter are inputted. Therefore, the DHSCR instruction does not generate any action even if the high-speed counter value is changed through the DMOV or MOV instruction.

  3. Like general instructions, multiple DHSCR (DHSCI, DHSCS, DHSZ, DHSP, and DHST) instructions can be used simultaneously, but the total number of these instructions cannot exceed 6.The first 6 instructions are executed in order, and the 7th or later instructions are not effective and therefore not executed.

  4. The max. frequency supported by the high-speed counter of the PLCs. If you use the DHSCS, DHSCI, DHSCR, DHSZ, DHSP, or DHST instruction, it is limited by the max. response frequency and comprehensive frequency. For details, refer toChapter 8"Operating guide for high-speed input function".

- Application instance




```
LD     SM255
OUT    Y10
LD     M1
HCNT   C255   1000
LD     C255
OUT    Y20
LD     M2
DHSCR  2000   C255   Y1
```

1. When M1 and X7 are both ON, C255 counts the phase difference of X3 and X4 in the interrupt mode (for the input frequency of X0, refer to the instruction for high-speed I/O). When C255 changes from 999 to 1000, the C255 contact is set, and when C255 changes from 1001 to 1000, the C255 contact is reset. When Y20 is driven by C255, the execution of Y20 is determined by the scan cycle of the user program.

2. When M2 is ON and the DHSCR high-speed instruction meets the requirement stated in the preceding "Note", Y1 is outputted immediately if C255 reaches 2000, regardless of the scan cycle.

3. When the input pulse of X3 is ahead of the one of X4, SM255 is ON; when the input pulse of X4 is ahead of the one 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 both ON, if X5 is ON, the C255 counter is cleared and the C255 auxiliary contact is also cleared.

6.10.6 DHSZ: High-speed count range comparison instruction

LAD: DHSZ (S1) (S2) (S3) (D)		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: DHSZ *(S1) (S2) (S3) (D)*		**Step length**	**13**

Operand	Type	Applicable soft element													Indexing
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: Data 1 to be compared by the high-speed counter, 32-bit DINT data, ranging from -2147483648 to +2147483647.

S2: Data 2 to be compared by the high-speed counter,32-bit DINT data, ranging from 2147483648 to +2147483647.

S3: High-speed counter. The applicable range: C236 - C255.

D: Bit element output object, immediately setting the output of Y, M, and S regardless of the scan period.

- Function description

1. A high-speed counter can count in interrupt mode only when it is driven by the HCNT instruction and the counting input changes from OFF to ON.

2. Setting the bit element designated by ***D***, and resetting the two bit elements following the one designated by ***D*** when the value of the high-speed counter is less than ***S1*** in the instruction.

3. Resetting the bit element designated by ***D***, and setting the two bit elements following the one designated by ***D*** when the value of the high-speed counter is greater than or equal to ***S1*** and less than or equal to ***S2***.

4. Resetting the bit element designated by ***D***, and resetting the two bit elements following the one designated by ***D*** when the value of high-speed counter is larger than ***S***2 in the DHSZ instruction.

5. If it is Y element, Y element outputs the corresponding state immediately, and the output action is irrelevant to the scan cycle of the program.

- Note

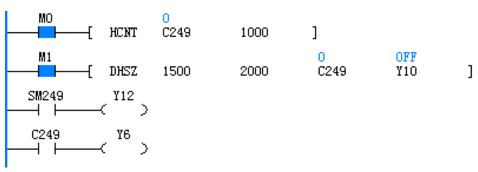
1. The DHSZ instruction needs to worktogether with the HCNT instruction. DHSZ can be executed correctly only when the high-speed counter is driven by HCNT.

2. The result compared by the DHSZ instruction takes effect when external pulses of the high-speed counter are inputted. Therefore, the DHSZ instruction does not generate any action even if the high-speed counter value is changed through the DMOV or MOV instruction.

3. Like general instructions, multiple DHSCZ (DHSCI, DHSCS, DHSCR, DHSP, and DHST) instructions can be used simultaneously, but the total number of these instructions cannot exceed 6.The first 6 instructions are executed in order, and the 7th or later instructions are not effective and therefore not executed.

4. The max. frequency supported by the high-speed counter of the PLCs. If you use the DHSCS, DHSCI, DHSCR, DHSZ, DHSP, or DHST instruction, it is limited by the max. response frequency and comprehensive frequency. For details, refer toChapter 8"Operating guide for high-speed input function".

- Application instance




```
LD    M0
HCNT  C249   1000
LD    M1
DHSZ  1500 2000 C249   Y10
LD    SM249
OUT   Y12
LD    C249
OUT   Y6
```

1. When M0 and X6 are both ON, C249 counts up when X0 changes from OFF to ON, or counts down when X1 changes from OFF to ON (for the input frequency of X0, refer to the instruction for high-speed I/O). When C249 changes from 999 to 1000, the C249 contact is set and when C249 changes from 1001 to 1000, the C249 contact is reset. When Y6 is driven by C249, the execution of Y6 is determined by the scan cycle of the user program.

2. When M1 is ON and the DHSZ high-speed instruction meets the requirements stated in the preceding "Note", the state of Y10, Y11, and Y12 are as below:

(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 cycle.

3. When M0 and X6 are both ON, if X0 counts up from OFF to ON,SM249 is reset. If X1 counts down from OFF to ON, SM249 is set.

4. When X6 is OFF, the C249 counter cannot count.

5. When M0 and X6 are both ON, if X2 is ON, the C249 counter is cleared and the C249 auxiliary contact is also cleared.

6.10.7 DHST: High-speed count table comparison instruction

LAD: DHST (S1) (S2) (S3)	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: DHST *(S1)* *(S2)* *(S3)*	**Step length**	**10**

Operand	Type	Applicable soft element														Indexing
S1	DINT								D						R	
S2	INT	Constant														
S3	DINT										C					

- Operand description

S1: Start unit of the data forthe table comparison (Start number of D element). The subsequent three D elements with consecutive serial numbers are used to specify the data to be compared by the high-speed counter, the serial number and corresponding output state of the Y element. The D elements connected to these four consecutive serial numbers are collectively referred to beone record.

S2: Number of records to be compared. The data range: 1 – 128

S3: High-speed counter. The applicable range: C236 – C255 and C301 – 306.

- Function description

1. A high-speed counter can count in interrupt mode only when it is driven by the HCNT instruction and the counting input changes from OFF to ON.

2. When the value of the high-speed counter is equal to the data to be compared currently, the corresponding Y element state is output according to the recorded data, and the output object can be the Y element only.

3. The output action is irrelevant to the scan cycle, and the Y element designated by the current record outputs the designated state immediately.

4. If it is desired that the the user program executesthe immediate output operation according to the comparison data designated by a certain table and Y elements, you can use the DHST instruction.

- Note

1. The DHST instruction needs to work together with the HCNT instruction. DHST can be executed correctly only when the high-speed counter is driven by HCNT.

2. The result compared by the DHST instruction takes effect when external pulses of the high-speed counter are inputted. Therefore, the DHST instruction does not generate any action even if the high-speed counter value is changed through the DMOV or MOV instruction.

3. Like general instructions, multiple DHST (DHSCI, DHSCS, DHSCR, DHSP, and DHSZ) instructions can be used simultaneously, but the total number of these instructions cannot exceed 6.The first 6 instructions are executed in order, and the 7th or later instructions are not effective and therefore not executed. If DHSP is a valid instruction in the user's instruction, DHST is not executed. On the contrary, if DHST is a valid instruction,DHSP is not executed. Only one instruction (DHST or DHSP) can be valid at the same time in the user program.

4. The max. frequency supported by the high-speed counter of the PLCs. If you use the DHSCS, DHSCI, DHSCR, DHSZ, DHSP, or DHST instruction, it is limited by the max. response frequency and comprehensive frequency. For details, refer to Chapter 8"Operating guide for high-speed input function".

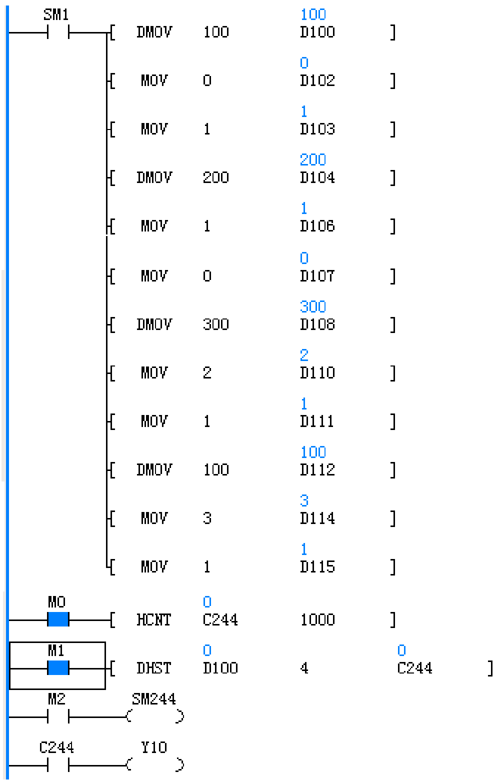
- Application instance

The table data is shown as below:

Comparison data		Output Y number	Set/Reset	Operation process
MSB	LSB			

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↓ Return to 1

The LAD is shown as below:




```
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. Assigning the initial value to D100－D115, and generating the table to be compared in the first scan cycle of the user program.

2. When M0 and X6 are both ON, C244 counts when X0 changes from OFF to ON (for the input frequency of X0, refer to the instruction for high-speed I/O). When C244 changes from 999 to 1000, the C244 is set and when C244 changes from 1001 to 1000, the C244 is reset. When Y10 is driven by C244, the execution of Y10 is determined by the scan cycle of the user program.

3. When M1 is ON and the DHST high-speed instruction meets the requirements stated in the preceding "Note", the DHST high-speed instructionstarts from No.1 record of the table and enters the comparison of No.2 record only after No. 1 record is completed. The comparison of the next record can be entered onlyafter the previous instruction is completed. When the comparison of the last record is completed, it returns to the comparison of the first record again and sets SM185. SD184 indicates the record No. to be compared currently while SD182 and SD183 indicate the data to be compared currently. The comparison result is outputted immediately, regardless of the scan cycle.

4. When M2 is ON, SM244 is ON, and C244 counts down. If M2 is OFF, SM244 is OFF, and C244 counts up.

5. When X6 is OFF, C244 counter cannot count.

6. When M0 and X6 are both ON, if X2 is ON, C244 counter is cleared while C244 auxiliary contact is also cleared.

6.10.8 DHSP: Instruction for pulse output based on high-speed count table comparison

LAD: DHSP (S1) (S2) (S3)		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: DHSP *(S1) (S2) (S3)*		**Step length**	**10**

Operand	Type	Applicable soft element													Indexing
S1	DINT								D					R	
S2	INT	Constant													
S3	DINT										C				

- Operand description

S1: Start unit of the data forthe table comparison (Start number of D element). The subsequent three D elements with consecutive serial numbers are used to specify the data to be compared by the high-speed counter, and the data outputted to SD180 and SD181. The D elementsconnected to these four consecutive serial numbers are collectively referred to beone record.

S2: Number of records to be compared. The data range: 1－128.

S3: High-speed counter. The applicable range: C236－C255.

- Function description

1. A high-speed counter can count in interrupt mode only when it is driven by the HCNT instruction and the counting input changes from OFF to ON.

2. When the value of the high-speed counter is equal to the comparison data of the current record, SD180 and SD181 are modified according to output data of current data.

3. If it is needed to make user program determine high speed output or other data assignment value according to a certain table, use DHSP table comparison instruction. For instance, user can assign SD180 and SD181 (double word) as the output frequency operand of PLSY instruction to make PLSY output frequency adjust according to table comparison result.

3. You can use the DHSP instruction when you hope the user program to determine the assignment of high-speed output or other data based on a certain table. For example, SD180 and SD181 (double word) can be specified as the output frequency operands of the PLSY instruction, so as to adjust the PLSY output frequency according to the table comparison result.

- Note

1. The DHSP instruction needs to work together with the HCNT instruction. DHSP can be executed correctly only when the high-speed counter is driven by HCNT.

2. When DHSP is used in conjunction with PLSY, the data transmitted to SD180 and SD 181 need to meet the frequency output requirement of PLSY. For details, refer to the PLSY instruction.

3. If you want to stop the comparison at the last row, you need to set the data sent to SD180 and SD181 by the last cell of the comparison table to 0. In this case, the other DHST and DHSP instructions are invalid, but the DHSP instruction is not counted in the total instruction number of other high speed instructions.

4. The result compared by the DHSP instruction takes effect when external pulses of the high-speed counter are inputted. Therefore, the DHSP instruction does not generate any action even if the high-speed counter value is changed through the DMOV or MOV instruction.

5. Like general instructions, multiple DHSP (DHSCI, DHSCS, DHSCR, DHST, and DHSZ) instructions can be used simultaneously, but the total number of these instructions cannot exceed 6.The first 6 instructions are executed in order, and the 7^{th} or later instructions are not effective and therefore not executed.

6. If DHSP is a valid instruction in the user instruction, DHST is not executed, on the contrary, if DHST is a valid instruction, DHSP is not executed. Only one instruction (DHST or DHSP) can be valid at the same time in the user program, and the others are invalid.

7. The max. frequency supported by the high-speed counter of the PLCs. If you use the DHSCS, DHSCI, DHSCR, DHSZ, DHSP, or DHST instruction, it is limited by the max. response frequency and comprehensive frequency. For details, refer to Chapter 8"Operating guide for high-speed input function".

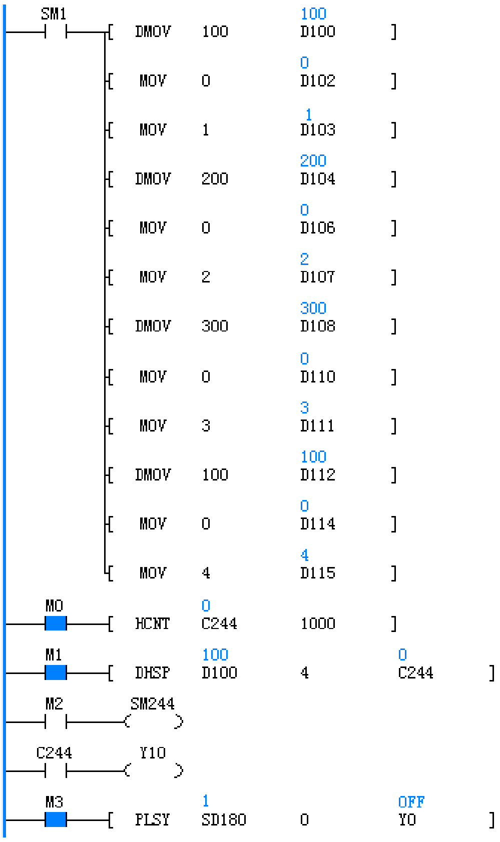
- Application instance

Table data is shown as below:

Comparison data	Data outputted to SD180 and SD181	Operation process

MSB	LSB	MSB	LSB	
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 LAD is shown as below:




```
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    SD180  0   Y0
```

1. Assigning the initial value to D100 - D115, and generating the table to be compared in the first scan cycle of the user program.

2. When M0 and X6 are both ON, C244 counts when X0 changes from OFF to ON (for the input frequency of X0, refer to the instruction for high-speed I/O). When C244 changes from 999 to 1000, the C244 is set and when C244 changes from 1001 to 1000, the C244 is reset. When Y10 is driven by C244, the execution of Y10 is determined by the scan cycle of the user program.

3. When M1 is ON and the DHSP high-speed instruction meets the requirements stated in the preceding "Note", the DHSP high-speed instructionstarts from the record No.1 of the table and enters the comparison of No.2 record only after No. 1 record is completed. The comparison of the next record can be entered onlyafter the previous instruction is completed. When the comparison of last record is completed, it returns to the comparison of first record again and sets SM185. SD184 indicates the record No. to be compared currently while SD182 and SD183 indicate the data to be compared currently. The comparison result is outputted immediately, regardless of the scan cycle. The output operands obtained after the comparison are put into SD180 and SD181, which is not affected by the scan cycle. If you want to stop at the last row in the comparison table, you need to set the data sent to SD180 and SD181 by the final table to 0.

4. When M2 is ON, SM244 is ON, and C244 counts down. If M2 is OFF, SM244 is OFF, and C244 counts up.

5. When X6 is OFF, C244 counter cannot count.

6. When M0 and X6 are both ON, if X2 is ON, C244 counter is cleared while C244 auxiliary contact is also cleared.

6.10.9 SPD: Frequency measuring instruction

LAD:		
SPD (S1) (S2) (D)	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	

IL: SPD (S1) (S2) (D)								Step length	7							
Operand	Type	Applicable soft element														Indexing
S1	BOOL		X													
S2	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD								D				V		R	√

- Operand description

  S1: Input point. Settable range: X0﹣X7

  S2: Time unit for input point detection. Unit: ms. Operand S2>0

  D: Pulse detection data storage unit. When the count value exceeds 65535, an automatic overflow is performed.

- Function description

  Detecting the number of input pulses of X0–X7 within the specified time (ms), and storing the obtained result in the designated soft element unit.

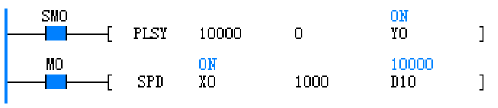
- Note

  1. There is a hardware conflict between SPD and HCNT, external input interrupt, and pulse capture. For details, refer to Chapter 8"Operating guide for high-speed input function" for use.

  2. For IVC1 and IVC2, the input points of SPD are within the range of X0 to X5. For IVC3, the input points of SPD are within the range of X0 to X7.

  3. The max. pulse input frequency of SPD is 10kHz, and an detection error may occur if the input frequency exceeds 10kHz.

- Application instance



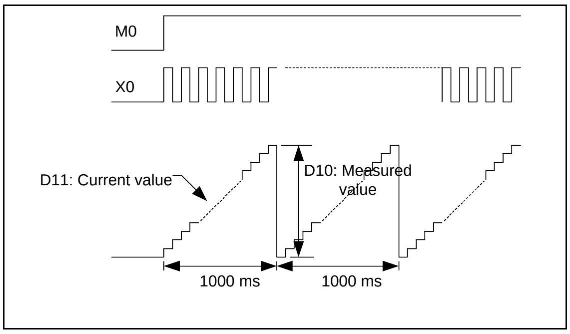
LD SM0

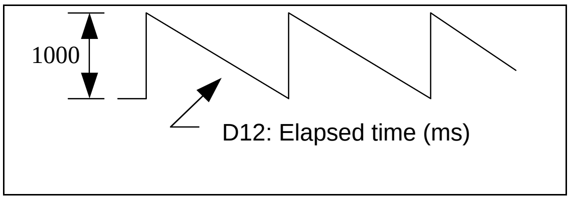
PLSY 10000 0 Y0

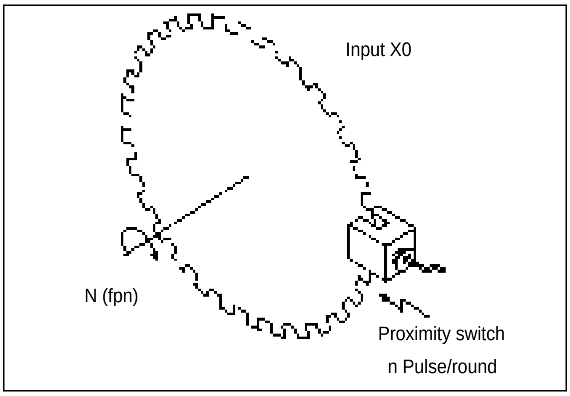
LD M0

SPD X0 1000 D10

The time sequence operation of program is shown as below:







1. Counting the input pulse designated by X0 within 1000ms, and storing the count result to the storage unit of D10 when M0 is ON, in which D11 is the current count value within 1000ms, and D12 is the elapsed time within 1000ms.

2. Data of D10 is proportional to the rotation speed in the above diagram.

3. Counting whenever X0 changes from OFF to ON, and storing the counting values in D10 at every 1000ms.

6.10.10 PLSY: High-speed pulse output instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
PLSY (S1) (S2) (D)	Influenced flag bit	

IL: PLSY (S1) (S2) (D)										**Step length**		**9**			
Operand	Type	Applicable soft element													Indexing
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S2	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	BOOL			Y												

- Operand description

S1: Specified frequency (Hz)

Settable range: IVC1, IVC2: 1－100000 (Hz); IVC1L: Y0 and Y1 can be set from 1–100000 (Hz). Y2 and Y3 can be set from 1－10000; IVC3: 1－200000 (Hz). When ***S1*** is not within the setting range, the system reports that the instruction operand is illegal and no system hardware resource is occupied. When the content of ***S1*** is changed while the instruction is running, the output frequency changes accordingly.

S2: Number of generated pulses (PLS).

Settable range: 0－2147483647. If the set operand is not within the setting range, the system reports that the instruction operand is illegal, the pulse does not output, and no system hardware resource is occupied. When ***S2*** is 0, the pulse always outputs when the instruction is valid.

If the content of ***S2*** is changed while the instruction is running, the change takes effect in the next round of the drive.

D: High-speed pulse output point. For IVC1 and IVC2L, only Y0 or Y1 can be designated; for IVC1L, only Y0, Y1, and Y2 can be designated; for IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

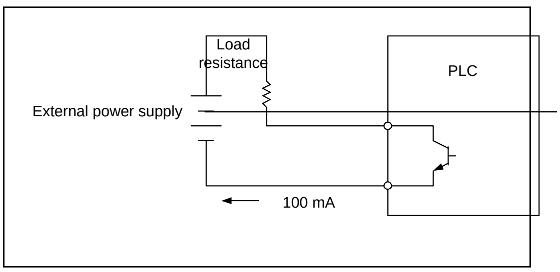
- Function description

Generating a specified amount of the high-speed pulse outputaccording to the frequency specified by the instruction. To output the high-speed pulses, the load current on the output transistor of the PLCs needs to be large enough, but it cannot exceed the rated load current.

- Note

1. The PLCs need to adopt the transistor output mode.2. When the PLCsare executing the high-speed pulse output, it is required to use the load current specified by the PLC output transistor described below.

3. The output circuit (transistor) for PLSY, PWM and PLSR is shown as follows.



4. With the large load current, the OFF time of the transistoris relatively long. When executing the PWM, PLSY and PLSR instructions, it is required the output port of the transist or be connected to the corresponding load. When the output waveform does not conform to the instruction operand, it is viable to increase the load current of the transistor (the transistor load≤100mA).

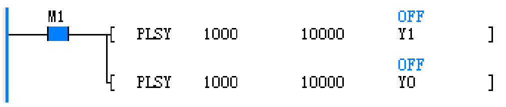
5. When the high-speed instruction is executed effectively (including output completion), other operations on the same portare invalid. Only when the high-speed pulse output instruction is invalid can other instructions operate this port.

6. Using multiple PLSR instructions can obtain separate high-speed pulse output at high-speed output points, or using the PWM (or PLSR) instruction to obtain separate high-speed pulse output at different output points.

7. When multiple PWM, PLSY or PLSR instructions operate on the same port, the first valid instruction controls the output state of the port and the later valid instructions do not affect the state of the output points.

8. Similar to other high-speed instructions (DHSCS, DHSCR,DHSZ, DHSP, DHST, and HCNT), the PLSY instruction needs to meet the requirements for high-speed input and high-speed pulse output in the system.

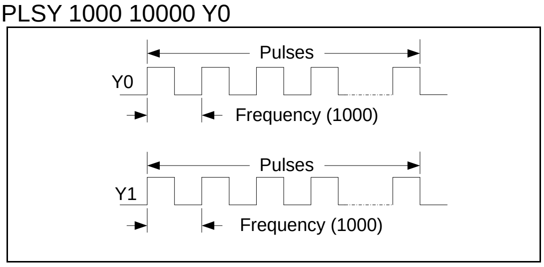
- Application instance



LD M1

PLSY 1000 10000 Y1

PLSY 1000 10000 Y0



1. When M1 is ON, 10000 pulses are output through Y0 and Y1 ports at the frequency of 1000

Hz. and pulse output is stopped after 10000 pulses are outputted. When M1 changes from OFF to ON, the next round of output starts. When M1 is OFF, the port output is OFF.

2. The duty cycle of the pulse is 50% ON, and 50% OFF. The output control is not affected by the scan cycle and is handled by the interrupts. At the high frequency output, the output duty cycle at the Y ports is related to the load. The waveforms obtained from the output terminals (Y0 and PORT0, Y1 and PORT1) are related to the user's output load. When the load cannot exceed the rated load current, the smaller the load, the closer the output waveform is to the set operand.

3. SM80 is used to control the output of Y0 while SM81 is used to control the output of Y1, and when SM80 or SM81 is 1, the output pulse of Y1 is disabled.

4. SM82 and SM83 indicate the output flags of Y0 and Y1 respectively. The flag is cleared when the output is completed or M0 is OFF.

5. SD50 indicates the MSB of the number of the Y0 output pulses in PLSY and PLSR instructions.

SD51 indicates the LSB of the number of the Y0 output pulses in PLSY and PLSR instructions.

SD52 indicates the MSB of the number of the Y1 output pulses in PLSY and PLSR instructions.

SD53 indicates the LSB of the number of the Y1 output pulses in PLSY and PLSR instructions.

SD54 indicates the MSB of the number of the Y0 and Y1 in PLSY and PLSR instructions.

SD55 indicates the LSB of the number of the Y0 and Y1 in PLSY and PLSR instructions.

6. SD50–SD55 can be modified through "DMOV ××× SD5×"or"MOV ××× SD5×", or through the monitoring.

7. If you want to use the number of input pulses to control the output pulse frequency of PLSY, refer to the DHSP instruction for details.

Relevant soft elements:

Address	Name	Action and function	R/W
SM80	Y0 high-speed pulse output control	Y0 high-speed pulse output stop instruction	R/W
SM81	Y1 high-speed pulse output control	Y1 high-speed pulse output stop instruction	R/W
SM82	Y000 pulse output monitoring (busy/ready)	Y0 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM83	Y001 pulse output monitoring (busy/ready)	Y1 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM86	Enabling Y0 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprogram. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow	R/W
SM87	Enabling Y1 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprogram. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow	R/W
SM262	Y002 pulse output stop instruction	When this bit is set, Y002 pulse is disabled	R/W
SM264	Y004 pulse output stop instruction	When this bit is set, Y004 pulse is disabled	R/W
SM265	Y005 pulse output stop instruction	When this bit is set, Y005 pulse is disabled	R/W
SM266	Y006 pulse output stop instruction	When this bit is set, Y006 pulse is disabled	R/W
SM267	Y007 pulse output stop instruction	When this bit is set, Y007 pulse is disabled	R/W
SM272	Y002 pulse output monitoring (busy/ready)	Y002 high-speed pulse output monitoring (when busy, it is ON, when ready it is OFF)	R
SM274	Y004 pulse output monitoring (busy/ready)	Y004 high-speed pulse output monitoring (when busy, it is ON, when ready it is OFF)	R
SM275	Y005 pulse output monitoring (busy/ready)	Y005 high-speed pulse output monitoring (when busy, it is ON, when ready it is OFF)	R
SM276	Y006 pulse output monitoring (busy/ready)	Y006 high-speed pulse output monitoring (when busy, it is ON, when ready it is OFF)	R
SM277	Y007 pulse output monitoring (busy/ready)	Y007 high-speed pulse output monitoring (when busy, it is ON, when ready it is OFF)	R

Address	Action and function	R/W

SD50	Accumulated total number of high-speed pulses outputted through Y0 (MSB) (IVC1L, IVC2L, and IVC3)	R/W
SD51	Accumulated total number of high-speed pulses outputted through Y0 (LSB) (IVC1L, IVC2L, and IVC3)	R/W
SD52	Accumulated total number of high-speed pulses outputted through Y1 (MSB) (IVC1L, IVC2L, and IVC3)	R/W
SD53	Accumulated total number of high-speed pulses outputted through Y1 (LSB) (IVC1L, IVC2L, and IVC3)	R/W
SD54	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (MSB) (IVC2L)	R/W
SD55	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (LSB) (IVC2L)	R/W
SD160	Accumulated total number of high-speed pulses outputted through Y2 (MSB) (IVC1L and IVC3)	R/W
SD161	Accumulated total number of high-speed pulses outputted through Y2 (LSB) (IVC1L and IVC3)	R/W
SD162	Accumulated total number of high-speed pulses outputted through Y3 (MSB) (IVC3)	R/W
SD163	Accumulated total number of high-speed pulses outputted through Y3 (LSB) (IVC3)	R/W
SD164	Accumulated total number of high-speed pulses outputted through Y4 (MSB) (IVC3)	R/W
SD165	Accumulated total number of high-speed pulses outputted through Y4 (LSB) (IVC3)	R/W
SD166	Accumulated total number of high-speed pulses outputted through Y5 (MSB) (IVC3)	R/W
SD167	Accumulated total number of high-speed pulses outputted through Y5 (LSB) (IVC3)	R/W
SD168	Accumulated total number of high-speed pulses outputted through Y6 (MSB) (IVC3)	R/W
SD169	Accumulated total number of high-speed pulses outputted through Y6 (LSB) (IVC3)	R/W
SD170	Accumulated total number of high-speed pulses outputted through Y7 (MSB) (IVC3)	R/W
SD171	Accumulated total number of high-speed pulses outputted through Y7 (LSB) (IVC3)	R/W
SD80	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1)	R/W
SD81	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC1)	R/W
SD82	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC1)	R/W
SD83	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC1)	R/W
SD200	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1L IVC2L, and IVC3)	R/W
SD201	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC1L, IVC2L, and IVC3)	R/W
SD210	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC1L, IVC2L, and IVC3)	R/W
SD211	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC1L, IVC2L, and IVC3)	R/W
SD320	Current position value in the positioning instruction outputted through Y2 (MSB) (IVC1L and IVC3)	R/W
SD321	Current position value in the positioning instruction outputted through Y2 (LSB) (IVC1L and IVC3)	R/W
SD330	Current position value in the positioning instruction outputted through Y3 (MSB) (IVC3)	R/W
SD331	Current position value in the positioning instruction outputted through Y3 (LSB) (IVC3)	R/W
SD340	Current position value in the positioning instruction outputted through Y4 (MSB) (IVC3)	R/W
SD341	Current position value in the positioning instruction outputted through Y4 (LSB) (IVC3)	R/W
SD350	Current position value in the positioning instruction outputted through Y5 (MSB) (IVC3)	R/W
SD351	Current position value in the positioning instruction outputted through Y5 (LSB) (IVC3)	R/W
SD360	Current position value in the positioning instruction outputted through Y6 (MSB) (IVC3)	R/W
SD361	Current position value in the positioning instruction outputted through Y6 (LSB) (IVC3)	R/W
SD370	Current position value in the positioning instruction outputted through Y7 (MSB) (IVC3)	R/W
SD371	Current position value in the positioning instruction outputted through Y7 (LSB) (IVC3)	R/W

6.10.11 PLSR: Instruction for count pulse output with acceleration/deceleration

<table>
<tr><td colspan="3" rowspan="2">LAD:
PLSR (S1) (S2) (S3) (D)</td><td>Applicable model</td><td>IVC2L IVC1 IVC1S IVC3 IVC1L</td></tr>
<tr><td>Influenced flag bit</td><td></td></tr>
<tr><td colspan="3">IL: PLSR (S1) (S2) (S3) (D)</td><td>Step length</td><td>10</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="2">Applicable soft element</td><td>Indexing</td></tr>
</table>

S1	WORD	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	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	BOOL			Y												

- Operand description

S1: Max frequency (Hz). Settable range: 10–20000 (Hz). For IVC1L, the settable range of Y2 and Y3 are within 10 to 10000 (Hz).

If the operand specified indirectly is larger than 20000, it is treated as 20000, if it is less than 10, it is treated as 10. In that case, the system an error, indicating that the value of the instruction operand is invalid, and high-speed pulse is outputted by default.

S2: Total number of output pulses (PLS). Settable range: 110–2147483647. If the set operand is not withinthe range, the system reports an error, indicating that the value of the instruction operand is invalid, outputs no pulse, and no hardware resources corresponding to this instruction is occupied.

S3: Acceleration/deceleration time (ms)

If ***S1***×***S3***<100000, ***S3*** is regarded as 100000/***S1***, the system reports a parameter error of the PLSR instruction, and the acceleration/deceleration time is uncertain.

If ***S1***×***S3***>S2×909, S3 is regarded as S2×909/***S1***, the system reports a parameter error of the PLSR instruction, and the acceleration/deceleration time is uncertain.

Note

For IVC1, the acceleration/deceleration time cannot be less than 50ms.

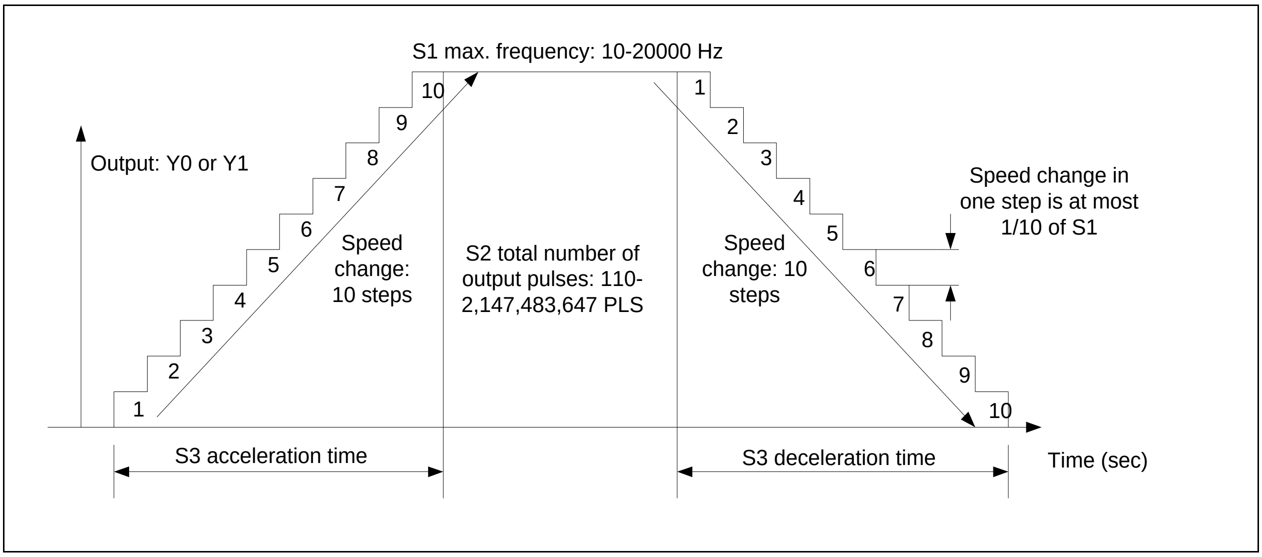
The speed change is evenly divided into 10 steps during the acceleration/deceleration is treated, and each step is ***S1***/10.

D: High-speed pulse output point. For IVC1 and IVC2L, only Y0 or Y1 can be designated; for IVC1L, only Y0, Y1, and Y2 can be designated; for IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

- Function description

The PLSR instruction is a high-speed pulse output instruction with acceleration/deceleration function for fix-dimension transmission. Targeting the specified max. frequency, the pulse output is accelerated evenly and after the number of output pulses reaches the preset value, the pulse output is decelerated evenly.

The operation process is shown in the following figure:



- Note

1. The output frequency of this instruction is 10–20000Hz. If the speed variation during acceleration/deceleration exceeds this range, the operand is adjusted automatically within the range. This instruction is not affected by the scan cycle.

2. The PLCs need to use the transistor output. During the high-speed pulse output, the output transistor is connected according to the specified load current. The waveforms obtained from the output terminals (Y0 and PORT0, Y1 and PORT1) are related to the user's output load. When the load cannot exceed the rated load current, the smaller the load, the closer the output waveform is to the set operand.

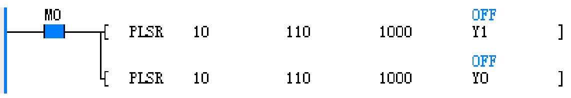
3. Whenthe high-speed instruction is executed effectively (including output completion), other operations on the same portare invalid. Only when the high-speed pulse output instruction is invalid can other instructions operate this port.

4. Usingtwo PLSR instructions can obtain separate high-speed pulse output at output points, or using the PWM (or PLSR) instruction to obtain separate high-speed pulse output at different output points.

5. When multiple PWM, PLSY or PLSR instructions operate on the same port, the first valid instruction controls the output state of the portand the later valid instructions do not affect the state of the output points.

6. Similar to other high-speed instructions (DHSCS, DHSCR,DHSZ, DHSP, DHST, and HCNT), the PLSR instruction needs to meet the requirements for high-speed input and high-speed pulse output in the system.

- Application instance



LD M0

PLSR 10 110 1000 Y1

PLSR 10 110 1000 Y0

1. When M0 is ON, pulses are output through Y0 and Y1 ports at the set frequency, and pulse output is stopped after 110 pulses are outputted. When M0 changes from OFF to ON, the next round of pulse output starts. When M0 is OFF, the port output is OFF.

2. All operandsare not changed during the execution of the instruction, they are processed according to the operand which is valid first. The new operand becomes valid only when M0 changes from ON→OFF→ON.

3. SM80 is used to control the output of Y0 while SM81 is used to control the output of Y1, and when SM80 or SM81 is 1, output pulses corresponding to the output points are disabled.

4. SM82 and SM83 indicate the output flags of Y0 and Y1 respectively. The corresponding SM82 or SM83 is OFF when the output is completed or M0 is OFF. The corresponding SM82 or SM83 is ON when the output is going on.

5. SD50－SD55 correspond to:

1) SD50: The MSB of the number of the Y0 output pulses in PLSY and PLSR instructions.

2) SD51: The LSB of the number of the Y0 output pulses in PLSY and PLSR instructions.

3) SD52: The MSB of the number of the Y1 output pulses in PLSY and PLSR instructions.

4) SD53: The LSB of the number of the Y1 output pulses in PLSY and PLSR instructions.

5) SD54: The MSB of the number of the Y0 and Y1 in PLSY and PLSR instructions.

6) SD55: The LSB of the number of the Y0 and Y1 in PLSY and PLSR instructions.

6. SD50–SD55 can be modified through "DMOV ××× SD5×"or"MOV ××× SD5×", or through the monitoring.

Relevant soft element:

Address	Name	Action and function	R/W
SM80	Y0 high-speed pulse output control	Y0 high-speed pulse output stop instruction	R/W
SM81	Y1 high-speed pulse output control	Y1 high-speed pulse output stop instruction	R/W
SM82	Y0 high-speed pulse output monitoring	Y0 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM83	Y1 high-speed pulse output monitoring	Y1 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM262	Y002 pulse output stop instruction	When this bit is set, Y002 pulse is disabled	R/W
SM264	Y004 pulse output stop instruction	When this bit is set, Y004 pulse is disabled	R/W
SM265	Y005 pulse output stop instruction	When this bit is set, Y005 pulse is disabled	R/W
SM266	Y006 pulse output stop instruction	When this bit is set, Y006 pulse is disabled	R/W
SM267	Y007 pulse output stop instruction	When this bit is set, Y007 pulse is disabled	R/W
SM272	Y002 pulse output monitoring (busy/ready)	Y002 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM274	Y004 pulse output monitoring (busy/ready)	Y004 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM275	Y005 pulse output monitoring (busy/ready)	Y005 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM276	Y006 pulse output monitoring (busy/ready)	Y006 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM277	Y007 pulse output monitoring (busy/ready)	Y007 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R

Address	Action and function	R/W
SD50	Accumulated total number of high-speed pulses outputted through Y0 (MSB) (IVC2L IVC1L, and IVC3)	R/W
SD51	Accumulated total number of high-speed pulses outputted through Y0 (LSB) (IVC2L IVC1L, and IVC3)	R/W
SD52	Accumulated total number of high-speed pulses outputted through Y1 (MSB) (IVC2L IVC1L, and IVC3)	R/W
SD53	Accumulated total number of high-speed pulses outputted through Y1 (LSB) (IVC2L IVC1L, and IVC3)	R/W
SD54	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (MSB) (IVC2L)	R/W
SD55	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (LSB) (IVC2L)	R/W
SD160	Accumulated total number of high-speed pulses outputted through Y2 (MSB) (IVC1L and IVC3)	R/W
SD161	Accumulated total number of high-speed pulses outputted through Y2 (LSB) (IVC1L and IVC3)	R/W
SD162	Accumulated total number of high-speed pulses outputted through Y3 (MSB) (IVC3)	R/W
SD163	Accumulated total number of high-speed pulses outputted through Y3 (LSB) (IVC3)	R/W
SD164	Accumulated total number of high-speed pulses outputted through Y4 (MSB) (IVC3)	R/W
SD165	Accumulated total number of high-speed pulses outputted through Y4 (LSB) (IVC3)	R/W
SD166	Accumulated total number of high-speed pulses outputted through Y5 (MSB) (IVC3)	R/W
SD167	Accumulated total number of high-speed pulses outputted through Y5 (LSB) (IVC3)	R/W
SD168	Accumulated total number of high-speed pulses outputted through Y6 (MSB) (IVC3,)	R/W
SD169	Accumulated total number of high-speed pulses outputted through Y6 (LSB) (IVC3)	R/W
SD170	Accumulated total number of high-speed pulses outputted through Y7 (MSB) (IVC3)	R/W
SD171	Accumulated total number of high-speed pulses outputted through Y7 (LSB) (IVC3)	R/W
SD80	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1)	R/W

SD81	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC1)	R/W
SD82	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC1)	R/W
SD83	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC1)	R/W
SD200	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC2L IVC1L, and IVC3)	R/W
SD201	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC2L IVC1L, and IVC3)	R/W
SD210	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC2L IVC1L, and IVC3)	R/W
SD211	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC2L IVC1L, and IVC3)	R/W
SD320	Current position value in the positioning instruction outputted through Y2 (MSB) (IVC1L and IVC3)	R/W
SD321	Current position value in the positioning instruction outputted through Y2 (LSB) (IVC1L and IVC3)	R/W
SD330	Current position value in the positioning instruction outputted through Y3 (MSB) (IVC3)	R/W
SD331	Current position value in the positioning instruction outputted through Y3 (LSB) (IVC3)	R/W
SD340	Current position value in the positioning instruction outputted through Y4 (MSB) (IVC3)	R/W
SD341	Current position value in the positioning instruction outputted through Y4 (LSB) (IVC3)	R/W
SD350	Current position value in the positioning instruction outputted through Y5 (MSB) (IVC3)	R/W
SD351	Current position value in the positioning instruction outputted through Y5 (LSB) (IVC3)	R/W
SD360	Current position value in the positioning instruction outputted through Y6 (MSB) (IVC3)	R/W
SD361	Current position value in the positioning instruction outputted through Y6 (LSB) (IVC3)	R/W
SD370	Current position value in the positioning instruction outputted through Y7 (MSB) (IVC3)	R/W
SD371	Current position value in the positioning instruction outputted through Y7 (LSB) (IVC3)	R/W

6.10.12 PLS: Envelopepulse output instruction

LAD: PLS (S1) (S2) (D1)	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	
IL: PLS *(S1) (S2) (D1)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD	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: Start address of D element designated by the parameter

S2: Number of the output segments. Range: 0 - 255.

D1: High-speed pulse output point. For IVCI and IVC2L, only Y0 or Y1 can be designated; for IVC1L, only Y0, Y1, and Y2 can be designated; for IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

- Function description

1. Using the Auto Station instruction wizard to generate the PLS instruction, which can be called like subprograms. When the energy flow is turned on, the system outputsthe corresponding pulses according to the configuration. You can control ON or OFF of the pulse generator and set the frequency and pulse number.

2. There is no output when the segment number is 0.

3. High-speed pulse output can be turned off by setting SM80 and SM81 to ON, and other flag bits are shared with high-speed outputs.

4. The contents of the subprogram PLS_SET generated by the Auto Station software are as follows (set n as its D element number and M as the total number of the segments):

LD SM0

DMOV Frequency of the first segment procedure Dn

DMOV Pulse number of the first segment procedure Dn＋2

DMOV Frequency of the second segment procedure Dn＋4

DMOV Pulse number of the second segment procedure Dn＋6

DMOV Frequency of the third segment procedure Dn＋8

DMOV Pulse number of the third segment procedure Dn＋10

………

DMOV Frequency of the M segment procedure Dn ＋4M－4

DMOV Pulse number of the M segment procedure Dn＋4M－2

DMOV Max. speed Dn＋4M

MOV Min. speed Dn＋4M＋2

MOV Acc eleration time Dn＋4M＋3

MOV Deceleration time Dn＋4M＋4

- Note

1. It is recommended to use the PLS instruction generated by the PTO instruction wizard. If you write the PLS instruction manually, note that the number of pulses in each procedure cannot be too small. At the set acceleration speed, the number of pulses in each procedure must be larger than the min. number of pulses required for the frequency conversion.

2. P indicates the number of pulses in a certain procedure, F_N indicates the frequency of the N segment, $F_{\max}$ and $F_{\min}$ indicate the max. speed and min. speed, and T_{up} and T_{down} indicate the acceleration and deceleration time in ms.

1) When the speed of the procedure N is larger than that of procedure N-1, the number of the pulses in the procedure N need to 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 procedure N is less than that of procedure N-1, the number of the pulses in the procedure N need to 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. In particular:

1) When N=1, the frequency of the procedure N－1 takes $F_{\min}$, and then $F_{\min}$ is put into the above formula.

2) When the number of procedures is 1, namely there is only one segment, the number of the pulses needs to meet the following condition:

$$P \geq \frac{(F_1 + F_{\min}) \times (F_1 - F_{\min}) \times (T_{up} + T_{down})}{2000 \times (F_{\max} - F_{\min})}$$

3) The number of the pulses in the last procedure needs tomeet 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 designated in each procedure needs to be within the setting range of the max. or min.speed.

5) The total number of pulses in all procedure isa maximum of 999,999.

4. The PLCs need to use the transistor output. During the high-speed pulse output, the output transistor is connected according to the specified load current. The waveforms obtained from the output terminalsare related to the user's output load. When the load cannot exceed the rated load current, the smaller the load, the closer the output waveform is to the set operand.

5. When the high-speed instruction is executed effectively (including output completion), other operations on the same portare invalid. Only when the high-speed pulse output instruction is invalid can other instructions operate this port.

6. PLSY, PLSR, PLS, and positioning instructions can output the high-speed pulses through ports. It is not allowed to use these instructions for high-speed pulse output on the same port at the same time.

Address	Name	Action and function	R/W
SM80	Y0 high-speed pulse output control	Y0 high-speed pulse output stop instruction	R/W
SM81	Y1 high-speed pulse output control	Y1 high-speed pulse output stop instruction	R/W
SM82	Y0 high-speed pulse output monitoring	Y0 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM83	Y1 high-speed pulse output monitoring	Y1 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM262	Y002 pulse output stop instruction	When this bit is set, Y002 pulse is disabled	R/W
SM264	Y004 pulse output stop instruction	When this bit is set, Y004 pulse is disabled	R/W
SM265	Y005 pulse output stop instruction	When this bit is set, Y005 pulse is disabled	R/W
SM266	Y006 pulse output stop instruction	When this bit is set, Y006 pulse is disabled	R/W
SM267	Y007 pulse output stop	When this bit is set, Y007 pulse is disabled	R/W

	instruction		
SM272	Y002 pulse output monitoring (busy/ready)	Y002 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM274	Y004 pulse output monitoring (busy/ready)	Y004 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM275	Y005 pulse output monitoring (busy/ready)	Y005 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM276	Y006 pulse output monitoring (busy/ready)	Y006 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM277	Y007 pulse output monitoring (busy/ready)	Y007 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM88	Envelop line cyclic execution	When it is ON, envelope line executes cyclically	R/W

Address	Action and function	R/W
SD56	Number of segment that is being outputted during the envelope output of Y000	R
SD57	Number of segment that is being outputted during the envelope output of Y001	R
SD88	Rising time (ms) of envelopeoutput	R/W
SD89	Falling time (ms) of envelope output	R/W
SD252	Number of segment that is being outputted during the envelope output of Y002	R
SD253	Number of segment that is being outputted during the envelope output of Y003	R
SD254	Number of segment that is being outputted during the envelope output of Y004	R
SD255	Number of segment that is being outputted during the envelope output of Y005	R
SD256	Number of segment that is being outputted during the envelope output of Y006	R
SD257	Number of segment that is being outputted during the envelope output of Y007	R
SD50	Accumulated total number of high-speed pulses outputted through Y0 (MSB)	R/W
SD51	Accumulated total number of high-speed pulses outputted through Y0 (LSB)	R/W
SD52	Accumulated total number of high-speed pulses outputted through Y1 (MSB)	R/W
SD53	Accumulated total number of high-speed pulses outputted through Y1 (LSB)	R/W
SD54	Accumulated total number of Y1 and Y0 pulses outputted by high-speed pulse output instruction (MSB)	R/W
SD55	Accumulated total number of Y1 and Y0 pulses outputted by high-speed pulse output instruction (LSB)	R/W
SD160	Accumulated total number of high-speed pulses outputted through Y2 (MSB)	R/W
SD161	Accumulated total number of high-speed pulses outputted through Y2 (LSB)	R/W
SD162	Accumulated total number of high-speed pulses outputted through Y2 (MSB)	R/W
SD163	Accumulated total number of high-speed pulses outputted through Y2 (LSB)	R/W
SD164	Accumulated total number of high-speed pulses outputted through Y4 (MSB)	R/W
SD165	Accumulated total number of high-speed pulses outputted through Y4 (LSB)	R/W
SD166	Accumulated total number of high-speed pulses outputted through Y5 (MSB)	R/W
SD167	Accumulated total number of high-speed pulses outputted through Y5 (LSB)	R/W
SD168	Accumulated total number of high-speed pulses outputted through Y6 (MSB)	R/W
SD169	Accumulated total number of high-speed pulses outputted through Y6 (LSB)	R/W
SD170	Accumulated total number of high-speed pulses outputted through Y7 (MSB)	R/W
SD171	Accumulated total number of high-speed pulses outputted through Y7 (LSB)	R/W
SD80	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1)	R/W
SD81	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC1)	R/W
SD82	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC1)	R/W
SD83	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC1)	R/W
SD200	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1L, IVC2L, and IVC3)	R/W
SD201	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC1L, IVC2L, and IVC3)	R/W
SD210	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC1L, IVC2L, and IVC3)	R/W
SD211	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC1L, IVC2L, and IVC3)	R/W
SD320	Current position value in the positioning instruction outputted through Y2 (MSB) (IVC1L and IVC3)	R/W

Address	Action and function	R/W
SD321	Current position value in the positioning instruction outputted through Y2 (LSB) (IVC1L and IVC3)	R/W
SD330	Current position value in the positioning instruction outputted through Y3 (MSB) (IVC3)	R/W
SD331	Current position value in the positioning instruction outputted through Y3 (LSB) (IVC3)	R/W
SD340	Current position value in the positioning instruction outputted through Y4 (MSB) (IVC3)	R/W
SD341	Current position value in the positioning instruction outputted through Y4 (LSB) (IVC3)	R/W
SD350	Current position value in the positioning instruction outputted through Y5 (MSB) (IVC3)	R/W
SD351	Current position value in the positioning instruction outputted through Y5 (LSB) (IVC3)	R/W
SD360	Current position value in the positioning instruction outputted through Y6 (MSB) (IVC3)	R/W
SD361	Current position value in the positioning instruction outputted through Y6 (LSB) (IVC3)	R/W
SD370	Current position value in the positioning instruction outputted through Y7 (MSB) (IVC3)	R/W
SD371	Current position value in the positioning instruction outputted through Y7 (LSB) (IVC3)	R/W

6.10.13 PLSB: Instruction for count pulse output with base frequency and acceleration/deceleration

LAD: [*PLSB (S1) (S2) (S3) (S4) (D)*]	**Applicable model**	**IVC1 IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: PLSB *(S1) (S2) (S3) (S4) (D)*	**Step length**	**12**

Operand	Type	Applicable soft element														Indexing
S1	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S2	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
S3	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
S4	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D1	BOOL			Y												

- Operand description

  S1: Base frequency (HZ)

  Settable range: 0﹣20000 Hz. If the set operand is not withinthe range, the system reports an error, indicating that the value of the instruction operand is invalid, outputs no pulse, and no hardware resources corresponding to this instruction is occupied.

  S2: Max. frequency (HZ)

  Settable range: 10﹣20000 (Hz). If the set operand is not within the range, the system reports an error, indicating the value of the instruction operand is invalid, outputs no pulse, and no hardware resources corresponding to this instruction is occupied.

  S3: Total number of output pulses (PLS)

  Settable range: 5-999999. If the set operand is not within the range, the system reports an error, indicating that the value of the instruction operand is invalid, outputs no pulse, and no hardware resources corresponding to this instruction is occupied.

  S4: Acceleration/deceleration time(ms)

  Settable range: 0-10000. If the set operand is not within the range, the system reports an error, indicating the value of the instruction operand is invalid, outputs no pulse, and no hardware resources corresponding to this instruction is occupied.

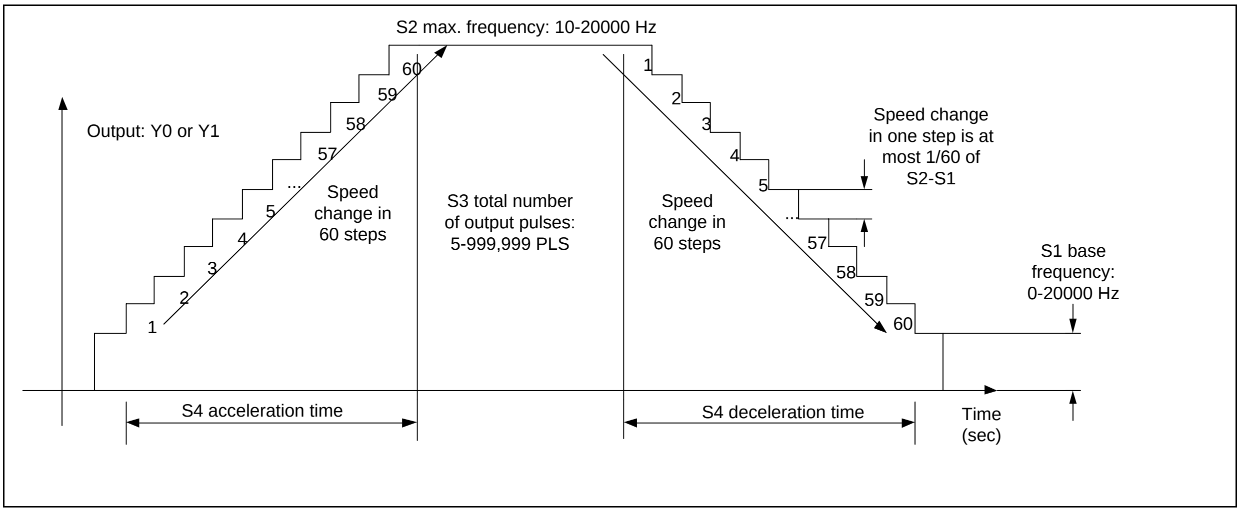
  D: Pulse output port

  For IVC1 and IVC2L, only Y0 or Y1 can be designated; for IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

- Function description

  The PLSB instruction is a high-speed pulse output instruction with base frequency and acceleration/deceleration function for fix-dimension transmission. Targeting the specified max. frequency, the pulse output is accelerated evenly at the base frequency, and after the number of output pulses reaches the preset value, the pulse output is decelerated evenly to the base frequency.

  The operation process is shown in the following figure:



- Note

  1. The output frequency of this instruction is 10–20000Hz. If the speed variation during acceleration/deceleration exceeds this range, the operand is adjusted automatically within the range. This instruction is not affected by the scan cycle.

  2. When the operand set by a user is not applicable (eg. the number of pulses and the corresponding acceleration/deceleration time are too small compared with the frequency), acceleration/deceleration is performed based on the operand set by the user and the number of output pulses is kept correct, but the output frequency may be lower than the set value of the operand, and the acceleration/deceleration time may be also lower than the set value.

  3. This instruction adopts 60-level acceleration/deceleration. However, if the operands set by the user are improper or the number of pulses is too small, the level of acceleration/decelerationis decreased properly.

  4. The PLCs need to use the transistor output. During the high-speed pulse output, the output transistor is connected according to the specified load current. The waveforms obtained from the output terminals are related to the user's output load. When the load cannot exceed the rated load current, the smaller the load, the closer the output waveform is to the set operand.

  5. When the high-speed instruction is executed effectively (including output completion), other operations on the same portare invalid. Only when the

high-speed pulse output instruction is invalid can other instructions operate this port.

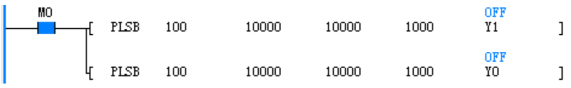
6. Using two PLSB instructions can obtain separate pulse output at Y000 and Y001, or using the PWM (or PLSR) instruction to obtain separate pulse output at different output ports.

7. When multiple PWM, PLSY or PLSR instructions operate on the same port, the first valid instruction controls the output state of the port and the later valid instructions do not affect the state of the output points.

8. Similar to other high-speed instructions (DHSCS, DHSCR,DHSZ, DHSP, DHST, and HCNT), the PLSB instruction needs to meet the requirements for high-speed I/O in the system.

9. PLSY, PLS, and positioning instructions can output the high-speed pulses through Y0 and Y1. It is not allowed to use these instructions for high-speed pulse output on the same port at the same time.

- Application instance



LD M0

PLSR 100 10000 10000 1000 Y1

PLSR 100 10000 10000 1000 Y0

1. When M0 is ON, pulses are output through Y0 and Y1 at the set frequency, and pulse output is stopped after 10000 pulses are outputted. When M0 changes from OFF to ON, the next round of pulse output starts. When M0 is OFF, the port output is OFF.

2. All operandsare not changed during the execution of the instruction, they are processed according to the operand which is valid first. The new operand becomes valid only when M0 changes from ON→OFF→ON.

3. SM80 is used to control the output of Y0 while SM81 is used to control the output of Y1, and when SM80 or SM81 is 1, output pulses corresponding to the output points are disabled.

4. SM82 and SM83 indicate the output flags of Y0 and Y1 respectively. The corresponding SM82 or SM83 is OFF when the output is completed or M0 is OFF. The corresponding SM82 or SM83 is ON when the output is going on.

5. SD50 – SD55 correspond to:

1) SD50: The MSB of the number of the Y0 output pulses in PLSB instruction.

2) SD51: The LSB of the number of the Y0 output pulses in PLSB instruction.

3) SD52: The MSB of the number of the Y1 output pulses in PLSB instruction.

4) SD53: The LSB of the number of the Y1 output pulses in PLSB instruction.

5) SD54: The MSB of the number of the Y0 and Y1 in PLSB instruction.

6) SD55: The LSB of the number of the Y0 and Y1 in PLSB instruction.

6. SD50–SD55 can be modified through "DMOV ××× SD5×"or"MOV ××× SD5×", or through the monitoring.

Relevant soft elements:

Address	Name	Action and function	R/W
SM80	Y0 high-speed pulse output control	Y0 high-speed pulse output stop instruction	R/W
SM81	Y1 high-speed pulse output control	Y1 high-speed pulse output stop instruction	R/W
SM82	Y0 high-speed pulse output monitoring	Y0 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM83	Y1 high-speed pulse output monitoring	Y1 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM262	Y002 pulse output stop instruction	When this bit is set, Y002 pulse is disabled	R/W
SM264	Y004 pulse output stop instruction	When this bit is set, Y004 pulse is disabled	R/W
SM265	Y005 pulse output stop instruction	When this bit is set, Y005 pulse is disabled	R/W
SM266	Y006 pulse output stop instruction	When this bit is set, Y006 pulse is disabled	R/W
SM267	Y007 pulse output stop instruction	When this bit is set, Y007 pulse is disabled	R/W
SM272	Y002 pulse output monitoring (busy/ready)	Y002 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM274	Y004 pulse output monitoring (busy/ready)	Y004 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM275	Y005 pulse output monitoring (busy/ready)	Y005 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM276	Y006 pulse output monitoring (busy/ready)	Y006 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM277	Y007 pulse output monitoring (busy/ready)	Y007 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R

Address	Action and function	R/W

Address	Action and function	R/W
SD50	Accumulated total number of high-speed pulses outputted through Y0 (MSB)	R/W
SD51	Accumulated total number of high-speed pulses outputted through Y0 (LSB)	R/W
SD52	Accumulated total number of high-speed pulses outputted through Y1 (MSB)	R/W
SD53	Accumulated total number of high-speed pulses outputted through Y1 (LSB)	R/W
SD54	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (MSB)	R/W
SD55	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (LSB)	R/W
SD160	Accumulated total number of high-speed pulses outputted through Y2 (MSB)	R/W
SD161	Accumulated total number of high-speed pulses outputted through Y2 (LSB)	R/W
SD164	Accumulated total number of high-speed pulses outputted through Y4 (MSB)	R/W
SD165	Accumulated total number of high-speed pulses outputted through Y4 (LSB)	R/W
SD166	Accumulated total number of high-speed pulses outputted through Y5 (MSB)	R/W
SD167	Accumulated total number of high-speed pulses outputted through Y5 (LSB)	R/W
SD168	Accumulated total number of high-speed pulses outputted through Y6 (MSB)	R/W
SD169	Accumulated total number of high-speed pulses outputted through Y6 (LSB)	R/W
SD170	Accumulated total number of high-speed pulses outputted through Y7 (MSB)	R/W
SD171	Accumulated total number of high-speed pulses outputted through Y7 (LSB)	R/W
SD80	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1)	R/W
SD81	Current position value in the positioning instruction outputted through Y0 (LSB)(IVC1)	R/W
SD82	Current position value in the positioning instruction outputted through Y1 (MSB)	R/W
SD83	Current position value in the positioning instruction outputted through Y1 (LSB)	R/W
SD200	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC2L and IVC3)	R/W
SD201	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC2L and IVC3)	R/W
SD210	Current position value in the positioning instruction outputted through Y1 (MSB)	R/W
SD211	Current position value in the positioning instruction outputted through Y1 (LSB)	R/W
SD320	Current position value in the positioning instruction outputted through Y2 (MSB)	R/W
SD321	Current position value in the positioning instruction outputted through Y2 (LSB)	R/W
SD340	Current position value in the positioning instruction outputted through Y4 (MSB)	R/W
SD341	Current position value in the positioning instruction outputted through Y4 (LSB)	R/W
SD350	Current position value in the positioning instruction outputted through Y5 (MSB)	R/W
SD351	Current position value in the positioning instruction outputted through Y5 (LSB)	R/W
SD360	Current position value in the positioning instruction outputted through Y6 (MSB)	R/W
SD361	Current position value in the positioning instruction outputted through Y6 (LSB)	R/W
SD370	Current position value in the positioning instruction outputted through Y7 (MSB)	R/W
SD371	Current position value in the positioning instruction outputted through Y7 (LSB)	R/W

6.10.14 PWM: Pulse output instruction

<table>
<tr><td colspan="9" rowspan="2">LAD:
[PWM (S1) (S2) (D)]</td><td colspan="3">Applicable model</td><td colspan="5">IVC2L IVC1 IVC1S IVC3 IVC1L</td></tr>
<tr><td colspan="3">Influenced flag bit</td><td colspan="5"></td></tr>
<tr><td colspan="9">IL: PWM (S1) (S2) (D)</td><td colspan="3">Step length</td><td colspan="5">7</td></tr>
<tr><td>Opera</td><td>Type</td><td colspan="14">Applicable soft element</td><td>Indexing</td></tr>
<tr><td>S1</td><td>INT</td><td>Constant</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnLM</td><td>KnSM</td><td>D</td><td>SD</td><td>C</td><td>T</td><td>V</td><td>Z</td><td>R</td><td>√</td></tr>
<tr><td>S2</td><td>INT</td><td>Constant</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>KnLM</td><td>KnSM</td><td>D</td><td>SD</td><td>C</td><td>T</td><td>V</td><td>Z</td><td>R</td><td>√</td></tr>
<tr><td>D</td><td>BOOL</td><td></td><td></td><td>Y</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

- Operand description

S1: Assigned pulse width (ms/us)

Settable range: 0－32767 (ms), when ***S1*** is larger than 32767, the system reports that the instruction operand is illegal and no system hardware resource is occupied.

If the content of ***S1*** is changed while the instruction is running, the output pulse is also changed. When SM84 is 0, ***S1*** is in the unit of ***ms***; when SM84 is 1, ***S1*** is in the unit of us.

S2: Assigned pulse cycle (ms)

Settable range: 1 – 32767, when the set operand is not within the setting range, the system reports that the instruction operand is illegal, the pulse does not output, and no system hardware resource is occupied.

If the content of ***S2*** is changed while the instruction is running, the output pulse is also changed. When SM84 is 0, ***S1*** is in the unit of ***ms***; when SM84 is 1, ***S1*** is in the unit of us.

S2 needs to be larger than or equal to ***S1***, otherwise the system reports an operand error, and no system resource is occupied.

D: High-speed pulse output point. For IVC1 and IVC2L, only Y0 or Y1 can be designated; for IVC1L, only Y0, Y1, and Y2 can be designated; for IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

- Function description

The PWM pulse whose output width is ***S1*** and the pulse cycle is ***S2*** is outputted at the port designated by ***D***.

Note

1. When ***S1*** is 0, the high-speed output port output is always OFF. When ***S1***=***S2***, the high-speed output port output is always ON.

2. The waveforms obtained from the output terminals are related to the user’s output load. When the max. output current is fulfilled, the smaller the load, the closer the output waveform is to the set operand. In order to output the high-speed pulses, the load current on the output transistors of the PLCs needs to be large enough within the range of the rated load current.

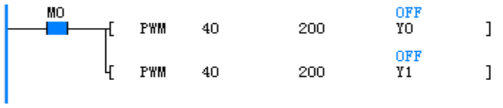
3. When the high-speed instruction is executed effectively (including output completion), other operations on the same portare invalid. Only when the high-speed pulse output instruction is invalid can other instructions operate this port.

4. Using two PWM instructions can obtain separate pulse output atoutput ports, or using the PLSY (or PLSR) instruction to obtain separate high-speed pulse output at different output ports.

5. When multiple PWM, PLSY or PLSR instructions operate on the same port, the first valid instruction controls the output state of the port and the later valid instructions do not affect the state of the output points.

6. Similar to other high-speed instructions (DHSCS, DHSCR,DHSZ, DHSP, DHST, and HCNT), the PWM instruction needs to meet the requirements for high-speed I/O in the system.

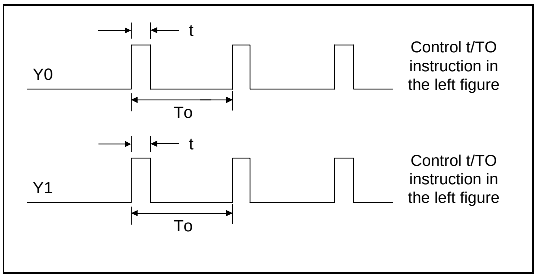
- Application instance



LD M0

PWM 40 200 Y0

PWM 40 200 Y1



Where "t" is pulse width and T0 is pulse cycle.

1. When M0 is ON, Y0 and Y1 output the PWM pulseswith the width of 40ms and the cycle of 200 ms. When M0 is OFF, theoutput is OFF. The output state is not affected by the scan cycle.

2. SM80 is used to disable the output of Y0, while SM81 is used to disable the output of Y1. When SM80 and SM81 are ON, the output is stopped.

3. SM82 and SM83 indicate the output flag of Y0 and Y1 respectively. When M0 is OFF, SM82 and SM83 are OFF.

Relevant soft elements:

Address	Name	Action and function	R/W
SM80	Y0 high-speed pulse output control	Y0 high-speed pulse output stop instruction	R/W
SM81	Y1 high-speed pulse output control	Y1 high-speed pulse output stop instruction	R/W
SM82	Y0 high-speed pulse output monitoring	Y0 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM83	Y1 high-speed pulse output monitoring	Y1 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM84	PWM time base unit	When it is ON, time base unit is us; when it is OFF, time base unit is ms	R/W
SM264	Y004 pulse output stop instruction	When this bit is set, Y004 pulse is disabled	R/W
SM265	Y005 pulse output stop instruction	When this bit is set, Y005 pulse is disabled	R/W
SM266	Y006 pulse output stop instruction	When this bit is set, Y006 pulse is disabled	R/W
SM267	Y007 pulse output stop instruction	When this bit is set, Y007 pulse is disabled	R/W
SM274	Y004 pulse output monitoring (busy/ready)	Y004 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM275	Y005 pulse output monitoring (busy/ready)	Y005 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM276	Y006 pulse output monitoring (busy/ready)	Y006 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R
SM277	Y007 pulse output monitoring (busy/ready)	Y007 high-speed pulse output monitoring (when busy, it is ON, when ready, it is OFF)	R

6.11 Control calculation instructions

6.11.1 PID: Function instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
├──┤ ├──[PID (S1) (S2) (S3) (D)	Influenced flag bit	
IL: PID (S1) (S2) (S3) (D)	Step length	9

Operand	Type	Applicable soft element													Indexing
S1	INT								D					R	√
S2	INT								D					R	√
S3	INT								D					R	√
D	INT								D					R	√

- Operand description

D: Operation result output (MV) when executing the program

S1: Set target value (SV)

S2: Current measurement value (PV)

S3: Sampling time (Ts). Range: 1–32767 (ms). It need to be set longer than the calculation time.

S3＋1: Word for setting the action, alarm and upper/lower limit function.

Bit	Set value and definition	
	0	1
0	Positive feedback	Adverse feedback
1	Input variation alarm invalid	Input variation alarm valid
2	Output variation alarm invalid	Output variation alarm valid
3 ~ 4	Reserved	
5	Output value upper/lower limit value setup is invalid	Output value upper/lower limit value setup is valid
6 ~ 15	Reserved	

S3＋2: Input filtering constant (α). Range:0–99[%]. When it is 0,there is no input filtering.

S3＋3: Proportional gain (Kp). Range:1–32767[%].

S3＋4: Integral time (TI). Range:0–32767 (×100ms). When it is 0, it is processed as ∞ (no integral).

S3＋5: Differential gain (KD). Range:0–100[%]. When it is 0, there is no differential gain.

S3＋6: Differential time (TD).Range:0–32767 (×10ms). When it is 0, it is processed as no differential.

S3＋7–***S3＋14***: Internal data register for PID operation.

S3＋15: Alarm set value of the PID input variation (positive change). Range:0–32767 (when bit 1 of ***S3＋1*** is 1).

S3＋16: Alarm set value of the PID input variation (negative change). Range:0–32767 (when bit 1 of ***S3＋1*** is 1).

S3＋17: Alarm set value of the PID output variation (positive change). Range:0–32767 (when bit 2 and bit 5 of ***S3＋1*** are 1 and 0 respectively).

Set value of the output upper limit. Range:-32768–+32767 (when bit 2 and bit 5 of ***S3＋1*** are 0 and 1 respectively).

S3＋18: Alarm set value of the PID output variation (negative change). Range: 0–32767 (when bit 2 and bit 5 of ***S3＋1*** are 1 and 0 respectively).

Set value of the output lower limit. Range: -32768–+32767 (when bit2 and bit5 of ***S3＋1*** are 0 and 1 respectively).

S3＋19: PID alarm output.

- Bit 0: Input variation (positive change) overflows.
- Bit 1: Input variation (negative change) overflows.
- Bit 2: Output variation (positive change) overflows.
- Bit 3: Output variation (negative change) overflows.

In which, S3–S3＋6are mandatory user set operands while S3＋15–S3＋19 areoptional user set operands. You can set the operand s through the PID instruction wizard of Auto station.

- Function description

1. Performing the PID operation when the energy flow is valid and the sampling time is reached.

2. Multiple PID instruction can be executed simultaneously by multiple times (on limit on the loop number). But you need to note that the S1, S2, S3 or D soft element number used by the operation cannot be overlayed.

3. The PID instruction is applicable to the timed interrupt subprograms, general subprograms, and main programs. Under this condition, you need to confirm

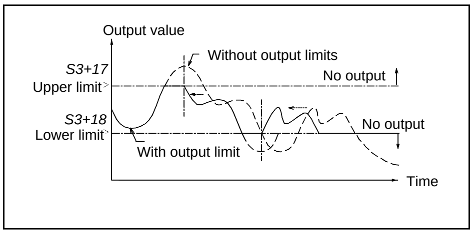
operand set unit and clear the internal processing data of ***S3＋7*** before executing the PID instruction.。

4. The input filtering constant can smooth the changes of measured value.

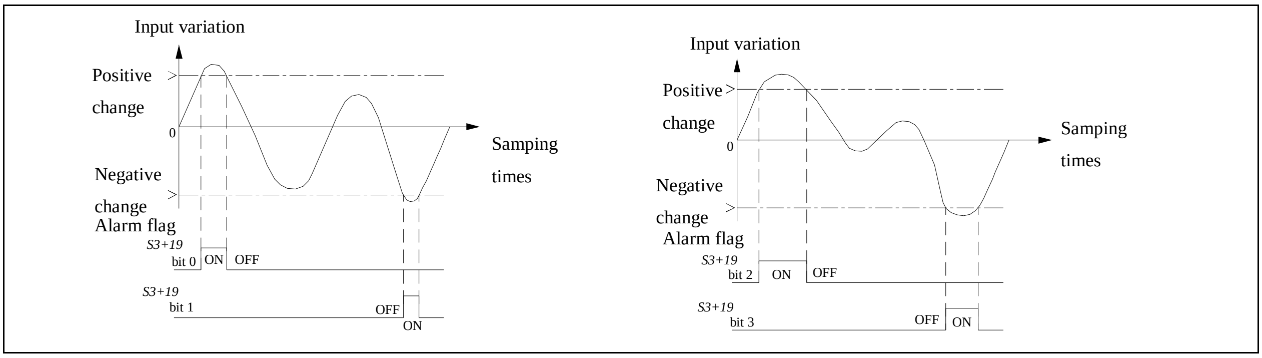
5. The differential gain can ease the dramatic change of output value.

6. Action direction: Setting the forward (positive feedback) and reverse actions (adverse feedback) of the system via bit 0 of ***S3＋1***.

7. Output upper/lower limit setting: When you set the output upper/lower limit to be valid (bit 5 and bit 2 of ***S3＋1*** are ON and OFF respectively) , the integral item of PID can be prevented from becoming too large. The output value is shown as below:



8. Alarm setting: When you set the output upper/lower limit to be valid (in ***S3＋1***, bit 1 is ON, bit 2 is ON, and bit 5 is OFF) , the PID instruction compares the I/O variation with the preset value in ***S3＋15*** –***S3＋18***. If the current value is larger than the preset value, PID reports an alarm, and the corresponding function bits in ***S3＋19*** are set immediately after executing the PID instruction. In this way, you can monitor the IO variation. The current output values are shown as below:



9. Basic expression of the PID instruction:

Action direction	PID expression
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 description is shown below:

Symbol	Description	Symbol	Description
EV_n	Current sampling deviation	D_n	Current differential item
EV_{n-1}	Deviation before one cycle	D_{n-1}	Differential item before one cycle
SV	Target value	KP	Proportional gain
PV_{nf}	Current sampling value (after filtering)	T_S	Sampling cycle
PV_{nf-1}	Sampling value before one cycle (after filtering)	T_I	Integral time
PV_{nf-2}	Sampling value before two cycles (after filtering)	T_D	Differential time
ΔMV	Output variation	α_D	Differential gain
MV	Current operation quantity		

- Application instance

// PID initialization, if the control operands are the same, you can initialize the operands only once.

```
LD      SM1             //Initialization, executed once only
MOV     1000    D500    //Setting the target value
```

MOV 500 D510 //Sampling time (Ts). Range: 1－32767 (ms). It needs to be larger than the //calculation time

MOV 7 D511 //Action direction

MOV 70 D512 //Input filtering constant (α). Range: 0－99[％]. When it is 0, there is no input filtering

MOV 100 D513 //Proportional gain (Kp). Range: 1－32767[％]

MOV 25 D514 //Integral time (TI). Range: 0－32767 (×100ms). When it is 0 , it is processed as //∞ (no integral)

MOV 0 D515 //Differential gain (KD). Range: 0－100[％]. When it is 0, there is no differential gain

MOV 63 D516 //Differential time (TD). Range: 0－32767 (×10 ms). When it is 0, it means no differential //processing

FMOV 0 D517 8 //Clearing the area for storing the transit data of PID operation

MOV 2000 D525 //Alarm set value of the input variation (positive change). Range: 0－32767

MOV 2000 D526 //Alarm set value of the input variation (negative change). Range:0－32767

MOV 2000 D527 //Alarm set value of the output variation (positive change). Range: 0－32767

MOV 2000 D528 //Alarm set value of the output variation (negative change). Range:0－32767

//PID instruction execution operation

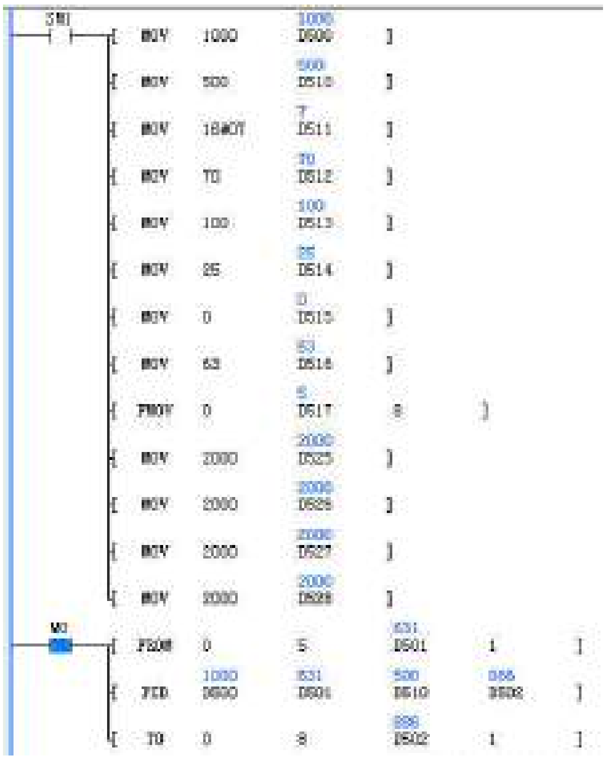
LD M0 //User-controlled PID operation program

FROM 0 5 D501 1 //Read and record the current measured value (user can inputthe measured value according to the actual situation)

PID D500 D501 D510 D502 //PID instruction: PID S1 S2 S3 D

TO 0 8 D502 1 //PID operation result is fed back to the controlled system (users can //handle the PID operation result according to the actual situation)

The LAD of the above instruction is shown below:



The PLCs initialize the PID operands only in the first scan cycle. When X2 is ON, the current measured value is read from the external A/D modules (the actual situation could be different), assigned to the measurement value unit, and the PID operation is executed. The operation result is converted into analog signals through the external D/A modules (the actual situation could be different), and fed to the controlled system.

- Note

1. For ***D***, you need to designate it into the data register outside of the Saving Range. Otherwise, it needs tobe cleared (LD SM0 MOV 0 D****) in the first operation.

2. The PID instruction needs to occupy 20 consecutive data registers starting from ***S3***.

3. The max. error of sampling time TS is － (one scancycle＋1ms)－＋ (one scan cycle). When the value of TSis relatively small, the PID effect is affected. It is recommended to use the PID instruction in timed interrupt.

4. When you set the PID output upper/lower limit to be valid, if the value of upper limit is less than that of the lower limit, the system reports an operand error, and does not execute the PID operation.

5. When you set the alarm of I/O variation to be valid, the set value of ***S3＋15－S3＋18*** cannot be negative, otherwise the system reports an operand error, and does not execute the PID operation.

6. When bit 2 and bit 5 of ***S3＋1*** are set to ON, the system determines that the setting is invalid (it is equivalent of that bit 2 and bit 5 are set to OFF), and therefore does not report alarms for upper limit, lower limit, or variation exceeding.

7. When the set value of the PID control operand (***S3*** – ***S3*** ＋***6*** unit) is outside the valid range, the system reports an operand error, and does not execute the PID operation.

8. When the sampling time is less than or equal to one scan cycle, if there is data overflow or result overflow during the operation, no alarm occurs, and the PID operation continues.

9. All PID operands need to be initialized before the initial execution of the PID instruction. If the operands remain the same during the operation, and the related operand elements are not covered by other programs, you can initialize the PID operands only once. If the data in the transit data registers are changed during the PID operation, the operation result is incorrect.

6.11.2 RAMP: Ramp signal output instruction

LAD: RAMP (S1) (S2) (D1) (S3) (D2)	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: RAMP *(S1) (S2) (D1) (S3) (D2)*	**Step length**	**12**

Operand	Type	Applicable soft element														Indexing
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: Startvalue

  S2: End value

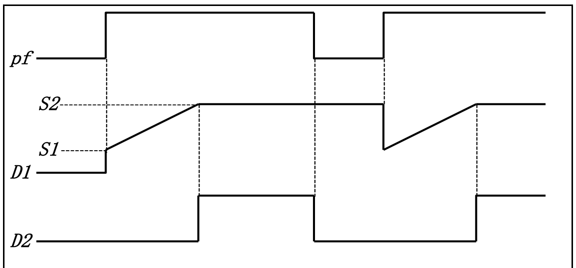
  D1: Output value

  S3: Number of steps (***S3***＞0, otherwise the system reports an operand error, and does not execute the operation)

  D2: Output state 0

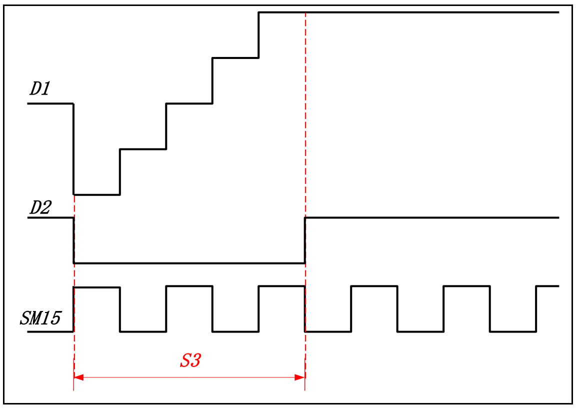
- Function description

  When a rising edge occurred to the energy flow and keeps ON, each scan cycle, based on the height and mobile scan times of the ramps, determines the incremental quantity and current output value. After reaching ***S2***, the output value (***D1***) stays in current state, and the state output position is ON. If a falling edge occurred to the energy flow, the output state (***D2***) is OFF, theoutput value (***D1***) stays in current state until the rising edge occurs to the energy flow again, the output value (***D1***) is initialized to be the value of ***S1***, and continues to generate next ramp operation, as shown below:



pf: state of the energy flow

The execution process of the ramp instructionis shown below (***S3***=5):



- Note

  1. If the result is not divisible when calculating the step length,you can adopt the "rounded off" method.

  2. The instruction generates the ramp data only once upon every rising edge.

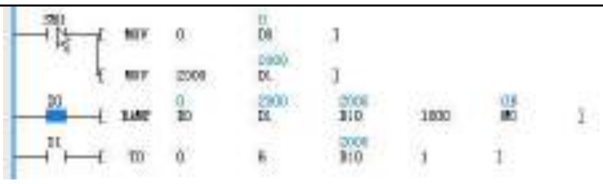
  3. When ***S1***=***S2***, ***D1***=***S2***, and ***D2***=ON.

  4. The total number of RAMP, HACKLE, and TRIANGLE instructions cannot exceed 100 in the program.

- Application instance

```
//Initializing the register upon the first scan cycle after the power-on
LD      SM1
MOV     0 D0
MOV     2000 D1
//Executing the ramp function instruction when X0=ON
LD      X0
RAMP        D0 D1 D10 1000 M0
//Sending the output value of ramp function to the external DA module when X1=ON, so as to generate ramp waveforms
LD      X1
TO      0 6 D10 1
```

The LAD of the above instructions is shown below:



1. When X0=ON, D10 (in the first cycle,D10=D0=0) increases by 2 (2000/1000) in each scan cycle. When D10=D1=2000, D10 keeps unchanged, and M0=ON.During the generation process of the ramp function, if the energy flow falls, the output state ***D2*** is OFF, the output value ***D1*** keeps its current state until the next rising edge arrives, D10=D0, and a new ramp process starts.

2. You can use an external special module to convert the data into the analog waveform.

6.11.3 HACKLE: Sawtooth wave signal output instruction

LAD: ┤├ ┤HACKLE (S1) (S2) (D1) (S3) (D2)]								Applicable model		IVC2L IVC1 IVC1S IVC3 IVC1L						
								Influenced flag bit								
IL: HACKLE *(S1)* *(S2)* *(D1)* *(S3)* *(D2)*								**Step length**		**12**						
Operand	Type	Applicable soft element														Indexing
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: Startvalue

S2: End value

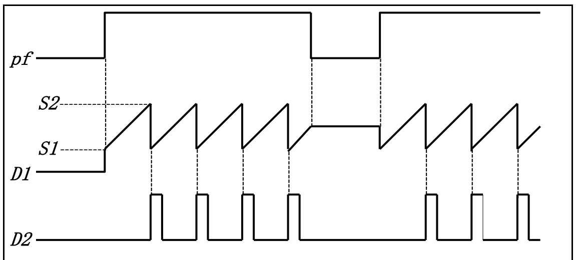
D1: Output value

S3: Number of steps (***S3***＞0, otherwise the system reports an operand error, and does not execute the operation)

D2: Output state

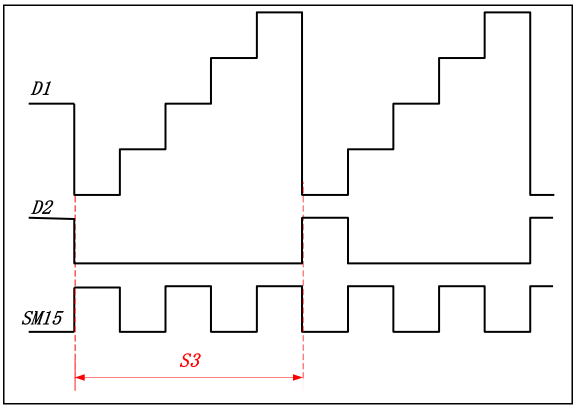
- Function description

When theenergy flow is valid, the increment and current output value (***D1***) are determined for each scan cycle according to the height and steps of the sawtooth wave (***D1***).When the output value reaches ***S2***, it is initialized to ***S1***, and the state output bit (***D2***) is set to ON. In the next scan cycle, if the energy flow keeps ON, the state output bit (***D2***) is set to OFF, and the next sawtooth wave is generated. During the generation process of sawtooth wave function, if afalling edge occurs to the energy flow, the output state (***D2***) is OFF and the output value (***D1***) remains in the current state. When a rising edge occurs to the energy flow again, the output value (***D1***) is initialized to ***S1*** and a new sawtooth wave generation process starts, as shown below.



pf: state of the energy flow

The execution process of a sawtooth wave instruction is shown as follows (***S3***=5).



- Note

1. If the result is not divisible when calculating the step length, you can adopt the "rounding-off" method.

2. The instruction generates a series of consecutive sawtooth wave data as long as the energy flow is valid.

3. When ***S1***=***S2***, ***D1***=***S2***, and ***D2***=ON (no counting pulse).

4. The total number of RAMP, HACKLE, and TRIANGLE instructions cannot exceed 100 in the program.

- Application instance

//Initializing the register upon the first scan cycle after the power-on

LD SM1

MOV 0 D0

MOV2000 D1

//Executing the sawtooth wave function instruction when X0=ON

LD X0

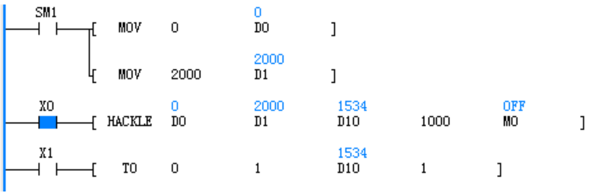
HACKLE D0 D1 D10 1000 M0

//Sending the output value of ramp function to the external DA module when X1=ON, so as to generate sawtooth waveforms

LD X1

TO 0 1 D10 1

The LAD of the above instructions is shown as follows.



1. When X0=ON, D10 (in the first cycle,D10=D0=0) increases by 2 (2000/1000) in each scan cycle. When D10=D1=2000, M0=ON. In the next scan cycle, if X0 keeps ON, D10=D0=0, and M0=OFF, the next sawtooth wave process starts. If the energy flow falls during the operation, the output state ***D2*** is OFF, the output value ***D1*** keeps its current state until the next rising edge arrives, the output value ***D1*** is initializedas ***S1***, and a new sawtooth wave process starts.

2. You can use an external special module to convert the data into the analog waveform.

6.11.4 TRIANGLE: Triangle wave signal output instruction

LAD: TRIANGLE *(S1)* *(S2)* *(D1)* *(S3)* *(D2)*		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: TRIANGLE ***(S1)*** ***(S2)*** ***(D1)*** ***(S3)*** ***(D2)***		**Step length**	**12**

Operand	Type	Applicable soft element														Indexing
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: Startvalue

S2: End value

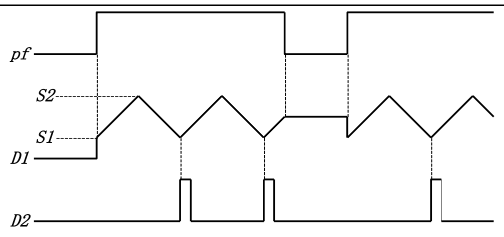
D1: Output value

S3: Number of steps (***S3***＞0, otherwise the system reports an operand error, and does not execute the operation)

D2: Output state

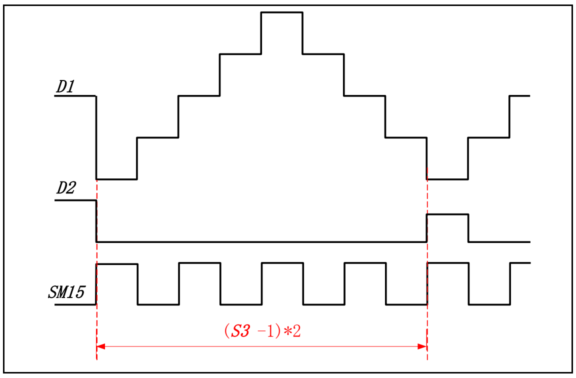
- Function description

When theenergy flow is valid, the increment and current output value (***D1***) are determined for each scan cycle according to the height and steps of the triangle wave (***D1***). When the output value reaches ***S2***, the first half slope of triangle wave has been completed, and the increment direction of the output valueis changed to continue the last half slope. When the output value (***D1***) reaches ***S1*** again, the state output bit (***D2***) is set to ON. In the next scan cycle, if the energy flow keeps ON, the state output bit (***D2***) is set to OFF, and the next triangle wave is generated. During the generation process of triangle wave function, if afalling edge occurs to the energy flow, the output state (***D2***) is OFF and the output value (***D1***)remains in the current state. When a rising edge occurs to the energy flow again, the output value (***D1***) is initialized to ***S1*** and a new triangle wave generation process starts, as shown below.



pf: state of the energy flow

The execution process of a triangle wave instruction is shown as follows (***S3***=5).



- Note

1. If the result is not divisible when calculating the step length, you can adopt the "rounding-off" method.

2. The instruction generates a series of consecutive triangle wave data as long as the energy flow is valid.

3. When ***S1***=***S2***, ***D1***=***S2***, and ***D2***=ON (no counting pulse).

4. The cycle of the triangle wave is (***S3***－1) ×2.

5. The total number of RAMP, HACKLE, and TRIANGLE instructions cannot exceed 100 in the program.

- Application instance

//Initializing the register upon the first scan cycle after the power-on

```
LD      SM1
MOV     0 D0
MOV     2000 D1
```

//Executing the triangle wave function instruction when X0=ON

```
LD      X0
TRIANGLE D0 D1 D10 1000 M0
```

//Sending the output value of ramp function to the external DA module when X1=ON, so as to generate the triangle waveform

```
LD      X1
TO      0 1 D10 1
```

The LAD of the above instructions is shown below:

1. When X0=ON, D10 (in the first cycle,D10=D0=0) increases by 2 (2000/1000) in each scan cycle.When D10=D1=2000, half triangle wave is completed, and D10 decreases by 2 in each scan cycle that follows. When D10=D0=0, a complete triangle wave is completed, and M0=ON. In the next scan cycle, if X0 keeps ON, and M0=OFF, the next triangle wave process starts. If the energy flow falls, the output state ***D2*** is OFF, the output value ***D1*** is initialized as ***S1***, and a new triangle wave process starts.

2. You can use an external special module to convert the data into the analog waveform.

6.11.5 ABSD: Cam absolute control instruction

LAD: ├─┤ ├─[ABSD *(S1)* *(S2)* *(D)* *(S3)*]	**Applicable model**	**IVC2 LIVC3**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: ABSD *(S1)* *(S2)* *(D)* *(S3)*	**Step length**	**9**	

Operand	Type	Applicable soft element													Indexing
S1	INT		KnX	KnY	KnM	KnS			D		C	T		R	√
S2	INT										C				√
D	BOOL			Y	M	S									
S3	WORD	Constant													

- Operand description

S1: Start element number of storing the table data (rising edge, and falling edge). n＝4.

S2: Number of the counterused for monitoring the current value compared with the table data.

D: Outputting the number of the start bit element.

S3: Number ofrows in the table and number ofpoints of the outputted bit elements. Range: 1≤***S3***≤64.

- Function description

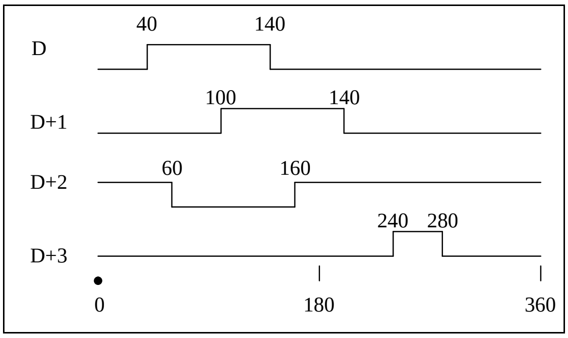
1. Comparing n rows of table data starting from ***S1***(occupying n rows×2 points) with current value ***S2*** of the counter, and performing ON/OFF control on the consecutive n-point ***D*** output.

2. Each rising/falling point can be changed by rewriting the data of ***S1***﹣***S1***＋n×2.

3. Firstly writing the data shown below into S1﹣S1＋2n＋1:

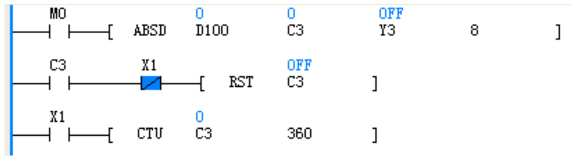
Rising point	Value example	Falling point	Value example	Output
S1	40	***S1***+1	140	***D***
S1+2	100	***S1***+3	200	***D***+1
S1+4	160	***S1***+5	60	***D***+2
S1+6	240	***S1***+7	280	***D***+3
…	…	…	…	…
S1+2n		***S1***+2n+1		***D***+n-1

4. After the instruction input is ON, the n point starting from ***D*** also changes as follows.



- Application instance

Outputting ON/OFF by rotating the platform once (0－360 degrees, rotation angle signal that indicates 1 degree per pulse).



Among them, M0 is the energy flow input, and X1 is a rotation angle signal indicating 1 degree per pulse.

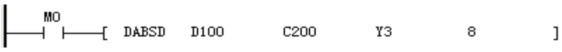
- Note

1. When the bit number of the bit soft elements is designated in ***S1***, k＝4.

2. When the counter number is designated in ***S2***, C0－C199 need to be designated for S2.

3. This instruction is affected by the scan cycle.

6.11.6 DABSD: Double word cam absolute control instruction

LAD:	Applicable model	IVC2L IVC3
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: DABSD *(S1)* *(S2)* *(D)* *(S3)*	Step length	11

Operand	Type	Applicable soft element													Indexing
S1	DINT		KnX	KnY	KnM	KnS			D		C	T		R	√
S2	DINT										C				√
D	BOOL			Y	M	S									
S3	WORD	Constant													

- Operand description

S1: Start element number of storing the table data (rising edge, and falling edge). n＝8.

S2: Number of the counter used for monitoring the current value compared with the table data. Range: C200﹣C255, and C301﹣C306.

D: Outputting the number of the start bit element.

S3: Number ofrows in the table and number of points of the outputted bit elements. Range: 1≤***S3***≤64.

- Function description

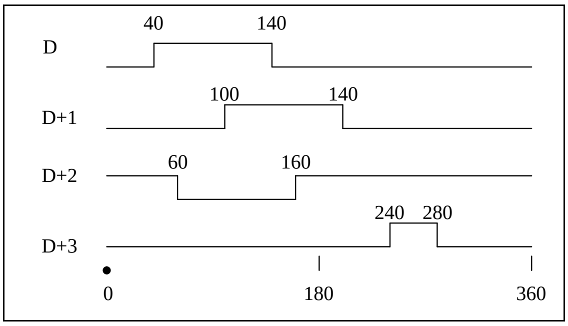
1. Comparing n rows of table data starting from ***S1***(occupying n rows×4 points) with current value ***S2*** of the counter, and performing ON/OFF control on the consecutive n-point ***D*** output.

2. Each rising/falling point can be changed by rewriting the data of [***S1***+1, ***S1***]﹣[***S1***+(n×2)+3, ***S1***+(n×2)+2].

3. Firstlyusing the transmission instruction, and writing the data shown below into [***S1***, ***S1***+1] ﹣ [***S1***, ***S1***+1]+4n+3.

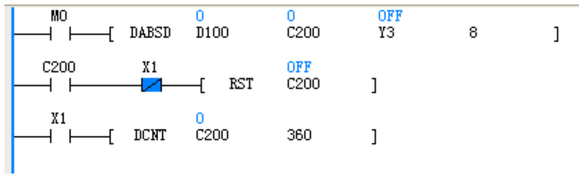
Rising point	Value example	Falling point	Value example	Output
[***S1***+1, ***S1***]	40	[***S1***+3, ***S1***+2]	140	***D***
[***S1***+5, ***S1***+4]	100	[***S1***+7, ***S1***+6]	200	***D***+1
[***S1***+9, ***S1***+8]	160	[***S1***+11, ***S1***+10]	60	***D***+2
[***S1***+13, ***S1***+12]	240	[***S1***+15, ***S1***+14]	280	***D***+3
……	……	……	……	……
[***S1***+4n+1, ***S1***+4n]		[***S1***+4n+3, ***S1***+4n+2]		***D***+n-1

4. After instruction input is ON, the n point starting with D will also change as below shows.



- Application instance

Outputting ON/OFF by rotating the platform once (0﹣360 degrees, rotation angle signal that indicates 1 degree per pulse).



Among them, M0 is the energy flow input, and X1 is a rotation angle signal indicating 1 degree per pulse.

Among them, M0 is the instruction input, and X1 is a rotation angle signal indicating 1 degree per pulse. At this time, the default value of SM200 is OFF, the DCNT instruction is incremented.

- Note

1. When the bit number of the bit soft elements is designated in ***S1***, n＝8.

2. When the counter number is designated in ***S2***,C200﹣C255 need to be designated for ***S2***.

3. This instruction is affected by the scan cycle.

6.11.7 ALT: Alternate output instruction

LAD: ├──┤ ├──[ALT (D)]	**Applicable model**	**IVC2L IVC3 IVC1L**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ALT (*D*)	**Step length**	**11**

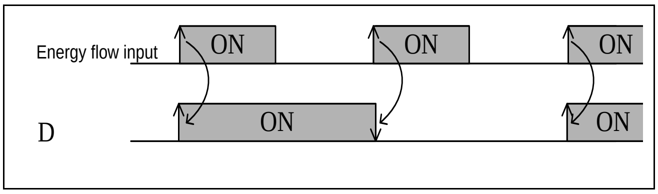
Operand	Type	Applicable soft element													Indexing
D	BOOL			Y	M	S									

- Operand description

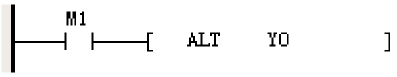
  D: Alternately outputted element address

- Function description

  When the energy flow is valid, soft elementsin each scan cycle act reversely, which is shown as below.



- Application instance



6.12 Communication instructions

6.12.1 MODBUS: Master station communication instruction

LAD: ├──┤ ├──[MODBUS (S1) (S2) (S3)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: Modbus (*S1*) (*S2*) (*S3*)	**Step length**	**8**

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	INT	D	V											R	
S3	INT	D												R	√

- Operand description

  S1: Designated communication channel

  S2: Start address of data to be transmitted

  S3: Start address for storing the received data

- Function description

  1. When being as a master station, and the input conditions are met, the system transmits the data stored in the address unit starting from ***S2***, receives data, and stores it to the address unit starting from ***S3***.

  2. When being a slave station, the system needs no instruction for receiving and transmitting data.

  3. This instruction is executed upon the rising edge.

- Note

  1. Data is transmitted through Modbus. Whether you set the RTU or ASCII mode, only RTU-format data is stored in units starting from S2. The start character, end character, and checksum are not stored. In the transmission process, the required start character, end character, and check bit are automatically added to the data.

  2. You do not need to set the length for the data to be transmitted. The system automatically transmits the data according to the internally specified length based on the instruction, as shown below:

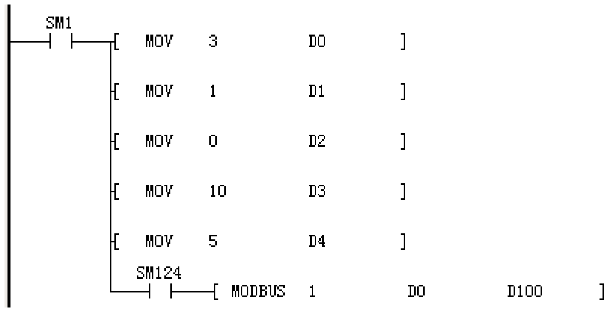
S2	Slave address
S2+1	Function code
S2+2	Data 1
	... ...
S2+N+1	Data N

3. Data is received through Modbus. Whether you set the mode to RTU or ASCII, the received data is stored in RTUformat. That is, when you set the mode to ASCII, the

system automaticallyconvertsthe data to the hex format, removes the startand end characters, and saves them in thedata area starting from ***S3***.

4. The sent and received data are stored in LSB of the word element, and MSB are not used.

- Application instance



LD SM1

MOV 3 D0

MOV 1 D1

MOV 0 D2

MOV 10 D3

MOV 5 D4

AND SM124

Modbus 1 D0 D100

1. Storing the data sent through the Modbus instruction into the element starting from D0.

2. Storing the received data in the elements starting from D100.

3. After receiving the data through Modbus, the system conducts CRC check, address check, and function code check. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

4. Serial idle flags of SM114 and SM124 can also be used in Modbus to indicate the communication state of Modbus.

Modbus communication error code tableis shown as follows.

Abnormal code	Meaning of abnormal code
0x01	Illegal function code
0x02	Illegal register address
0x03	Wrong number of data
0x10	Communication timeout. The communication time exceeds the preset communication time limit
0x11	Error in receiving data frame
0x12	Operand error, operand (mode ormaster/slave) setting error
0x13	Error occurs when the local station number is the same as the one set by the instruction.

For details about the application instance,refer to Chapter 10"Communication function guide".

6.12.2 IVFWD: Inverter forward instruction

LAD: IVFWD (S1) (S2)	Applicable model	IVC1 IVC1S IVC2L IVC3 IVC1L
	Influenced flag bit	
IL: IVFWD (*S1*) (*S2*)	Step length	6

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√

- Operand description

S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

S2*:* Address of the inverter (supportingthe broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

- Function description

1. Controlling the forward rotation of the inverter through the Modbus communication protocol.

2. This instruction is executed upon the rising edge.

- Note

The total number of Modbus, MODRW, and inverter communication instructions does not exceed 128.

- Application instance

LD M1

IVFWD 1 1

1. Setting the serial port 1 and inverter address #1, and controlling the forward rotation of the inverter through the Modbus communication protocol.

2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

The error code table of the inverter instruction communication is shown as follows.

Abnormal code	Meaning of abnormal code
0x1	Illegal function code
0x2	Illegal register address
0x3	Data error, namely data exceeds the range.
0x4	Slave station operation failure, including data is within the range, but an error occurs due to the invalid data.
0x5	The instruction is valid and in processing, mainly used to store the data to the non-volatile memory (NVM).

Abnormal code	Meaning of abnormal code
0x6	The slave station is busy, please try again later, mainly used to store the data to the non-volatile memory (NVM).
0x18	Information frame error: including the information length error, and check error.
0x20	The parameter is non-modifiable
0x21	The parameter cannot be modified in RUN (supporting EV3100 only)
0x22	The parameter is protected by a password

6.12.3 IVREV: Inverter reverse instruction

LAD: [IVREV *(S1)* *(S2)*]	**Applicable model**	**IVC1 IVC1S IVC2L IVC3 IVC1L**
	Influenced flag bit	
IL: IVREV *(S1)* *(S2)*	**Step length**	**6**

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√

- Operand description

  S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

  S2*:* Address of the inverter (supporting the broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

- Function description

  1. Controlling the reverse rotation of the inverter through the Modbus communication protocol.

  2. This instruction is executed upon the rising edge.

- Application instance

  1. Setting the serial port 1 and inverter address #1, and controlling the reverse rotation of the inverter through the Modbus communication protocol.

  2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.4 IVDFWD: Inverter jogging forward instruction

LAD: [IVDFWD *(S1)* *(S2)*]	**Applicable model**	**IVC1 IVC1S IVC2L IVC3 IVC1L**
	Influenced flag bit	
IL: IVDFWD (*S1*) *(S2)*	**Step length**	**6**

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√

- Operand description

  S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

  S2*:* Address of the inverter (supporting the broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

- Function description

1. Controlling the jogging forward of the inverter through the Modbus communication protocol.

2. This instruction is executed upon the rising edge.

- Application instance

LD M1
IVDFWD 1 1

1. Setting the serial port 1 and inverter address #1, and controlling the jogging forward of the inverter through the Modbus communication protocol.

2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.5 IVDREV: Inverter jogging reverse instruction

LAD:	Applicable model	IVC1 IVC1S IVC2L IVC3 IVC1L
IVDREV (S1) (S2)	Influenced flag bit	
IL: IVDREV (S1) (S2)	Step length	6

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√

- Operand description

S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

S2: Address of the inverter (supportingthe broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

- Function description

1. Controlling the jogging reverse of the inverter through the Modbus communication protocol.

2. This instruction is executed upon the rising edge.

- Application instance

LD M1
IVDREV 1 1

1. Setting the serial port 1 and inverter address #1, and controlling the jogging reverse of the inverter through the Modbus communication protocol.

2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.6 IVSTOP: Inverter stop instruction

LAD:	Applicable model	IVC1 IVC1S IVC2L IVC3 IVC1L
IVSTOP (S1) (S2) (S3)	Influenced flag bit	
IL: IVSTOP (S1) (S2) (S3)	Step length	8

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√
S3	WORD	Constant	D	V										R	√

- Operand description

S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

S2: Address of the inverter (supporting the broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

S3: Stop mode of the inverter

There are three stop modes: stop mode 0 (stop based on the preset deceleration time) , stop mode 1 (coast to stop), and stop mode 2 (stop based on the shortest deceleration time).

- Function description

1. Controlling the stop of the inverter through the Modbus communication protocol.
2. This instruction is executed upon the rising edge.

- Application instance

LD M1

IVSTOP 1 1 0

1. Setting the serial port 1 and inverter address #1, and stop mode 0 of the inverter (stop based on the preset deceleration time), and controlling the stop of the inverter through the Modbus communication protocol.
2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.7 IVFRQ: Inverter frequency setting instruction

LAD: 	Applicable model	IVC1 IVC1S IVC2LIVC3 IVC1L
	Influenced flag bit	
IL: IVFRQ (*S1*) (*S2*) (*S3*)	Step length	8

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√
S3	WORD	Constant	D	V										R	√

- Operand description

S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

S2: Address of the inverter (supportingthe broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

S3: Frequency of the inverter

- Function description

1. Controlling the operation frequency of the inverter through the Modbus communication protocol.
2. This instruction is executed upon the rising edge.

- Application instance

LD M1

IVFRQ 1 1 50

1. Setting the serial port 1 and inverter address #1, and the operation frequency of the inverter 50Hz, and controlling the operation frequency of the inverter through the Modbus communication protocol.
2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.8 IVWRT: Single register value writing instruction

LAD: 	Applicable model	IVC1 IVC1S IVC2L IVC3 IVC1L
	Influenced flag bit	
IL: IVWRT (*S1*) (*S2*) (*S3*) (*S4*)	Step length	10

Operand	Type	Applicable soft element	Indexing

S1	INT	Constant													
S2	WORD	Constant	D	V										R	√
S3	WORD	Constant	D	V										R	√
S4	WORD	Constant	D	V										R	√

- Operand description

  S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

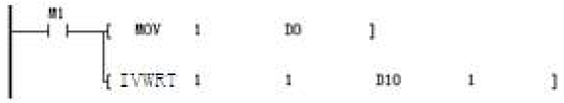
  ***S2**:* Address of the inverter (supporting the broadcast mode, and broadcast address is 00.Setting range of the slave station address: 1–247)

  S3: Register address

  S4: Register value
- Function description

  1. Writing a single register valuethrough the Modbus communication protocol.

  2. This instruction is executed upon the rising edge.
- Application instance



LD M1

MOV 1 D0

IVWRT 1 1 D10 1

1. Setting the serial port 1 and inverter address #1, and writing the register address 1 (digital frequency control) and register value 1 (do not save the set frequency at power outage), and writing the single register value through the Modbus communication protocol.

2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.9 IVRDST: Inverter state reading instruction

LAD: IVRDST *(S1)* *(S2)* *(S3)* *(D1)*	**Applicable model**	**IVC1 IVC1S IVC2L IVC3 IVC1L**
	Influenced flag bit	
IL: IVRDST *(S1)* *(S2)* *(S3)* *(D1)*	**Step length**	**10**

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√
S3	WORD	Constant	D	V										R	√
D1	WORD	D													√

- Operand description

  S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

  ***S2**:* Address of the inverter (not supportingthe broadcast mode.Setting range of the slave station address: 1–247)

  S3:State information to be read

  0: Running state word. 1: Actual operation value in the current main setting. 2: Inverter model. 3: Output current. 4: Output voltage. 5: Running speed. 6: Operation fault information.

  D1: Storage address of the returned state information
- Function description

  1. Reading the state of inverter through the Modbus communication protocol.

  2. This instruction is executed upon the rising edge.
- Application instance

LD M1

IVRDST 1 1 1 D0

1. Setting the serial port 1 and inverter address #1, reading the state information selection 1 (actual operation value in the current main setting), setting D0 as the storage register for the returned state information,

and reading the state of the inverter through the Modbus communication protocol.

2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.10 IVRD: Invertersingle register value reading instruction

LAD: ├┤ ├─[IVRD (S1) (S2) (S3) (D1)]	**Applicable model**	**IVC1 IVC1S IVC2L IVC3 IVC1L**
	Influenced flag bit	
IL: IVRD *(S1) (S2) (S3) (D1)*	**Step length**	**10**

Operand	Type	Applicable soft element													Indexing
S1	INT	Constant													
S2	WORD	Constant	D	V										R	√
S3	WORD	Constant	D	V										R	√
D1	WORD	D													√

- Operand description

  S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2)

  S2*:* Address of the inverter (not supportingthe broadcast mode.Setting range of the slave station address: 1–247)

  S3: Register address to read

  D1: Storage address of the returned value

- Function description

  1. Reading a single register value of the inverter through the Modbus communication protocol.

  2. This instruction is executed upon the rising edge.

- Application instance

LD M1

MOV 2 D10

IVRD 1 1 D10 D20

1. Setting the serial port 1 and inverter address #1, reading the register address 2(initially set frequency of the inverter), setting D20 as the storage register for the returned value, and reading the single register value of the inverter through the Modbus communication protocol.

2. After receiving the data, the inverter conducts CRC check, address check, and function code check. After the communication is done, the communication completion flag (SM135) is set. If there is any error, the error flag (SM136) is set, and the error details are recorded in the special register SD139.

6.12.11 XMT: Free-port sending instruction

LAD: ├┤ ├─[XMT (S1) (S2) (S3)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: XMT *(S1) (S2) (S3)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant														
S2	WORD	D	V												R	
S3	INT	Constant	KnX	KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	R	

- Operand description

  S1: Designated communication channel. For IVC1 and IVC2L, the value range is 0 and 1. For IVC1L and IVC3, the value range is 0, 1, and 2.

  S2: Start address of the data to be sent

  S3: Number of bytes to be sent

- Function description

  When the energy flow is turned on and the communication conditions are met, the data are sent through the designated channel and address.

- Note

  1. Size of the communication frame: Size of the communication frame: The end character of the frame transmitted cannot exceed D7999 or V63, depending on the selected element type (D or V).

  2. In the case of a shutdown, the transmission is stopped.

- Special register

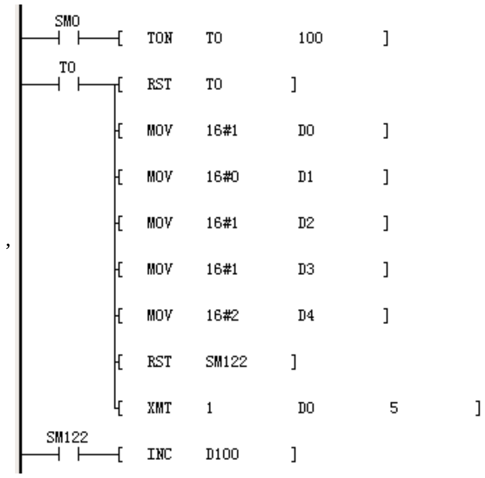
  1. SM110/SM120: Transmission enable flag. This bit is set when the XMT instruction is used, cleared after the transmission is over, and reset when the current transmission is terminated.

  2. SM112/SM122: Transmission completion flag. When it is judged that the transmission is completed, the transmission completion flag is set.

  3. SM114/SM124: Idle flag. When the serial port has no communication task, it is set. It can be used as the check bit for the communication.

  4. For details about the application instance, refer to Chapter 10"Communication function guide".

- Application instance




```
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 one data frame at 10s interval.

The following data are sent through the serial port 1:

01	00	01	01	02

1. First, setting port 1 in the system block as the free port, and setting the baud rate, parity check, data bit, and stop bit.

2. Writing the data to be sent into the transmission buffer area. For IVC2, only low bytes of the word element are sent.

3. Clearing the transmission completion flag (SM122) before sending the data.

4. Setting the transmission completion flag (SM122)when the transmission is done.

6.12.12 RCV: Free-port receiving instruction

LAD: RCV *(S1)* *(D)* *(S2)*	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	Influenced flag bit	
IL: RCV *(S1)* *(D)* *(S2)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant														
D	WORD	D	V												R	
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM		D	SD	C	T	V	Z	R	

- Operand description

  S1: Designated communication channel. For IVC1 and IVC2L, the value range is 0 and 1. For IVC1L and IVC3, the value range is 0, 1, and 2.

  D: Start address for storing the received data

  S2: Max. number of received bytes

- Function description

  When the energy flow is turned on and the communication conditions are met, the data are received through the designated channel and address.

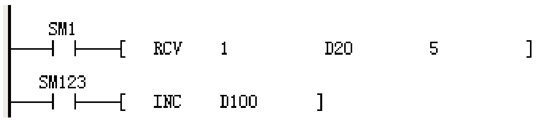
- Note

  1. Size of the communication frame: The end character of the frame received cannot exceed D7999 or V63,depending on the selected element type (D or V).

  2. In the case of a shutdown, the receiving is stopped.

  3. The value range of ***S1***: 0, 1, and 2.

- Application instance



LD SM1

RCV 1 D20 5

LD SM123

INC D100

1. When the energy flow is conducted, the RCV instruction is valid continuously. If you want to receive the data only once, you can use a rising edge or a special register that is valid only once, such as SM1, as the energy flow input.

2. For details about the application instance, refer to Chapter 10"Communication function guide".

- Special registers

  SM111 (SM121):Receiving the enable flag. This bit is set when the RCV instruction is used, cleared after the receiving is over, and reset when the current receiving is terminated.

  SM113 (SM123): Receiving the completion flag. When receiving is done, the receiving completion flag is set.

  SM114 (SM124): Idle flag. this flag is set when there is no communication task on the serial port, and it can be used as the check bit for the communication.

  SD111 (SD121): Start character which can be set in the system block.

  SD112 (SD122): End character which can be set in the system block.

  SD113 (SD123): Inter-character timeout time, namely the max. interval between receiving two characters, which can be set in the system block.

  SD114 (SD124): Frame time-out time, namely the time from when the energy flow is conducted to the time when the receiving ends, which can be set in the system block.

  SD115 (SD125): Receiving completion information code. The definitionsof the data bits are shown below:

Flag for ending user receiving	Flag for receiving the specified end word	Flag for receiving the maximum number of characters	Inter-character timeout flag	(Frame) receiving timeout flag	Parity check error flag	Reserved (You do not need to take them into consideration.)
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6－15

  SD116 (SD126): The characters currently received

  SD117 (SD127): Total number of the characters currently received.

6.12.13 MODRW: Modbusread/write instruction

LAD: ┤├ ┤├─[MODRW (S1) (S2) (S3) (S4) (S5) (D)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: MODRW *(S1) (S2) (S3) (S4) (S5) (D)*	**Step length**	**14**

Operand	Type	Applicable soft element	Indexing

S1	INT	Constant													
S2	WORD	Constant	D	V										R	√
S3	WORD	Constant	D	V										R	√
S4	WORD	Constant	D	V										R	√
S5	WORD	Constant	D	V										R	√
D	WORD		D											R	√

- Operand description

S1:Designated communication channel (IVC1 and IVC2L can only specify channel 1 while IVC1L and IVC3 can specify channel 1 and channel 2).

***S2**:* Address (Setting range of the slaveaddress: 1–247, and the broadcast address is applicable to the write element).

S3: Function code. IVC1 supports 01 (read coil), 02 (read discrete input), 03 (read register), 04 (read input register), 05 (write single coil), 06 (write single register), 15 (write multiple coils) , 16 (write multiple registers)

IVC2L and IVC3 support 01 (read coil), 03 (read register), 15 (write multiple coils), and 16 (write multiple registers).

Inverter function code only supports 03 10 01 0f.

S4: Start address of to-be-read/written inverter elements.

S5: Number of to-be-read/written elements. Number of to-be-read/written elements of IVC1 is limited by the max frame length (256) of RTU, as shown below.

Function code	Name	Number of elements	Number ofrequired D elements
01	Read coils	1－2000	(S5＋15)/16
02	Read discrete input	1－2000	(S5＋15)/16
03	Read registers	1－125	S5
04	Read input registers	1－125	S5
05	Write single coil	0(fixed)	1
06	Write single register	0(fixed)	1
15	Write multiple coils	1－1968	(S5＋15)/16
16	Write multiple registers	1－123	S5

*The value of 05, 06 and S5 must be 0

The number of IVC2 and IVC3 to-be-read/written elements (S5≤16), word elements and bit elements cannot exceed 16 respectively, and all the bit elements are stored as one word.

The number of the word and bit elements cannot exceed 16 respectively, and all the bit elements are stored as one word.

D1: Storage address of to-be-read/written elements. For details about the number of elements required by IVC1, refer to the above table.

- Function description

Sending the message and receiving the returned data when the energy flow is valid.

- Note

For IVC2 and IVC3

1. The number of elements cannot exceed 16.

2. Reading a maximum of 16 bit elements. The bit elements with small addresses are stored in LSB, and one byte stores 16 bits.

3. The returned abnormal code is the same as the Modbus instruction.

- Application instance

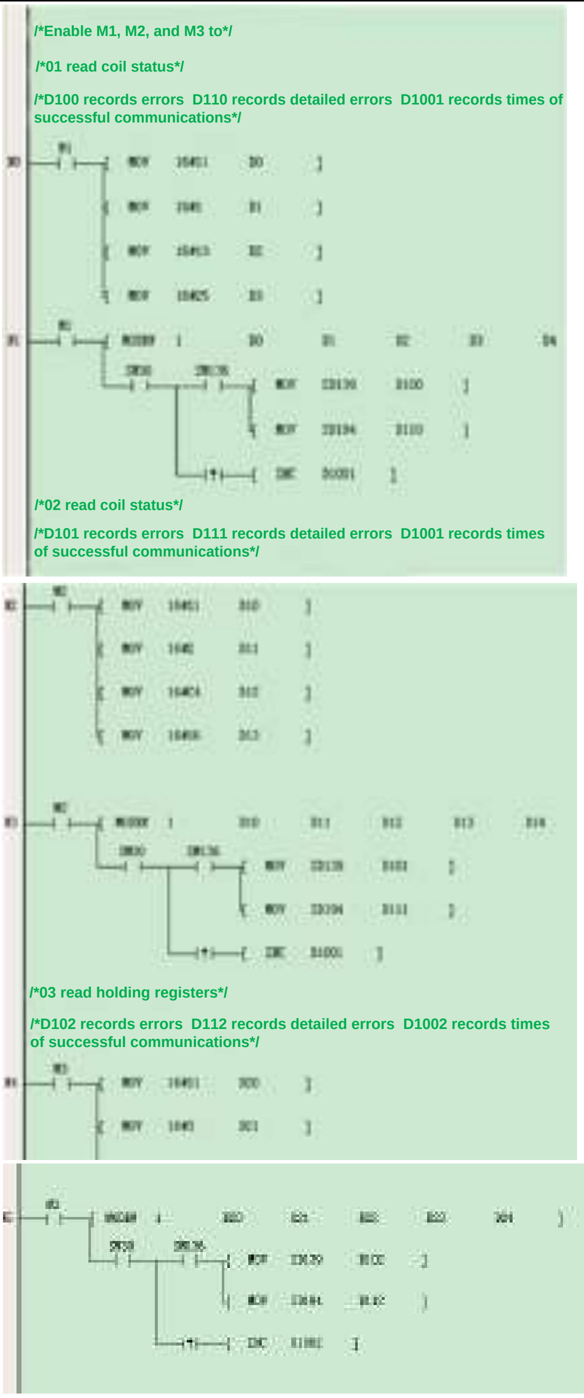
* The following application instance is currently only valid for the IVC1 series PLCs.

1. Standard polling

This instance is a simple polling instance, sets M1, M2, and M3, and three MODRW instruction access the device in turn according to the settings.

At runtime, if any one of M elements is reset, its corresponding MODRW instruction exits polling, but other MODRWs are still executed in polling mode. For instance, if M2 is reset, then the MODRW instructions corresponding to M1 and M3 access the device in turn.

Similarly, it is viable to insert one MODRW instruction during running. For instance, if M2 is set again, thenthree MODRW instructions access the device in turn.



In the above programs, SM30 reflects whether the MODRW instruction has been executed. After the MODRW instruction is executed, SM30 is set, and when the MODRW instruction enters the next execution, SM30 is reset. The time sequence diagram in the instance 2 illustrates the difference between SM30 and SM135.

When multiple MODRW instructions occur in the program, SM30 can be used to reflect the execution states of multiple MODRW instructions. In the above program, each SM30 reflects the state of the previous line of the MODRW instruction, and the use of multiple SM30 elements does not affect each other.

When an error occursin the MODRW instruction,SM136 is set, and SD139 and SD194 indicate its error codes.The values of SM136, SD139 and SD194 can be changed by other MODRW instructions, so their state need to be recorded before the next MODRW instruction is executed.

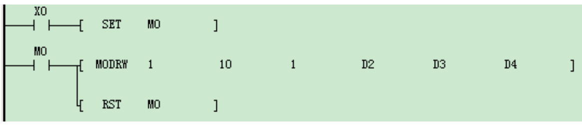
MODRW instruction error code is listed as below:

Code	Error name	Detailed description
1	Illegal function code	
2	Illegal address	
3	Illegal data	
4-15	Reserved	
16	Communication timeout	The communication time exceeds the max. communication time set by the user
17	Reserved	
18	Parameter setting error	Parameter (mode or master/slave) setting error
19	Parameter S2 error, namely slave address parameter error	The station no. is the same with the one set by the instruction, or address overlimit error occurs
20	Parameter D error, namely element address overflow	Element address overflows (The amount of the received or transmitted data exceeds the storage space of the element)
21	Instruction execution failed	
22	Address does not match	The received slave address does not match the requested slave address, and error code element stores the received slave address
23	Function code does not match	The received function code does not match the requested function code, and the error code element stores the received function code
24	Info frame error	Info frame error: it means element starting address does not match, the error code element stores the received element start address
25	Data length does not match	The received data length does not comply with protocol or the element number exceed the max limit specified by this function code
26	CRC/LRC check error	
27	Reserved	
28	Parameter S3 error, namely parameter error of the element address	Setting error in the start address of the instruction parameter element

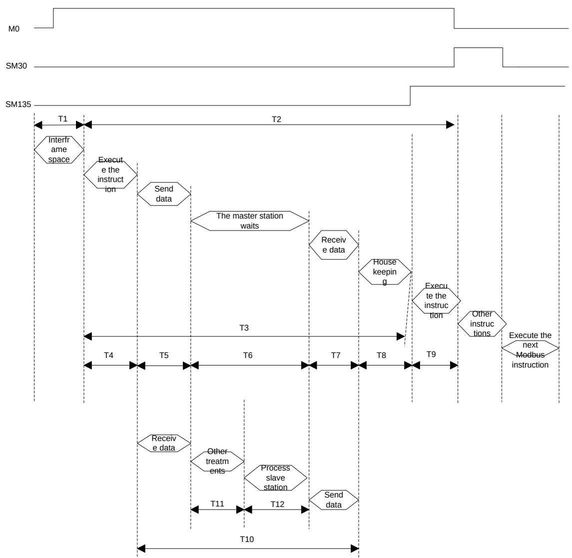
29	Parameter S4 error, namely parameter error of the function codes	Instruction parameters set unsupported function codes or illegal function codes
30	Parameter S5 error, namely setting error in the number of the elements	Setting error in the instruction parameter elements
31	Reserved	
32	The parameter is non-modifiable	The parameter is non-modifiable
33	The parameter is non-modifiable during running	The parameter is non-modifiable during running (only EV3000 support)
34	Parameter is protected by a password	The parameter is protected by a password

2. Link time

You can use the following LAD to realize a communication between a Modbus master and slave stations. For a complete communication, the required time for each stage is shown below:



One complete Modbus communication time (T_m) is comprised of T_1 and T_1, namely



$$T_m = T_1 + T_2$$

In which, T_1 is guaranteed by the user, according to the Modbus communication protocol, the inter-frame interval time needs to be 3.5 character at least.

The length of one character is: Start bit (1 bit)＋data length (7 bits or 8 bits)＋check (0 bit or 1 bit)＋stop bit (1 bit or 2 bits).

$$T_2 = (INT(\frac{T_3}{T_s}) + 1))T_s$$

In which, T_s is the max. scan cycle of the PLC.

$$T_3 = T_4 + T_5 + T_6 + T_7 + T_8 + T_9$$

T_4, T_8, and T_9 require less than 1 ms time.

$$T_5 = \frac{\text{Number of bytes to be sent} \times \text{Character length}}{\text{Baudrate (bps)}} \times 1000(\text{ms}) + 1\text{ms}$$

T_6: The waiting time of the master station depends on the slave station, which cannot exceed the timeout timeof the preset master mode.

$$T_7 = \frac{\text{Number of bytes to be received} \times \text{Character length}}{\text{Baudrate (bps)}} \times 1000(\text{ms}) + 1\text{ms}$$

The processing time of the slave station can be calculated by the following formula:

$$T_{10} = T_5 + T_{11} + T_{12} + T_7$$

In which, the max. scan time of T_{11}

T_{12} requires less than 1 ms time

For instance: Setting the communication specification as 19200, even parity, 8-bit data bit, 1-bit stop bit, RTU transmission mode, 10 characters to be sent, and 20 characters to be received. Then the processing time of the master station is calculated as follows:

$$T_5 = \frac{10 \times 10}{19200} \times 1000 + 1 = 6.2ms$$

$$T_7 = \frac{20 \times 10}{19200} \times 1000 + 1 = 11.4ms$$

$$T_4 = T_8 = T_9 \approx 1ms$$

Suppose $T_6 = 35ms$, then

$$T_3 = 1 + 6.2 + 35 + 11.4 + 1 + 1 = 55.6ms$$

Suppose the max. scan cycle is 10 ms

Then

$$T_2 = (INT(\frac{55.6}{15}) + 1))15 = 60ms$$

The processing time of the slave station

$$T_{10} = 6.2 + 15 + 1 + 11.4 = 33.6ms$$

6.12.14 CANNMT

LAD: 		**Applicable model**	**IVC2L IVC3**
		Influenced flag bit	
IL: CANNMT *(S1)*		**Step length**	**3**

Operand	Type	Applicable soft element													Indexing
S1	WORD		D												√

- Operand description

  S1:Switching the state. The value range: 1-4. 1 indicates resetting CANopen communication; 2 indicates resetting CANopen node; 3 indicates switching to pre-processing mode; 4 indicates switching to the run mode.
- Function description

  Sending the message, and putting the CANopen network into the specified state when the energy flow is valid.
- Note

  When the instruction is being executed, if the PLCs change RUN to STOP, the instruction may fail to complete.
- Application instance

  SM440 CANopen instruction execution completed (=1 execution completed, =0 other case).

  SM441 CANopen instruction execution error (=1 instruction error, =0 no error).

  SM442 CANopen instruction is being executed (=1 instruction is being executed, =0 no instruction is executed), mainly preventing multiple CANopen instructions from being executed at the same time.

6.12.15 CANSDORD

LAD: 		**Applicable model**	**IVC2L IVC3**
		Influenced flag bit	
IL: CANSDORD *(S1) (S2) (S3) (S4) (D1)*		**Step length**	**12**

Operand	Type	Applicable soft element													Indexing
S1	WORD	Constant	D												√
S2	WORD	Constant	D												√
S3	WORD	Constant	D												√
S4	WORD	Constant	D												√
D1	WORD	D													√

- Operand description

  S1: Address range of the device: 1-126.

  S2: SDO index.

  S3: SDO subindex.

  S4: The length of to-be-readdata(1, 2, 4 indicates byte, word, and double word respectively).

  D1: The storage address of data to be read back(Bytes and words occupy only 16 bits).
- Function description

  Sending the message, and reading the index data of the specified nodes when the energy flow is valid.
- Note

  When the instruction is being executed, if the PLC changes from RUN to STOP, the instruction may fail to complete. You need to make sure that the read index and subindex are valid, otherwise error information is returned.
- Application instance

  SM440 CANopen instruction execution completed (=1 execution completed, =0 other case).

SM441 CANopen instruction execution error (=1 instruction error, =0 no error).

SM442 CANopen instruction is being executed (=1 instruction is being executed, =0 no instruction is executed), mainly preventing multiple CANopen instructions from being executed at the same time.

6.12.16 CANSDOWR

LAD: [CANSDOWR 2 16#1017 16#0 2 D1200]	**Applicable model**	**IVC2L IVC3**
	Influenced flag bit	
IL: CANSDOWR *(S1) (S2) (S3) (S4) (D1)*	**Step length**	**12**

Operand	Type	Applicable soft element													Indexing
S1	WORD	Constant	D												√
S2	WORD	Constant	D												√
S3	WORD	Constant	D												√
S4	WORD	Constant	D												√
D1	WORD	D													√

- Operand description

  S1: Address range of the device: 1-126.

  S2: SDO index.

  S3: SDO subindex.

  S4: The length of to-be-read data(1, 2, 4 indicates byte, word, and double word respectively).

  D1: The storage address of data to be written (Bytes and words occupy only 16 bits).

- Function description

  Sending the message, and writing the index data of the specified nodes when the energy flow is valid.

- Note

  When the instruction is being executed, if the PLC changesfrom RUN to STOP, the instruction may fail to complete. You need to make sure that the index and subindex are valid, otherwise error information is returned.

- Application instance

  SM440 CANopen instruction execution completed (=1 execution completed, =0 other case).

  SM441 CANopen instruction execution error (=1 instruction error, =0 no error).

  SM442 CANopen instruction is being executed (=1 instruction is being executed, =0 no instruction is executed), mainly preventing multiple CANopen instructions from being executed at the same time.

6.13 Check instructions

6.13.1 CCITT: Check instruction

LAD: [CCITT *(S1)* *(S2)* *(D)*]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: CCITT *(S1) (S2) (D)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	WORD								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD								D				V		R	√

- Operand description

  S1: Start unit ofdata to be checked

S2: Number of data to be checked (***S2***≥0, otherwise the system report an operand error)

D: Check result

- Function description

1. Performing the CCITT check operation on ***S2*** data starting from (***S1***) , and assigning the operation result to ***D***.

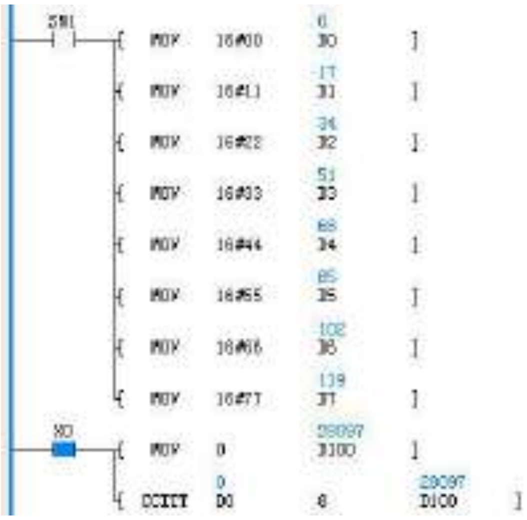
2. The polynomial of the CCITT check algorithm is: X^16＋X^12＋X^5＋1.

- Note

1. The system usually brings the content of ***D*** before the instruction execution into the operation when executing the instruction each time, so ***D*** needs to be initialized before executing the instruction.

2. Data in the data check area starting from ***S2*** unit are stored in byte mode by default, that is, the high byte is taken as 0, and the check result is 16bits.

- Application instance



```
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
```

Performing the CCITT check operation on 8 data starting from D0, and assigning the obtained result to D100 when X0=ON.

6.13.2 CRC16: Check instruction

LAD: CRC16 (*S1*) (*S2*) (*D*)		Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
		Influenced flag bit	
IL: CRC16 (*S1*) (*S2*) (*D*)		Step length	7

Operand	Type	Applicable soft element														Indexing
S1	WORD								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD								D				V		R	√

- Operand description

S1: Start unit ofdata to be checked

S2: Number ofdata to be checked (***S2***≥0, otherwise the system report an operand error)

D: Check result

- Function description

1. Performing the CRC16 check operation on ***S2*** data starting from (***S1***) , and assigning the operation result to ***D***.

2. The polynomial of the CRC16 check algorithm is: X^16＋X^15＋X^2＋1.

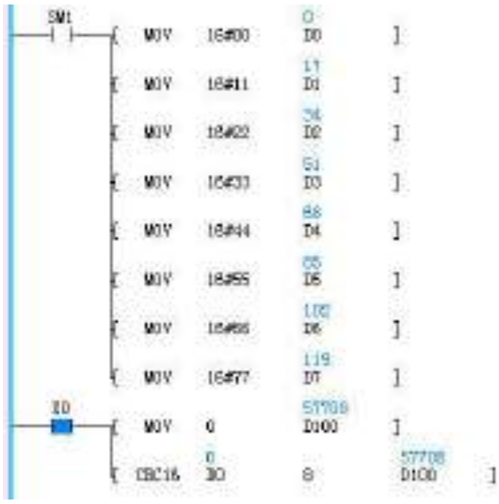
- Note

1. The system usually brings the content of ***D*** before the instruction execution into the operation when executing the instruction each time, so ***D*** needs to be initialized before executing the instruction.

2. Assigning 16#FFFF to the initial value of D element (checksum), and swopping the high/low bytes (high 8 bits and low 8 bits) when you use the standard Modbus CRC check.

3. Data in the data check area starting from ***S2*** unit are stored in byte mode by default, that is, the high byte is taken as 0, and the check result is 16bits.

- Application instance



```
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
```

Performing theCRC16 check operation on 8 data starting from D0, and assigning the obtained result to D100 when X0=ON.

6.13.3 LRC: Check instruction

LAD: LRC (S1) (S2) (D)		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: LRC *(S1)* *(S2)* *(D)*		**Step length**	**7**

Operand	Type	Applicable soft element															Indexing
S1	WORD								D				V		R	√
S2	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	WORD								D				V		R	√

- Operand description

  S1: Start unit of data to be checked

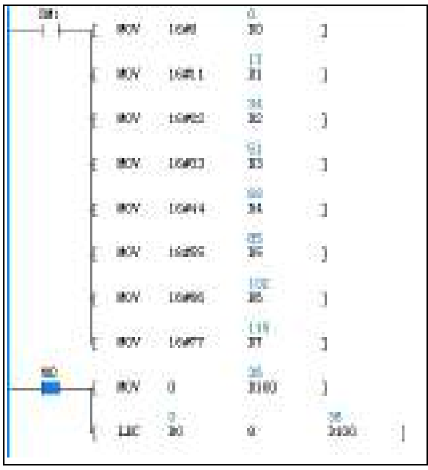
  S2: Number of data to be checked (***S2***≥0, otherwise the system report an operand error)

  D: Check result
- Function description

  Performing theLRC check operation on ***S2*** data starting from (***S1***) , and assigning the operation result to ***D***.
- Note

  1. The system usually brings the content of ***D*** before the instruction execution into the operation when executing the instruction each time, so ***D*** needs to be initialized before executing the instruction.

  2. Data in the data check area starting from ***S2*** unit are stored in byte mode by default, that is, the high byte is taken as 0, and the check result is 8 bits stored in the low bytes of ***D***.
- Application instance



```
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
```

Performing the LRC check operation on 8 data starting from D0, and assigning the obtained result to D100 when X0=ON.

6.14 Enhanced bit processing instructions

6.14.1 ZRST: Instruction for resetting bits to 0 in batch

LAD: ZRST (D) (S)		**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
		Influenced flag bit	
IL: ZRST *(D)* *(S)*		**Step length**	**5**

Operand	Type	Applicable soft element															Indexing
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

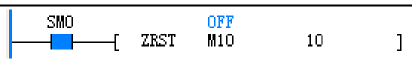
  S: Source operand
- Function description

  Clearing ***S*** consecutive bit elements starting from ***D*** when the energy flow is valid.
- Note

  1. Clearing the counter value in C element when the C element is cleared.

2. Clearing the timing value in T element when the T element is cleared.

- Application instance



LD SM0
ZRST M10 10

Clearing the data of 10 units starting from M10 (namely M10, M11, M12…M19) when SM0=ON.

6.14.2 ZSET: Instruction for resetting bits in batch

LAD: ZSET (D) (S)		Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
		Influenced flag bit	
IL: ZSET (D) (S)		Step length	5

Operand	Type	Applicable soft element														Indexing
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

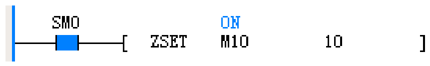
D: Destination operand

S: Source operand

- Function description

Setting ***S*** consecutive bit elements starting from ***D*** to 1when the energy flow is valid.

- Application instance



LD SM0
ZSET M10 10

Setting the data of 10 units starting from M10 (namely M10, M11, M12…M19) to 1 when SM0=ON.

6.14.3 DECO: Decoding instruction

LAD: DECO (S) (D)		Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
		Influenced flag bit	
IL: DECO (S) (D)		Step length	5

Operand	Type	Applicable soft element														Indexing
S	WORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

S: Source operand

D: Destination operand

- Function description

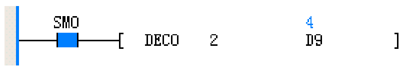
Setting the bit ***S*** in the word element ***D*** to 1, and clearing other bits when the energy flow is valid.

- Note

1. The valid range of ***S*** is 0 to 15.

2. When ***S*** is larger than 15 or less than 0, and the energy flow is valid, the value of ***D*** is not changed, but the system reports a value error of the instruction operand.

- Application instance



LD SM0
DECO 2 D9

Setting the bit 2 in D9 to 1, and clearing other bits when the energy flow is valid.

6.14.4 ENCO: Encoding instruction

LAD: ENCO (S) (D)		Applicable model	IVC2L IVC1 IVC1S IVC3IVC1L
		Influenced flag bit	
IL: ENCO (S) (D)		Step length	5

Operand	Type	Applicable soft element	Indexing

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

  Writing the bit number whose value is "1" in the word element ***S*** into ***D*** when the energy flow is valid.

- Note

  When the value of multiple bits in ***S*** are "1", the smallest position number will be written to ***D***, as shown in the following figure.

  M0 —[ENCO 2#1100 D0] (2)

- Application instance

  M0 —[ENCO 2#0010 D0] (1)

  LD M0
  ENCO 2#0010 D0

  Operand 1 is 2#0010, and the bit 1 is "1", hence the result is 1, which is written into D0 when the energy flow is valid.

6.14.5 BITS: Instruction for counting on bit in word

LAD: —[BITS (S) (D)]									Applicable model			IVC2L IVC1 IVC1S IVC3 IVC1L				
									Influenced flag bit							
IL: BITS (S) (D)									Step length			5				
Operand	Type	Applicable soft element														Indexing
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√
D	INT			KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

  S: Source operand; ***D***: Destination operand

- Function description

  Counting the number of "1" bitin the operand ***S***, and storing the statistical result in the operand ***D*** when the energy flow is valid.

- Application instance

  SM0 —[BITS 16#F0F0 D1] (8)

  LD SM0
  BITS 16#F0F0 D1

  When the energy flow is valid, ***S*** in the BITS instruction is constant 16#F0F0 which includes 8 bits whose value is "1" (ON state). The calculation result is 8 and stored in ***D*** (D1).

6.14.6 DBITS: Instruction for counting on bit in double word

LAD: —[DBITS (S) (D)]									Applicable model			IVC2L IVC1 IVC1S IVC3 IVC1L				
									Influenced flag bit							
IL: DBITS (S) (D)									Step length			6				
Operand	Type	Applicable soft element														Indexing
S	DWORD	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		R	√
D	INT			KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

  S: Source operand; ***D***: Destination operand

- Function description

  Counting the number of "1" bit in the double word ***S***, and storing the statistical result in ***D*** when the energy flow is valid.

- Application instance

  SM0 —[DBITS 16#FF0FF D10] (16)

  LD SM0
  DBITS 16#FF0FF D10

  When the energy flow is valid, ***S*** in the BITS instruction is constant 16#FF0FF which includes 16 bits whose value is "1" (ON state). The calculation result is 16 and stored in ***D*** (D10).

6.14.7 BON: Instruction for judging on bit in word

LAD: ─┤├──[BON (S1) (D) (S2)]	**Applicable model**	**IVC3**
	Influenced flag bit	
IL: BON *(S1)* *(D)* *(S2)*	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V			√
D	BOOL		Y	M	S											
S2	INT								D				V		R	

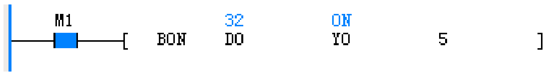
- Operand description

  S: Source operand; ***D***: Destination operand

- Function description

  Counting the state of the bit ***S2*** in the word ***S1***, and outputting the obtained result to ***D*** when the energy flow is valid.

- Application instance



LD M1
BON D0 Y0 5

When the energy flow is valid, ***S1*** in the BON instruction is a constant D0, and its fifth bit state (ON) is output to D (Y0).

6.15 Word contact instructions

6.15.1 BLD: Word bit contact LD instruction

LAD: ├──┤ BLD (S1) (S2) ├─( )	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: BLD *(S1)* *(S2)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
S1	WORD		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: Designated bit,0≤***S2***≤15, otherwise the system reports an operand error.

- Function description

  Taking the state of the bit ***S2*** in ***S1***, and using the obtained valueto drive the subsequent operation.

- Application instance

BLD D0 5
OUT Y0

Taking the state of the bit 5 (ON) in D0 (1000: 2#0000001111101000), and using the obtained value to determine the state of Y0 in the subsequent operation.

6.15.2 BLDI: Word bit contact LDI instruction

LAD: ├──┤ BLDI (S1) (S2) ├─( )	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: BLDI *(S1)* *(S2)*	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
S1	WORD		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: Designated bit,0≤***S2***≤15, otherwise the system reports an operand error.

- Function description

  Performing the NOT operation on the state of the bit ***S2*** in ***S1***, and using the obtained value to drive the subsequent operation.

- Application instance

BLDI D0 5
OUT Y0

Performing the NOT operation (OFF) on the state of the bit 5 (ON) in D0 (1000: 2#0000001111101000), and using the obtained value to determine the output state of Y0 in the subsequent operation.

6.15.3 BAND: Word bit contact AND instruction

LAD[note]: BLD (S1) (S2) Note: The logical relationship has been manifested in the diagram, so the BAND instruction is displayed as BLD in LAD															**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
															Influenced flag bit	
IL: BAND *(S1)* *(S2)*															**Step length**	**5**
Operand	Type	Applicable soft element														Indexing
S1	WORD		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: Designated bit (0≤***S2***≤15, otherwise the system reports an operand error)

- Function description

  Taking the state of the bit ***S2*** in ***S1***, and using the obtained value in serial connection with other nodes to drive the subsequent operation.

- Application instance

LD X0
BAND D0 5
OUT Y0

Taking the state of the bit 5 (ON) in D0 (1000: 2#0000001111101000), and using the obtained value in serial connection with other nodes (X0=ON) to determine the output state ofY0 in the subsequent operation.

6.15.4 BANI: Word bit contact ANI instruction

LAD[note]: BLDI (S1) (S2) Note: The logical relationship has been manifested in the diagram, so the BANI instruction is displayed as BLDI in LAD															**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
															Influenced flag bit	
IL: BANI *(S1)* *(S2)*															**Step length**	**5**
Operand	Type	Applicable soft element														Indexing
S1	WORD		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: Designated bit (0≤***S2***≤15, otherwise the system reports an operand error)

- Function description

  Performing the NOT operation on the state of the bit ***S2*** in ***S1***, and using the obtained value in serial connection with other nodes to drive the subsequent operation.

- Application instance

Performing the NOT operation (OFF) on the state of the bit 5 (ON) in

LD X0
BANI D0 5
OUT Y0

D0（1000：2#0000001111101000）, and using the obtained value in serial connection with other nodes (X0=ON) to determine the output state of Y0 in the subsequent operation.

6.15.5 BOR: Word bit contact OR instruction

LAD[note]:															
 Note: The logical relationship has been manifested in the diagram, so the BOR instruction is displayed as BLD in LAD															
Applicable model	**IVC2L IVC1 IVC1S IVC3 IVC1L**														
Influenced flag bit															
IL: BOR *(S1)* *(S2)*															
Step length	**5**														
Operand	Type	Applicable soft element													Indexing
S1	WORD		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: Assignment bit (0≤***S2***≤15, otherwise operand error will be reported)

- Function description

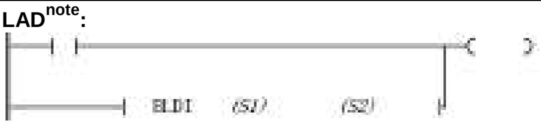
  Taking the state of the bit ***S2*** in ***S1***, and using the obtained value in parallel connection with other nodes to drive the subsequent operation.

- Application instance

LD X0
BOR D0 5
OUT Y0

Taking the state of the bit 5 (ON) inD0 (1000: 2#0000001111101000), and using the obtained value in parallel connection with other nodes (X0=ON) to determine the output state ofY0 in the subsequent operation.

6.15.6 BORI: Word bit contact ORI instruction

LAD[note]:															
 Note: The logical relationship has been manifested in the diagram, so the BORI instruction is displayed as BLDI in LAD															
Applicable model	**IVC2L IVC1 IVC1S IVC3 IVC1L**														
Influenced flag bit															
IL: BORI *(S1)* *(S2)*															
Step length	**5**														
Operand	Type	Applicable soft element													Indexing
S1	WORD		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: Designated bit (0≤***S2***≤15, otherwise the system reports an operand error)

- Function description

  Performing the NOT operation on the state of the bit ***S2*** in ***S1***, and using the obtained value in parallel connection with other nodes to drive the subsequent operation.

- Application instance

LD X0
BORI D0 5
OUT Y0

Performing the NOT operation (OFF) on the state of the bit 5 (ON) in D0 (1000: 2#0000001111101000), and using the obtained value in parallel connection with other nodes (X0=ON) to determine the output state of Y0 in the subsequent operation.

6.15.7 BOUT: Word bit coil output instruction

LAD: ├─┤ ├─[BOUT (D) (S)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: BOUT (*D*) (*S*)	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
D	WORD			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

  S2: Designated bit (0≤***S2***≤15, otherwise the system reports an operand error)
- Function description

  Assigningthe state of the current energy flow to the bit ***S*** of ***D***.
- Application instance

LD X0
BOUT D0 4

Assigning the state of the current energy flow(X0=ON) to the bit 4 in D0 (1000:2#0000001111101000). After execution: D0=1016 (2#0000001111111000).

6.15.8 BSET: Word bit coil setinstruction

LAD: ├─┤ ├─[BSET (D) (S)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: BSET (*D*) (*S*)	**Step length**	**5**

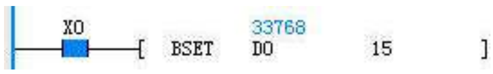
Operand	Type	Applicable soft element														Indexing
D	WORD			KnY	KnM	KnS	KnLM		D		C	T	V	Z	R	√
S	INT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T	V	Z	R	√

- Operand description

  D: Destination operand

  S2: Designated bit (0≤***S2***≤15, otherwise the system reports an operand error)
- Function description

  Setting the bit ***S*** of ***D*** element.
- Application instance



LD X0
BSET D0 15

Setting thebit 15 in D0 (1000: 2#0000001111101000) when the energy flow is valid. After execution, D0=33768 (2#1000001111101000).

6.15.9 BRST: Word bit coil reset instruction

LAD: ├─┤ ├─[BRST (D) (S)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: BRST (*D*) (*S*)	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
D	WORD			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: Designated bit (0≤***S2***≤15, otherwise the system reports an operand error)
- Function description

Clearing the bit ***S*** of ***D*** element.

- Application instance

Clearing thebit 8 in D0 (1000: 2#0000001111101000) when the energy flow is valid. After execution, D0=744 (2#0000001011101000).

```
LD    X0
BRST  D0  8
```

6.16 Comparison contact instructions

6.16.1 LD (=, <, >, <>, >=, <=): Integer comparison LD※instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
= (S1) (S2) < (S1) (S2) > (S1) (S2) <> (S1) (S2) >= (S1) (S2) <= (S1) (S2)	Influenced flag bit	
IL: LD= (S1) (S2) **LD< (S1) (S2)** **LD> (S1) (S2)** **LD<> (S1) (S2)** **LD>= (S1) (S2)** **LD<= (S1) (S2)**	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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

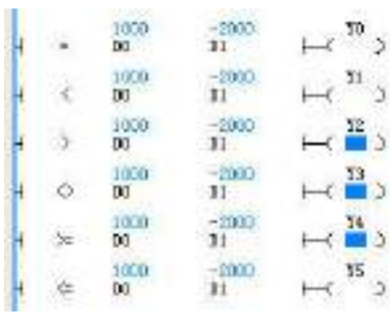
S1: Comparison parameter 1

S2: Comparison parameter 2

- Function description

Conducting the BIN comparison on the content of ***S1*** and ***S2***, and using the comparison result to drive the subsequent operation.

- Application instance




```
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
```

Conducting the BIN comparison on the data of D0 and D1, and using the comparisonresult to determine the output state of the subsequent elements.

6.16.2 AND (=, <, >, <>, >=, <=): Integer comparison AND ※instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
= (S1) (S2) < (S1) (S2) > (S1) (S2) <> (S1) (S2) >= (S1) (S2) <= (S1) (S2)	Influenced flag bit	

IL: AND= (S1) (S2) **AND< (S1) (S2)** **AND> (S1) (S2)** **AND<> (S1) (S2)** **AND>= (S1) (S2)** **AND<= (S1) (S2)**	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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

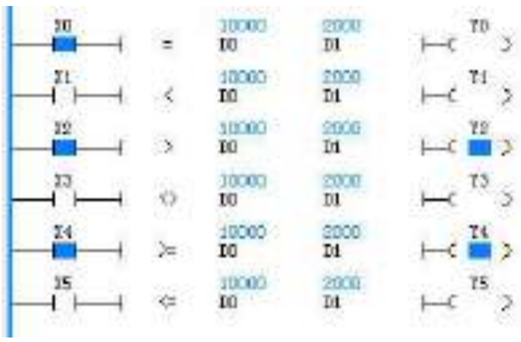
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the BIN comparison on the content of ***S1*** and ***S2***, and using the comparison result in serial connection with other nodes to drive the subsequent operation.

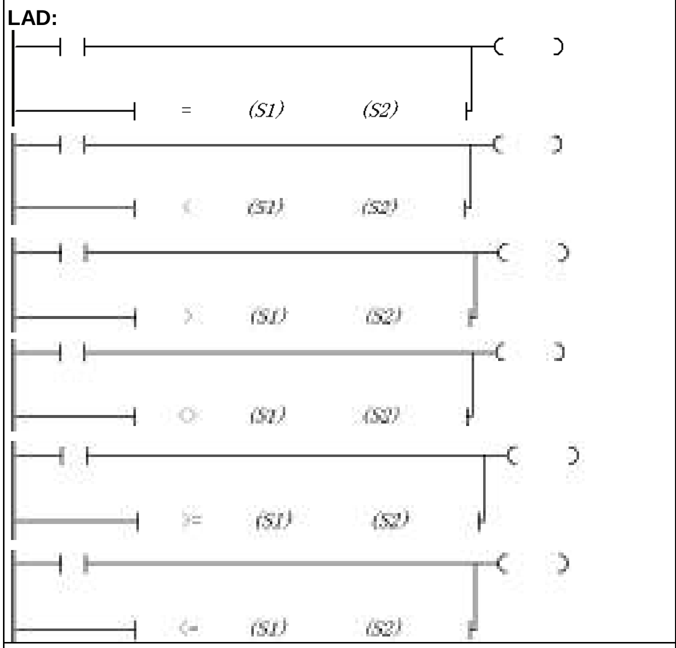
- Application instance



```
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
```

Conducting the BIN comparison on the data of D0 and D1, and using the comparison result in serial connection with other nodes to determine the output state of the subsequent elements.

6.16.3 OR (=, <, >, <>, >=, <=): Integer comparison OR※instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
	**Influenced flag bit**	
IL: OR＝ (S1) (S2) **OR＜ (S1) (S2)** **OR＞ (S1) (S2)** **OR＜＞ (S1) (S2)** **OR＞＝ (S1) (S2)** **OR＜＝ (S1) (S2)**	**Step length**	**5**

Operand	Type	Applicable soft element														Indexing
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

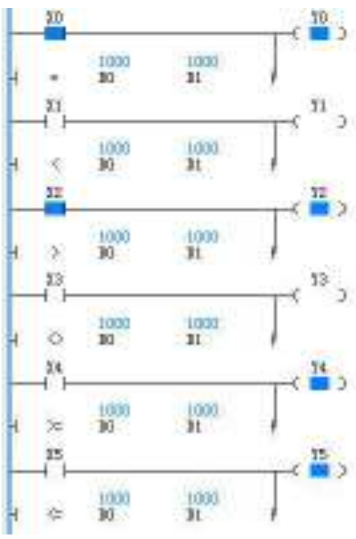
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result in parallel connection with other nodes to drive the subsequent operation.

- Application instance



```
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
```

Conducting the comparison on the data of D0 and D1, and using the comparisonresult in parallel connection with other nodes to determine the output state of the subsequent elements.

6.16.4 LDD (=, <, >, <>, >=, <=): Long integer comparison LDD※instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
D= (S1) (S2) D< (S1) (S2) D> (S1) (S2) D<> (S1) (S2) D>= (S1) (S2) D<= (S1) (S2)	Influenced flag bit	
IL: LDD= (S1) (S2) **LDD< (S1) (S2)** **LDD> (S1) (S2)** **LDD<> (S1) (S2)** **LDD>= (S1) (S2)** **LDD<= (S1) (S2)**	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
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

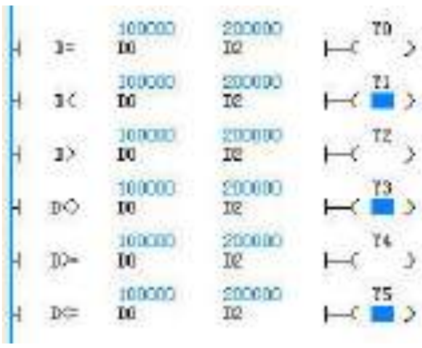
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result to drive the subsequent operation.

- Application instance



LD=D0 D2
OUTY0
LD<D0 D2
OUT Y1
LD<> D0 D2
OUT Y2
LD>= D0 D2
OUT Y3
LD>= D0 D2
OUT Y4
LD<=D0 D2
OUT Y5

Conducting the comparison on (D0,D1) and (D2,D3), and using the comparisonresult to determine the output state of the subsequent elements.

6.16.5 ANDD (=, <, >, <>, >=, <=): Long integer comparison ANDD※instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
D= (S1) (S2) D< (S1) (S2) D> (S1) (S2) D<> (S1) (S2) D>= (S1) (S2) D<= (S1) (S2)	Influenced flag bit	

IL: ANDD＝ (S1) (S2) **ANDD＜ (S1) (S2)** **ANDD＞ (S1) (S2)** **ANDD＜＞ (S1) (S2)** **ANDD＞＝ (S1) (S2)** **ANDD＜＝ (S1) (S2)**	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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

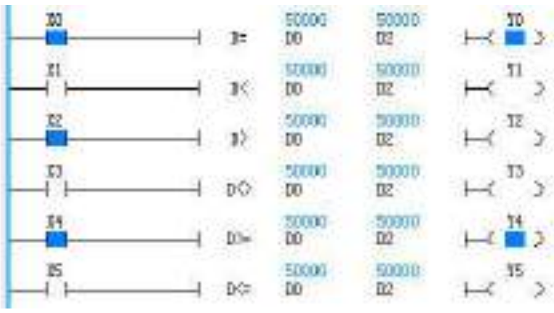
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result in serial connection with other nodes to drive the subsequent operation.

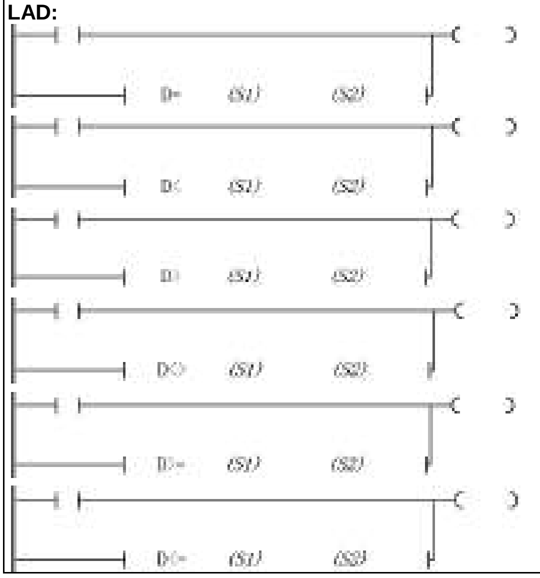
- Application instance



```
LD      X0
LDD=    D0 D2
OUT     Y0
LD      X1
LDD<    D0 D2
OUT     Y1
LD      X2
LDD<>   D0 D2
OUT     Y2
LD      X3
LDD<>   D0 D2
OUT     Y3
LD      X4
LDD>=   D0 D2
OUT     Y4
LD      X5
LDD<=   D0 D2
OUT     Y5
```

Conducting the comparison on (D0,D1) and(D2,D3), and using the comparisonresult in serial connection with other nodes to determine the output state of the subsequent elements.

6.16.6 ORD (=, <, >, <>, >=, <=): Long integer comparison ORD※instruction

LAD: 	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	

IL: ORD＝ (S1) (S2) **ORD＜ (S1) (S2)** **ORD＞ (S1) (S2)** **ORD＜＞ (S1) (S2)** **ORD＞＝ (S1) (S2)** **ORD＜＝ (S1) (S2)**	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
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

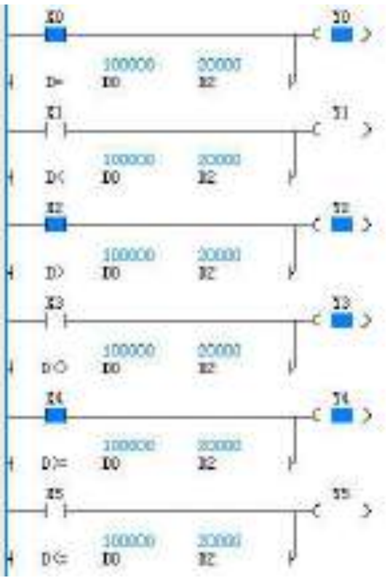
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result in parallel connection with other nodes to drive the subsequent operation.

- Application instance



```
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
```

Conducting the comparison on (D0,D1) and (D2,D3), and using the comparison result in parallel connection with other nodes to determine the output state of the subsequent elements.

6.16.7 LDR: Floating-point number comparison instruction

LAD: R= (S1) (S2) R< (S1) (S2) R> (S1) (S2) R<> (S1) (S2) R>= (S1) (S2) R<= (S1) (S2)	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	
IL: LDR＝ (S1) (S2) **LDR＜ (S1) (S2)** **LDR＞ (S1) (S2)** **LDR＜＞ (S1) (S2)** **LDR＞＝ (S1) (S2)** **LDR＜＝ (S1) (S2)**	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	REAL	Constant							D				V		R	√
S2	RAEL	Constant							D				V		R	√

- Operand description

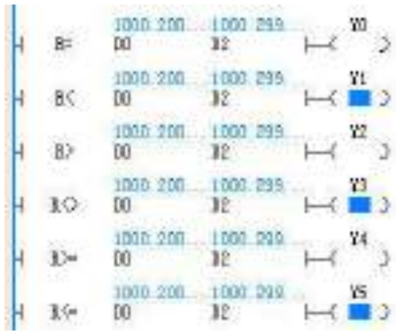
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result to drive the subsequent operation.

- Application instance



```
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
```

Conducting the comparison on (D0,D1) and (D2,D3), and using the comparison result to determine the output state of the subsequent elements.

6.16.8 ANDR: Floating-point number comparison instruction

LAD: R= (S1) (S2) R< (S1) (S2) R> (S1) (S2) R<> (S1) (S2) R>= (S1) (S2) R<= (S1) (S2)	**Applicable model**	**IVC2L IVC1 IVC3 IVC1L**
	Influenced flag bit	
IL: ANDR＝ (S1) (S2) **ANDR＜ (S1) (S2)** **ANDR＞ (S1) (S2)** **ANDR＜＞ (S1) (S2)** **ANDR＞＝ (S1) (S2)** **ANDR＜＝ (S1) (S2)**	**Step length**	**7**

Operand	Type	Applicable soft element														Indexing
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√

- Operand description

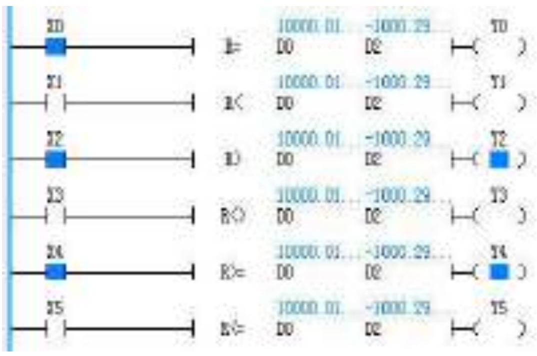
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result in serial connection with other nodes to drive the subsequent operation.

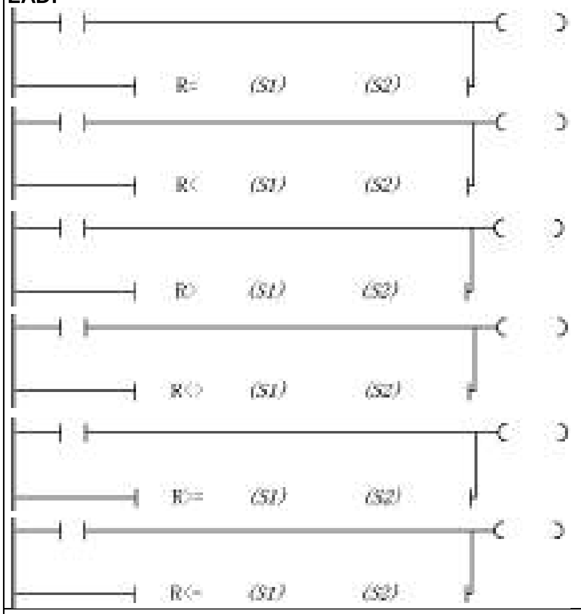
- Application instance



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

Conducting the comparison on (D0,D1) and(D2,D3), and using the comparison result in serial connection with other nodes to determine the output state of the subsequent elements.

6.16.9 ORR: Floating-point number comparison instruction

<table>
<tr><td rowspan="2">LAD:
</td><td>Applicable model</td><td>IVC2L IVC1 IVC3 IVC1L</td></tr>
<tr><td>Influenced flag bit</td><td></td></tr>
<tr><td>IL: ORR＝ (S1) (S2)
ORR＜ (S1) (S2)
ORR＞ (S1) (S2)
ORR＜＞ (S1) (S2)
ORR＞＝ (S1) (S2)
ORR＜＝ (S1) (S2)</td><td>Step length</td><td>7</td></tr>
</table>

Operand	Type	Applicable soft element														Indexing
S1	REAL	Constant							D				V		R	√
S2	REAL	Constant							D				V		R	√

- Operand description

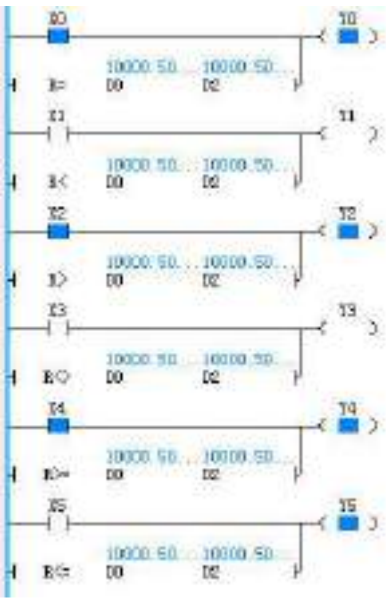
  S1: Comparison parameter 1

  S2: Comparison parameter 2

- Function description

  Conducting the comparison on the content of ***S1*** and ***S2***, and using the comparison result in parallel connection with other nodes to drive the subsequent operation.

- Application instance



```
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
```

Conducting the comparison on (D0,D1) and (D2,D3), and using the comparison result in parallel connection with other nodes to determine the output state of the subsequent elements.

6.16.10 CMP: Instruction for setting integer comparison to ON

LAD: CMP (S1) (S2) (D)]	**Applicable model**	**IVC2L IVC3**	
	Influenced flag bit		
IL: CMP *(S1) (S2) (D)*	**Step length**	**7**	

Operand	Type	Applicable soft element														Indexing
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: Number of data or soft element to be compared.

  S2: Number of data or soft element that is the source for comparison.

  D: Number of the start element that is used to output the result

- Function description

  Executing the instruction, and comparing ***S1*** and ***S2*** when the energy flow is valid. Making one of (***D***) (***D***+1) (***D***+2) ON according to its result (less, equal, or larger).

- Application instance

LD m0
CMP 1000 2000
M3

6.16.11 LCMP: Instruction for setting long integer comparison to ON

LAD: LCMP (S1) (S2) (D)]	**Applicable model**	**IVC2L IVC3**	
	Influenced flag bit		
IL: LCMP *(S1) (S2) (D)*	**Step length**	**9**	

Operand	Type	Applicable soft element														Indexing
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: Number of the start element that is used to output the result.

- Function description

  Executing the instruction, and comparing ***S1*** and ***S2*** when the energy flow is valid. Making one of (***D***) (***D***+1) (***D***+2) ON according to its result (less, equal, or larger).

- Application instance

LD m1
LCMP 200000 300000 M6

6.16.12 RCMP: Instruction for setting floating-point number comparison to ON

LAD: ⊢⊣ ⊢—[RCMP (S1) (S2) (D)]								**Applicable model**		**IVC2L IVC3**					
								Influenced flag bit							
IL: RCMP *(S1) (S2) (D)*								**Step length**		**9**					
Operand	Type	Applicable soft element													Indexing
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: Number of the start element that is used to output the result.

- Function description

  Executing the instruction, and comparing ***S1*** and ***S2*** when the energy flow is valid. Making one of (***D***) (***D***+1) (***D***+2) ON according to its result (less, equal, or larger).

- Application instance

  ```
  M2
  ⊢⊣ ⊢—[ RCMP 500.3400 200.4000 Y7 ]
  ```

  ```
  LD      m2
  RCMP 500.3400  200.4000
  Y7
  ```

6.17 Bulk data processing instructions

6.17.1 BKADD: Bulk data addition operation instruction

LAD: ⊢⊣ ⊢—[BKADD (S1) (S2) (D) (S3)]								**Applicable model**		**IVC3**					
								Influenced flag bit		**Zero flag, carry flag, and borrow flag**					
IL: BKADD *(S1) (S2) (D) (S3)*								**Step length**		**9**					
Operand	Type	Applicable soft element													Indexing
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: Number of the start soft element for savingthe data on which the addition operation is to be performed

  S2: Constant for performing the addition operation, or number of the start soft element that saves data on which the addition operation is to be performed

  D: Number of the start soft element for saving the operation result

  S3: Data quantity

- Function description

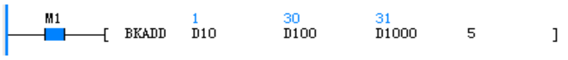
  1. Executing the instruction, increasing S3 units 16-bit data starting from S1 byS3 units 16-bit data (BIN) starting from S2, and storing the operation result in S3 units starting from D when the energy flow is valid.

  2. You can directly specify a 16-bit constant in S2. Sequentially performing, when S2 is a constant, the addition operation on S3 units16-bit data starting from S1 and S2, and storing the operation result in S3 units starting from D.

- Note

  When the operation result overflows, the carry flag bit is not setto ON.

- Application instance



LD M1

BKADD D10 D100 D1000 5

Sequentially increasing the contents of 5 units starting from D10 by that of 5 units starting from D100, and storing the operation results in 5 units starting from D1000 when M1=ON. In this case, D1000=D10+D100, D1001=D11+D101,……, D1004=D14+D104.

6.17.2 BKSUB: Bulk data subtraction operation instruction

LAD: 	**Applicable model**	**IVC3**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: BKSUB *(S1) (S2) (D) (S3)*	**Step length**	**9**	

Operand	Type	Applicable soft element													Indexing
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: Number of the start soft element for savingthe data on which the subtraction operation is to be performed

S2: Constant for performing the addition operation, or number of the start soft element that saves data on which the subtraction operation is to be performed

D: Number of the start soft element for saving the operation result

S3: Data quantity

- Function description

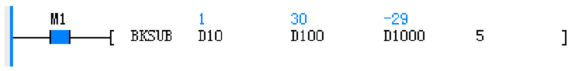
1. Executing the instruction, subtracting S3 units 16-bit data starting from S1 by S3 units 16-bit data (BIN) starting from S2, and storing the operation result in S3 units starting from D when the energy flow is valid.

2. You can directly specify a 16-bit constant in S2. Sequentially performing, when S2 is a constant, the subtraction operation on S3 units 16-bit data starting from S1 and S2, and storing the operation result in S3 units starting from D.

- Note

When the operation result overflows, the carry flag bit is not setto ON.

- Application instance



LD M1

BKSUB D10 D100 D1000 5

Sequentially subtracting the contents of 5 units starting from D10 by that of 5 units starting from D100, and storing the operation results in 5 units starting from D1000 when M1=ON. In this case, D1000=D10-D100, D1001=D11-D101, ……, D1004=D14-D104.

6.17.3 BKCMP=,>,<,<>,<=,>=: Bulk data comparison instruction

LAD: 	**Applicable model**	**IVC3**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: BKCMP=,>,<,<>,<=,>= *(S1) (S2) (D) (S3)*	**Step length**	**9**	

Operand	Type	Applicable soft element													Indexing
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: Number of the start soft element to be compared or saved

  S2: Number of the startsoft element that stores the comparison source

  D: Number of the start soft elementthat saves thecomparison result

  S3: Data quantity

- Function description

  1. Comparing the S3 units 16-bit data starting from S1 with S3 units 16-bit data (BIN) starting from S2, and storing the operation result in S3 units starting from D.

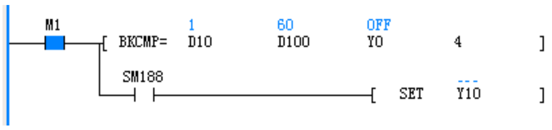
  2. You can directly specify a 16-bit constant in S1. Sequentially performing, when S1 is a constant, the comparison on S3 units16-bit data starting from S1 with that of S2, and storing the operation result in S3 units starting from D.

  3. Settingthe comparison set flag bit of the data block (SM188) when S3 units comparison results starting from D are ON.

- Note

  When the operation result overflows, the carry flag bit is not set ON.

- Application instance



LD M1

BKCMP= D10 D100 Y0 4

LD SM188

SET Y10

When M1=ON, the contents of the four units starting from D10 are compared with the contents of the four units starting from D100, and the result is stored in the four units starting from Y0. In addition, when the comparison result is all ON, Y10 turns ON.

Sequentially comparing the contents of 4 units starting from D10 by that of 4 units starting from D100, and storing the operation results in 4 units starting fromY0 when M1=ON. In this case, Y0 is set ON.

6.18 Datasheet instructions

6.18.1 LIMIT: Upper/lower limit control instruction

LAD: 	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: LIMIT *(S1) (S2) (S3) (D)*	**Step length**	**9**

Operand	Type	Applicable soft element													Indexing
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

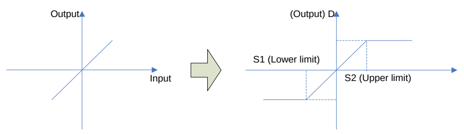
  S2: Upper limit value

  S3: Input value to be controlled by the upper and lower limits

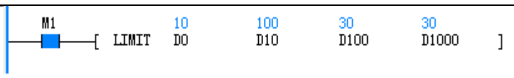
  D: Number of the start soft element that stores the output value that has been controlled by the upper and lower limits

- Function description

  You can judge whether the input value specified in S3 is within the range of the upper and lower limits specified by S1 and S2 so as to control and save the value in D. In this case, when S3<S1, D=S1; when S3>S2, D=S2; when S1 <=S3<=S2, D=S2.



- Application instance



LD M1

LIMITD0 D10 D100 D1000

Performing the limit control of D0﹣D10 on the content of D100, and storing the obtained result in D1000 when M1=ON.In this case, D0(10)<=D100(30)<=D10(100), and D1000=30.

6.18.2 DBAND: Deadband control instruction

LAD: 	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: DBAND *(S1) (S2) (S3) (D)*	**Step length**	**9**

Operand	Type	Applicable soft element													Indexing
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 of the deadband

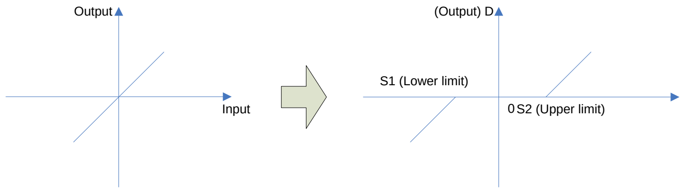
  S2: Upper limit value of the deadband

  S3:Input value to be controlled by the deadband

  D: Number of the start soft element that stores the output value that has been controlled by the deadband

- Function description

  You can judge whether the input value specified in S3 is within the range of the deadband specified by S1 and S2 so as to control and save the value in D. In this case, when S3 < S1, D = S3-S1; when S3 > S2, D = S3-S2; When S1<=S3<=S2, D=0.



- Application instance

LD M1

DBAND D0 D10 D100 D1000

Performing the deadband control of D0﹣D10 on the content of D100, and storing the obtained result in D1000 when M1=ON.In this case, D0 (-100)<D100(30)<D10(100), and D1000=0.

6.18.3 ZONE: Zone control instruction

LAD: 	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ZONE *(S1) (S2) (S3) (D)*	**Step length**	**9**

Operand	Type	Applicable soft element													Indexing
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 deviation value that is added to the input value

S2: Positive deviation value that is added to the input value

S3: Input value to controlled by the zone control

D: Number of the start soft element that stores the output value that hasbeen controlled byzone

- Function description

You can judge the input value specified in S3 plus the deviation value specified by S1 or S2 so as to control and save the values in D. In this case, when S3<0, D=S3+S1; when S3>0, D=S3+S2; when S3 =0, D=0.

- Application instance

LD M1

ZONE D0 D10 D100 D1000

Performing the zone control of D0－D10 on the content of D100, and storing the obtained result in D1000 when M1=ON.In this case, D100(30)>0,D1000=D100(30)+D10(100), and D1000=130.

6.18.4 SCL: Coordinatesetting instruction

LAD:

Applicable model	**IVC3**
Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: SCL *(S1) (S2) (S3) (D)*	
Step length	**7**

Operand	Type	Applicable soft element													Indexing
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: Input value on which coordinates are set, or number of the soft element that stores the inputted value.

S2: Number of the start table conversion soft element that is used to set the coordinates.

D: Number of the soft element that stores the output value on which coordinates have been set.

- Function description

1. Setting the coordinates on the input values specified by S1 according to the specified conversion characteristics, and storing the obtained results in the soft element numbers specified by D.

2. The conversion for setting the coordinates is performed based on the data table stored in soft elements, of which the start soft element is specified by S2. However, when the outputted data is not an integer value, the data is outputted after rounding off the first decimal place.

3. The coordinates are set through the conversion table.

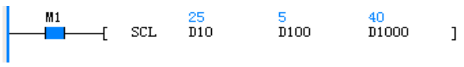
Points of the coordinates		S2
Point 1	X coordinate	S2+1
	Y coordinate	S2+2
Point 2	X coordinate	S2+3
	Y coordinate	S2+4
…	…	…
Point n (End)	X coordinate	S2+2n-1
	Y coordinate	S2+2n

- Note

1. The X data of the data table needs to be arranged in ascending order. If only part of the data is not arranged in ascending order and the detection is started from button, and the operation before this part is still executed;

2. S1 must be within the range set by the data table;

- Application instance



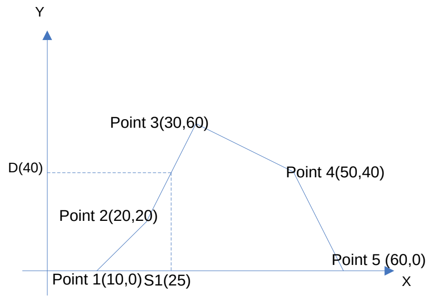
LD M1

SCL D10 D100 D1000

Setting the coordinates on the content of D10, and storing the obtained results in D1000 when M1=ON.

Points of the coordinates		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 search instruction

LAD: 	Applicable model	IVC3
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: SER *(S1)* *(S2)* *(S3)* *(D)*	Step length	9

Operand	Type	Applicable soft element													Indexing
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: Number of the start soft element to be searched.

S2: Reference value or number of the soft element that stores the target value for search.

D: Number of the start soft element that stores the number of the pieces of same data, the position of the maximum value, and the position of the minimum value.

S3: Number of units of data to be searched (1 ≤ S3 ≤ 256).

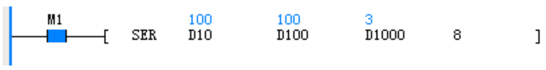
- Function description

1. Searching S3 data starting from S1, searching the same data as S2, and storing the obtained results in D-D+4.

2. When the same data exists, the number of the same data, and positions of the initial value/final value, min. and max. value are stored in 5 soft elements starting from D.

3. When the same data does not exist, the first 3 soft elements store 0, and the other two are the same as above.

- Application instance



LD M1

SER D0 D10 D100 D1000 8

Searchingthe contents of 8 unit starting from D10, and storing the search results in5 units starting from D1000 when M1=ON.

The element value to be searched S1	value	The element value to be compared S2	Data position	Search result D	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 Character string instructions

6.19.1 STRADD: String combination instruction

<table>
<tr><td rowspan="2">LAD:
</td><td>Applicable model</td><td>IVC3</td></tr>
<tr><td>Influenced flag bit</td><td>Zero flag, carry flag, and borrow flag</td></tr>
<tr><td>IL: STRADD (S1) (S2) (D)</td><td>Step length</td><td>7</td></tr>
</table>

Operand	Type	Applicable soft element													Indexing
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 connection

- Function description

  1. Connecting the string units starting from S1 and S2, and storing them in the soft elements starting from D when the energy flow is valid;

  2. The combination of strings indicates that the first character of the S2 stringis connected to the end character of the S1 string, and the end identifier of the S1string is ignored.

  3. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

  4. When the number of connected characters is an odd number, '00H' is added to the high byte of the last character soft element. When it is even, '0000H' is added to the next element of the last character soft element.

- Note

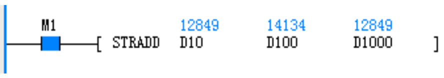
  1. When S1 and S2 specify a character string, a maximum of 32 characters are allowed. The comma and the double quotation mark represent the separators in the host computer software, so the character cannot be recognized by the host computer software;

  2. If '00H' is stored in both S1 and S2, then '0000H' can be added directly to D;

  3. When the soft element addresses of the string unit of S1 and D or S2 and D overlaps, the system reports the "The value of the Instruction operand is illegal" error.

  4. When '00H' does not appear in the corresponding soft element range of the string unit starting from S1 or S2, the system reports the "The element number of the instruction operand is out of range" error.

- Application instance



LD M1

STRADD D10 D100 D1000

Connecting the string unit starting from D10 and the string unit starting from D100, and storing the obtained result in the unit starting from D1000 when M1=ON.

	B15---b8	b7---b0
D10	0x32	0x31
D11	0x34	0x33
D12	0x36	0x35
D13	0x00	0x00

+

	B15---b8	b7---b0
D100	0x38	0x37
D101	0x61	0x39
D102	0x00	0x62

	B15---b8	b7---b0
D1000	0x32	0x31
D1001	0x34	0x33
D1002	0x36	0x35
D1003	0x38	0x37
D1004	0x61	0x39
D1005	0x00	0x62

6.19.2 STRLEN: Instruction for detecting the string length

LAD: 	Applicable model	IVC3
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: STRLEN (S) (D)	Step length	5

Operand	Type	Applicable soft element													Indexing
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: The string unit

  D:The length of the string unit

- Function description

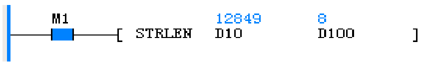
  1. Detecting the length of the S string, and storing the obtained value in D when the energy flow is valid.

  2. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

- Note

  When '00H' does not appear in the corresponding soft element range of the string unit starting from S, the system reports the "The element number of the instruction operand is out of range" error.

- Application instance



LD M1

STRLEN D10 D100

Detecting the length of the string unit starting from D10, and storing the obtained result in D100 when M1=ON.

6.19.3 STRRIGHT: Instruction for reading a string from right

LAD: 	Applicable model	IVC3
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: STRRIGHT (S1) (D) (S2)	Step length	7

Operand	Type	Applicable soft element													Indexing
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: The string unit

  D: Storing the extracted string unit

  S2: Number of the extracted characters

- Function description

  1. Extracting S2 characters starting from the last valid character of the S1 string (not counting '00H'), and storing them in the soft element starting from D when the energy flow is valid;

  2. Storing "00H" in D soft element when S2 is equal to zero;

  3. Adding '00H'to the high byte of the soft element that holds the last character when the number of extracted characters is an odd number. Adding '0000H' to the next element of the soft element that holds the last character when it is even.

  4. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

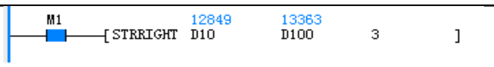
- Note

  1. When '00H' does not appear in the corresponding soft element range of the string unit starting from S1, the system reports the "The element number of the instruction operand is out of range" error.

  2. S2 is greater than or equal to 0.

  3. S2 needs to be less than or equal to the character number of the S1 string.

- Application instance



LD M1

STRRIGHT D10 D100 3

Extracting 3 characters from the right side of the string unit starting from D10, and storing them in the unit starting from D100 when M1=ON.

	B15---b8	b7---b0
D10	0x32	0x31
D11	0x34	0x33
D12	0x36	0x35
D13	0x00	0x00

	B15---b8	b7---b0
D100	0x35	0x34
D101	0x00	0x36

6.19.4 STRLEFT: Instruction for reading a string from left

LAD: 	Applicable model	IVC3
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: STRLEFT *(S1)* *(D)* *(S2)*	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
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: The string unit

D: Storing the extracted string unit

S2: Number of the extracted characters

- Function description

1. Extracting S2 characters rightward starting from the left side of the S1 string, and storing them in the soft element starting from D when the energy flow is valid;

2. Storing "00H" in D soft element when S2 is equal to zero;

3. Adding '00H'to the high byte of the soft element that holds the last character when the number of extracted characters is an odd number. Adding '0000H' to the next element of the soft element that holds the last character when it is even.

4. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

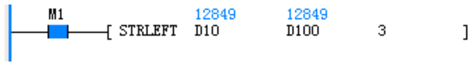
- Note

1. When '00H' does not appear in the corresponding soft element range of the string unit starting from S1, the system reports the "The element number of the instruction operand is out of range" error.

2. S2 is greater than or equal to 0.

3. S2 needs to be less than or equal to the character number of the S1 string.

- Application instance



LD M1

STRLEFT D10 D100 3

Extracting 3 characters from the left side of the string unit starting from D10, and storing them in the unit starting from D100 when M1=ON.

	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: Instruction for reading any characters of a string

LAD: 	Applicable model	IVC3
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: STRMIDR *(S1)* *(D)* *(S2)*	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
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

  D:The extracted string unit

  S2: Start position of the characters to be extracted

  S2+1: Number of characters to be extracted (n)

- Function description

  1. Extracting n character data of the S1 string starting from the S2 character, and storing them in the soft element starting from D when the energy flow is valid;

  2. Adding '00H'to the high byte of the soft element that holds the last character when the number of extracted characters is an odd number. Adding '0000H' to the next element of the soft element that holds the last character when it is even.

  3. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

  4. Performing no action when n is 0.

  5. Extractingall character data of the S1 string, and storing them in the soft element starting from D when n is -1.

- Note

  1. S2 needs to be less than or equal to the character number of the S1 string.

  2. n is greater than 2.

  3. S2 is greater than or equal to 1.

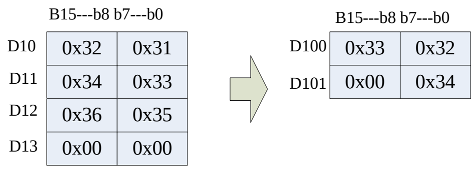
  4. When '00H' does not appear in the corresponding soft element range of the string unit starting from S1, the system reports the "The element number of the instruction operand is out of range" error.

- Application instance

LD M1

STRMIDR D10 D100 D0

Reading D1 (D1=3) characters from D0 (D0=2) of the string unit starting from D10, and storing them in the unit starting from D100 when M1=ON.



6.19.6 STRMIDW: Instruction for replacing any characters of a string

LAD: 	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: STRMIDW *(S1)* *(D)* *(S2)*	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
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 be used

  D: The string unit to be replaced

  S2: Start positionof the replacement

  S2+1: Number of characters to be replaced (n)

- Function description

  1. Replacing n character data starting from the S2 character of the D string with n characters data of S1 string when the energy flow is valid;

  2. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

  3. Performing no action when n is 0.

  4. When n is -1, the contents to the final character data specified by S1 is stored in the soft elements following the soft element specified by D.

- Note

  1. S2 needs to be less than or equal to the character number of the S1 string.

  2. n is greater than 2.

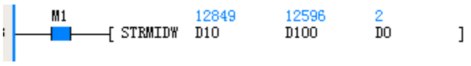
  3. S2 is greater than or equal to 1.

  4. When the number of characters for replacement exceeds the last character of the string unit starting from D, the extra characters are not saved.

  5. When '00H' does not appear in the corresponding soft element range of the string unit starting from S1 and D, the system reports

the "The element number of the instruction operand is out of range" error.

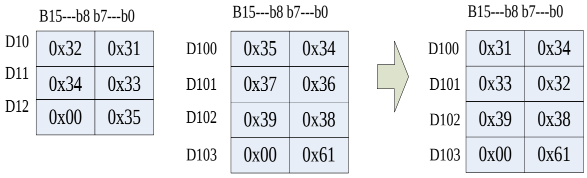
- Application instance



LD M1

STRMIDW D10 D100 D0

Replacing D1 (D1=3) characters after the D0 (D0=2) characters of the string unit starting from D100 with the first D1 (D1=3) characters of the string unit starting from D10 when M1=ON.



6.19.7 STRINSTR: String search instruction

<table>
<tr><td colspan="9">LAD:
</td><td colspan="2">Applicable model</td><td colspan="5">IVC3</td></tr>
<tr><td colspan="9"></td><td colspan="2">Influenced flag bit</td><td colspan="5">Zero flag, carry flag, and borrow flag</td></tr>
<tr><td colspan="9">IL: STRINSTR (S1) (S2) (D) (S3)</td><td colspan="2">Step length</td><td colspan="5">9</td></tr>
<tr><td>Operand</td><td>Type</td><td colspan="13">Applicable soft element</td><td>Indexing</td></tr>
<tr><td>S1</td><td>INT</td><td>String</td><td></td><td></td><td></td><td></td><td></td><td></td><td>D</td><td>SD</td><td>C</td><td>T</td><td>V</td><td>R</td><td>√</td></tr>
<tr><td>S2</td><td>INT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>D</td><td>SD</td><td>C</td><td>T</td><td>V</td><td>R</td><td>√</td></tr>
<tr><td>D</td><td>INT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>D</td><td>SD</td><td>C</td><td>T</td><td>V</td><td>R</td><td>√</td></tr>
<tr><td>S3</td><td>INT</td><td>Constant</td><td></td><td></td><td></td><td></td><td></td><td></td><td>D</td><td></td><td></td><td></td><td>V</td><td>R</td><td></td></tr>
</table>

- Operand description

S1: The string unit to be searched

S2: Search source

D: Search results

S3: Start positionof searching

- Function description

1. Searching the same string as the S1 string starting from the the S3 character of the S2 string, and storing the string position information of search results in D when the energy flow is valid.

2. Storing "0" in D when there is no consistent string.

3. No operation is performed when the position S3 where the search is started is a negative number or "0".

4. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

- Note

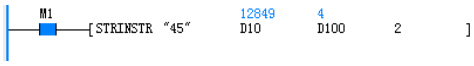
1. When '00H' does not appear in the corresponding soft element range of the string unit starting from S1 or S2, the system reports the "The element number of the instruction operand is out of range" error.

2. S3 is less than or equal to the number of characters in the S2 string.

3. When S1 specifies a character string, a maximum of 32 characters are allowed. The comma and the double quotation mark represent the separators in the host computer software, so the character cannot be recognized by the host computer software.

4. When S1 is an empty character string ('00H'), the position of the S2 character string '00H' is detected (if S2 is an even number of characters, it is the position of the first '00H').

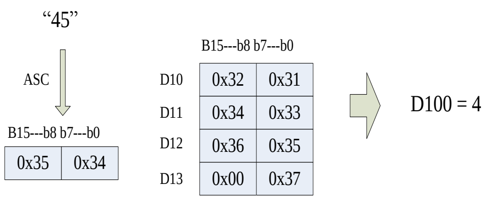
- Application instance



LD M1

STRINSTR "45" D10 D100 2

Searching the same string as the "45" starting fromthe second character of the character string unit starting from D10, and storing the search results in D100 when M1=ON.



6.19.8 STRMOV: String transmission instruction

LAD: STRMOV (S) (D)	**Applicable model**	**IVC3**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: STRMOV *(S) (D)*	**Step length**	**5**	

Operand	Type	Applicable soft element												Indexing
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. Transferring all data of the S string unit, including '00H', to the element unit starting from D.

  2. The valid data of a string unit indicates the data starting from the specified soft element for the string unit to the first data in which "00H" is detected.

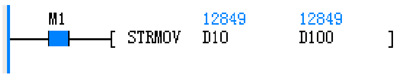
- Note

  1. When '00H' does not appear in the corresponding soft element range of the string unit starting from S, the system reports the "The element number of the instruction operand is out of range" error.

  2. When the number of characters of the S string unit is even, '00H' is stored in LSB, and both MSB and LSB of the corresponding position in D stores '00H'.

  3. When S1 specifies a character string, a maximum of 32 characters are allowed. The comma and the double quotation mark represent the separators in the host computer software, so the character cannot be recognized by the host computer software.

- Application instance



LD M1

STRMOV D10 D100

Transferring the character string data starting from D10 to the unit starting from D100 when M1=ON.

	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 Extension file register instructions

6.20.1 LOADR: Instruction for reading data from an extension file register

LAD: LOADR (S1) (S2)	**Applicable model**		
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**	
IL: LOADR *(S1) (S2)*	**Step length**	**5**	

Operand	Type	Applicable soft element													Indexing
S1	INT													R	√
S2	INT	Constant							D						

- Operand description

S1: Soft element unit of the extension register for storing data

S2: Number of data that has been read (1 ≤ S2 ≤ 1024)

- Function description

  Reading out S2 data from S1 of the extension file register stored in the storage box to the soft elements starting from S1 of the extension register in the PLC;

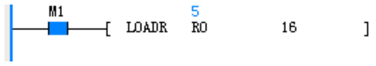
- Note

1. No operation is executed when S2 is specified as 0.

2. When the storage box is not connected, if you execute this instruction,the system reports that there is no memory card.

3. When S2=1024, the execution time of the instruction is about 80ms. You need to set the watchdog time correctly in the practical applications.

- Application instance



LD M1

LOADR R0 16

Reading 16 data starting from R0 of the memory card R backup area, and storing these data in the 16 soft element units starting from R0 when M1=ON.

6.20.2 SAVER: Instruction for writing data to an extension file register

LAD: 	**Applicable model**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: SAVER ***(S1) (S2) (D)***	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	INT													R	√
S2	INT	Constant													
D	INT								D						√

- Operand description

S1: Soft element unit of the extension register for storing data(only the start address of soft elements in the segment can be specified)

S2: Number of data that has been written per operation cycle (1 ≤ S2 ≤ 2048)

D: Storing number of data that has been written

- Function description

1. Writing data from S1﹣S1+2047 in the extension register to the same unit in the storage box by 2048/S2 (if 20148 is not divisible by S2, the quotient is increased by 1) operation cycle.

2. Storing number of data that has been written in D during the writing process.

3. Setting the instruction execution end flag SM189 when the execution of the instruction execution ends.

- Note

1. When S2=2048, the execution time of the instruction is about several hundredms. You need to set the watchdog time correctly.

2. You need to execute the INITER or INITR instruction to initialize the operated segments before driving the SAVER instruction. If the data written to the extension file register do not match those of the extension register, the system reports that the operation on the memory card is incorrect.

3. When a value of 0 is specified in S2, it is executed according to S2 = 2048.

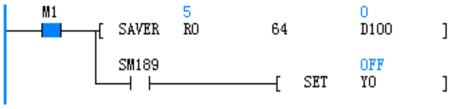
4. Data in S1 need to be number of the start soft elementin the extension register segment. Number of the start soft elementin each segment is as follows.

Segment number	Number of the start soft element	Range of writing to the memory card R backup area	Segment number	Number of the start soft element	Range of writing to the memory card R backup area
0	R0	R0–R2047	8	R16384	R16384–R18431
1	R2048	R2048–R4095	9	R18432	R18432–R20479
2	R4096	R4096–R6143	10	R20480	R20480–R22527
3	R6144	R6144–R8191	11	R22528	R22528–R24575
4	R8192	R8192–R10239	12	R24576	R24576–R26623
5	R10240	R10240–R12287	13	R26624	R26624–R28671
6	R12288	R12288–R14335	14	R28672	R28672–R30719
7	R14336	R14336–R16383	15	R30720	R30720–R32767

5. When no memory is connected, the system reports that there is no memory card.

6. When hardware writing protection is performed for the memory card, the system reports a memory card operation error.

- Application instance



LD M1

SAVER R0 64 D100

LD SM189

SET Y0

Storing 2048 soft element data starting from R0 in 2048 units starting from ER0, and storing the number of units that are already stored in D100 when M1=ON. When the execution of the instruction is completed, SM189=ON, and Y0=ON.

6.20.3 INITR: Instruction for initializing an extension register

LAD: 									**Applicable model**					
									Influenced flag bit	**Zero flag, carry flag, and borrow flag**				
IL: INITR *(S1) (S2)*									**Step length**	**5**				
Operand	Type	Applicable soft element												Indexing
S1	INT												R	√
S2	INT	Constant												

- Operand description

S1: Units of the extension register and extension file register to be initialized (only the address of the start soft element in the segment can be specified)

S2: Segments of the extension register and extension file register to be initialized (S2=1)

- Function description

1. Initializing S2 extension registers and extension file registers starting from S1, and writing the initial value 0xFFFF.

2. The initialization is executed by segments.

- Note

1. S1 needs to specify number of the start soft element in the segment. Number of the start soft element in each segment is as follows.

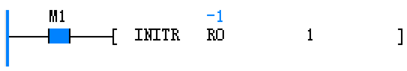
Segment number	Number of the start soft element	Range of writing to the memory card R backup area	Segment number	Number of the start soft element	Range of writing to the memory card R backup area
0	R0	R0–R2047	8	R16384	R16384–R18431
1	R2048	R2048–R4095	9	R18432	R18432–R20479
2	R4096	R4096–R6143	10	R20480	R20480–R22527
3	R6144	R6144–R8191	11	R22528	R22528–R24575
4	R8192	R8192–R10239	12	R24576	R24576–R26623
5	R10240	R10240–R12287	13	R26624	R26624–R28671
6	R12288	R12288–R14335	14	R28672	R28672–R30719
7	R14336	R14336–R16383	15	R30720	R30720–R32767

2. When the memory card is not used, the initialization of the extension file register is not performed.

3. If the memory card is connected,and hardware writing protection is performed for the memory card, the system reports a memory card operation error.

4. The instruction can only initialize one segment for each execution. When the memory card is used, the operation time of initializing each segment is about 100ms. In actual application, You need to set the watchdog time correctly.

- Application instance



LD M1

INITR R0 1

Initializing the extension registers R－R2047 in segment 0 when M1=ON. In the case of using a memory card, the extension file registers ER－ER2047 are also initialized.

6.20.4 LOGR: Instruction for logging on an extension register

LAD: 	**Applicable model**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: LOGR *(S1) (S2) (S3) (S4) (D)*	**Step length**	**11**

Operand	Type	Applicable soft element													Indexing
S1	INT								D		C	T			√
S2	INT	Constant							D						
S3	INT													R	
S4	INT	Constant													
D	INT								D						√

- Operand description

  S1: Object unit that performs the login

  S2: Number oflogined object units (1－1024)

  S3: Address of the start soft element unit used in login

  S4: Number of soft element segments used in login (1－16)

  D: Number of logined data

- Function description

  1. Continuously logining in S2 soft elements starting from S1 until S4 segments of the extension register and the extension file register starting starting from S3 are filled in when the energy flow is valid.

  2. Performing login in each execution cycle.

  3. Storing the number of logined data in D.

- Note

  1. When using the memory card, you need to initialize the segments to be logged. If the data logged on the extension file register is inconsistent with the data starting from S1, the system reports that the operation on the memory card is incorrect. In this case, you can initialize the segments to be logged one by one through the INITR or INITER command, or initialize all ER elements through the menu clear instruction of the background memory card (you need to select the user program, global variables, data block, and system block simultaneously). To avoid data loss, you need to backup the memory card file content through the background software before initialization.

2. S3 needs to specify number of the start soft element in the segment. Number of the start soft element in each segment is as follows.

Segment number	Number of the start soft element	Range of writing to the memory card R backup area	Segment number	Number of the start soft element	Range of writing to the memory card R backup area
0	R0	R0–R2047	8	R16384	R16384–R18431
1	R2048	R2048–R4095	9	R18432	R18432–R20479
2	R4096	R4096–R6143	10	R20480	R20480–R22527
3	R6144	R6144–R8191	11	R22528	R22528–R24575
4	R8192	R8192–R10239	12	R24576	R24576–R26623
5	R10240	R10240–R12287	13	R26624	R26624–R28671
6	R12288	R12288–R14335	14	R28672	R28672–R30719
7	R14336	R14336–R16383	15	R30720	R30720–R32767

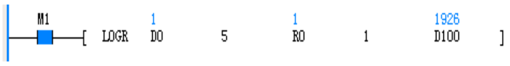
3. If there is memory hardware protection, the system reports that the operation on the memory card is incorrect.

4. Login data format

S3–S3+S2-1	The storage address of the first logined data S3 – S3+S2-1	D=S2	Data write area 1926×S4	Login data storage area
S3+S2–S3+2S2-1	The storage address of the second logined data S3 – S3+S2-1	D=2S2		
S3+2S2–S3+3S2-1		D=3S2		

…………	…………	……		
S3+1926×S4-1 –S3+2048×S4-1	Writing intothe location management area When using the data write area of one word, the sequence is changed from ON to OFF from 0 bits of S3+1926×S4-1, and the next soft element S3+1926×S4 is used when S3+1926×S4-1 is all OFF.		Management area 122 × S4	

- Application instance



LD M1

LOGR D0 5 R0 1 D100

Logging the data of the D0–D4 unit in the R0–R2047 unit of the segment 0, and recording the number of logined data in D100 when M1=ON.

6.20.5 INITER: Instruction for initializing an extension file register

LAD: 	**Applicable model**	
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: INITER *(S1) (S2)*	**Step length**	**5**

Operand	Type	Applicable soft element												Indexing
S1	INT												R	√
S2	INT	Constant												

- Operand description

S1: The same extension register unit that share the same address with the extension file register unit to be initialized(only the start address of extension register in the segment can be specified)

S2: Segments of the extension register and extension file register to be initialized (S2=1)

- Function description

1. Initializing S2 segments of extension file registers starting from S1 in the storage box, and writing the value 0xFFFF into it.

2. The initialization is executed by segments.

- Note

1. S1 needs to specify number of the start soft element in the segment. Number of the start soft element in each segment is as follows.

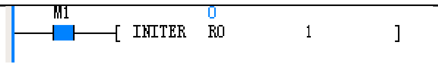
Segment number	Number of the start soft element	Range of writing to the memory card R backup area	Segment number	Number of the start soft element	Range of writing to the memory card R backup area
0	R0	R0–R2047	8	R16384	R16384–R18431
1	R2048	R2048–R4095	9	R18432	R18432–R20479
2	R4096	R4096–R6143	10	R20480	R20480–R22527
3	R6144	R6144–R8191	11	R22528	R22528–R24575
4	R8192	R8192–R10239	12	R24576	R24576–R26623
5	R10240	R10240–R12287	13	R26624	R26624–R28671
6	R12288	R12288–R14335	14	R28672	R28672–R30719
7	R14336	R14336–R16383	15	R30720	R30720–R32767

2. If there is memory hardware protection, the system reports that the operation on the memory card is incorrect.

3. When no storage box is connected, the system reports that there is no memory card.

4. The instruction can only initialize one segment for each execution, and the operation time of initializing each segment is about 100 ms. In actual application, You need to set the watchdog time correctly.

- Application instance



Initializing the extension file registers ER0–ER2047 in segment 0 when M1=ON.

LD M1

INITER R0 1

6.21 Positioning instructions

6.21.1 ZRN: Zero return instruction

LAD: ZRN (S1) (S2) (S3) (D)										**Applicable model**		**IVC2L IVC1 IVC1S IVC3 IVC1L**				
										Influenced flag bit						
IL: ZRN (*S1*) (*S2*) (*S3*) (*D*)										**Step length**		**11**				
Operand	Type	Applicable soft element														Indexing
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: Zero return speed, specifying the speed at which the zero return starts.

  32-bit instruction: IVC1 and IVC2L: 10–100000 (Hz), IVC1L: Y0, Y1 10–100000 (Hz), Y2, Y3 10–10000 (Hz), and IVC3: 10–200000 (Hz).

  S2: Crawling speed, the lower speed after the specified near-point dog signal (DOG) is turned ON.

  S3: DOG to be inputted in X element.

  When an element other than the input relay (X) is specified, the position offset of the zero point increases due to the influence of the PLC calculation cycle.

  D: Start address of the high-speed pulse output. For IVC1 and IVC2L, only Y0 or Y1 can be designated. For IVC1L, Y0, Y1, and Y2 can be designated. For IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

- Function description

  IVC1: When SM85 clearing signal is valid, the output points of the clearing signal that correspond to the high-speed pulse output points (Y0 and Y1) are Y2 and Y3 respectively. When SM85 is set, clearing signal is sent to the servo amplifier through Y2 and Y3.

  IVC1L and IVC3: Taking Y0 as an example. When SM280 clearing signal is valid, and elements specified by the SM281 clearing signal are invalid, Y10 is the output port of the clearing signal. When elements specified by the SM281 clearing signal are invalid, Y (N) specified by SD206 is the output port of the clearing signal.

- Note

  1. Because the ZRN instruction is incapable of searching for DOG automatically, it is required to start the zero return operation further than the front end of DOG detection device.

  2. During the zero return process, the value of the current value register decreases.

  3. The min. output pulse frequency that can be outputted actually, is determined by the formula below:

  $$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

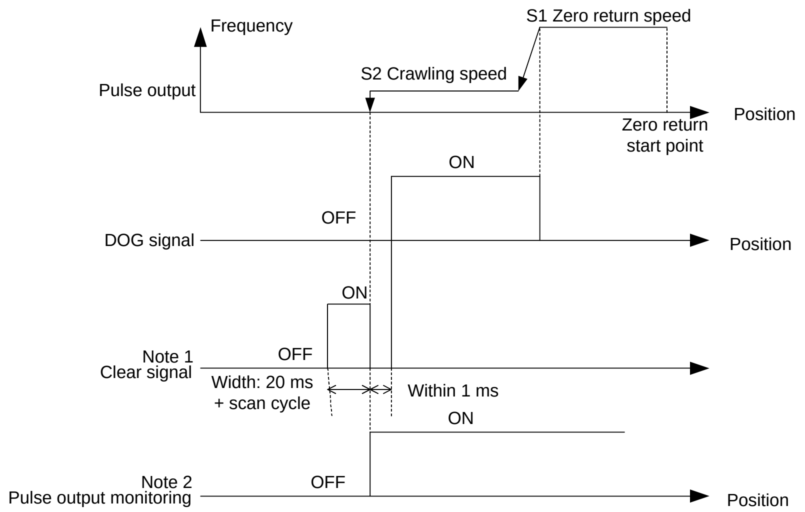
  In the above formula, $F_{\max}$ indicates the max. speed while T indicates the acceleration/deceleration time withthe unit of ms. The calculation result $F_{\min\_acc}$ is the limit value of min. output frequency.

  4. For the pulse output frequency, the calculated frequency is output even if a value less than the calculated one is assigned. The frequency of the starting section of ACC and end part of DEC cannot be less than the above calculation result. If the max. speed is lower than the above calculation result, there is not pulse output.

  5. Crawling speed needs to be larger than zero and less than one tenth of the max. speed.

  6. For details, refer to Chapter 11 "User Guide for Positioning Function".

- Time sequence diagram



Note 1: Enabling the clear function when SM85 is set
Note 2: SM82 and SM83 are the monitors of Y0 and Y1 pulses outputs respectively.

6.21.2 PLSV:Variable speed pulse output instruction

LAD: ─┤ ├──[PLSV (S) (D1) (D2)]	**Applicable model**	**IVC2L IVC1 IVC1S IVC3 IVC1L**
	Influenced flag bit	
IL: PLSV (*S*) (*D1*) (*D2*)	**Step length**	**8**

Operand	Type	Applicable soft element														Indexing
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: Output pulse frequency (Hz)

  32-bit instruction:

  IVC1 and IVC2L: 10–100000 (Hz), -10– -100000 (Hz),

  IVC1L: Y0, Y1 10–100000 (Hz), -10– -100000 (Hz), Y2, Y3 10–10000 (Hz), -10– -10000 (Hz)

  IVC3: 10–200000 (Hz), -10–-200000 (Hz).

  D1: Start address of the high-speed pulse output. For IVC1 and IVC2L, only Y0 or Y1 can be designated. For IVC1L, Y0, Y1, and Y2 can be designated. For IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

  D2: Start address of the rotating direction signal output. Corresponding to the positive/negative condition of ***S***, it acts as follows:

  - When S is positive: D2 is ON.
  - When S is negative: D2 is OFF.

- Function description

  1. You can change the output pulse frequency (***S***) freely even in the high-speed pulse output state.

  2. Because there is no acceleration or deceleration during the start/stop, if buffer is needed during the start or stop, it is recommended to use the RAMP instruction to change the value of the output pulse frequency (***S***).

  3. During the process of the high-speed pulse output, when the energy flow driven by the instruction turns OFF, the output stops without deceleration.

  4. If the high-speed pulse output monitoring (SM82 or SM83) is ON, the energy flow driven by the instruction is not driven by the instruction again after the energy flow turns OFF.

  5. The direction is determined by the positive or negative nature of ***S***.

- Note

  1. PLSY, PLS, and positioning instructions can output the high-speed pulses through Y0 and Y1. It is not allowed to use these instructions for high-speed pulse output on the same port at the same time.

  2. For details, refer to Chapter 11 "User Guide for Positioning Function".

6.21.3 DRVI: Relative position control instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
├──┤ ├──[DRVI (S1) (S2) (D1) (D2)]	Influenced flag bit	
IL: DRVI **(*S1*)** **(*S2*)** **(*D1*)** **(*D2*)**	Step length	11

Operand	Type	Applicable soft element														Indexing
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 (relatively specified)

32-bit instruction: IVC1 and IVC2L: －999999－＋999999, IVC1 and IVC3: -99999999－+99999999

S2: Output pulse frequency (Hz)

32-bit instruction: IVC1 and IVC2L: 10－100000 (Hz), IVC1L: Y0, Y1 10－100000 (Hz), Y2, Y3 10－10000 (Hz), IVC3: 10－200000 (Hz).

D1: Start address of the high-speed pulse output. For IVC1 and IVC2L, only Y0 or Y1 can be designated. For IVC1L, Y0, Y1, and Y2 can be designated. For IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

D2: Start address of the rotating direction signal output. Corresponding to the positive/negative condition of ***S***, it actsas follows:

  - When S1 is positive: D2 is ON.
  - When S1 is negative: D2 is OFF.

- Function description

1. ***S1*** corresponds to the following current value registers as a relative position.

  - When outputting to Y0: SD80, SD81 (32-bit)
  - When outputting to Y1: SD82, SD83 (32-bit)

2. During the reverse rotation, the value of current value register decreases.

3. The rotating direction is determined by the positive or negative nature of ***S1***.

4. During the execution of the instruction, even if the contents of the operands are changed, these changes cannot be displayed in the current operation and can only take effect at the next execution of instruction.

5. During the execution of the instruction, the output decelerates to stop when the energy flow driven by the instruction turns OFF. The execution completion flag SM does not act at the moment.

6. If the high-speed pulse output flag (SM80 or SM81) is ON, the energy flow driven by the instruction is not driven by the instruction again after the energy flow turns OFF.

- Note

1. The min. output pulse frequency that can be outputted actually, is determined by the formula below:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ indicates the max. speed while T indicates the acceleration/deceleration time withthe unit of ms. The calculation result $F_{\min\_acc}$ is the limit value of min. output frequency.

2. For the pulse output frequency, the calculated frequency is output even if a value less than the calculated one is assigned. The frequency of the starting section of ACC and end part of DEC cannot be less than the above calculation result. If the max. speed is lower than the above calculation result, there is not pulse output.

3. Crawling speed needs to be larger than zero and less than one tenth of the max. speed.

4. For details, refer to Chapter 11 "User Guide for Positioning Function".

6.21.4 DRVA: Absolute position control instruction

LAD:	Applicable model	IVC2L IVC1 IVC1S IVC3 IVC1L
├──┤ ├──[DRVA (S1) (S2) (D1) (D2)]	Influenced flag bit	
IL: DRVA **(*S1*)** **(*S2*)** **(*D1*)** **(*D2*)**	Step length	11

Operand	Type	Applicable soft element	Indexing

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: Target position (absolutelyspecified)

32-bit instruction: IVC1 and IVC2L: －999999﹣＋999999, IVC1 and IVC3: -99999999﹣+99999999

S2: Output pulse frequency (Hz)

32-bit instruction: IVC1 and IVC2L: 10﹣100000 (Hz), IVC1L: Y0, Y1 10﹣100000 (Hz), Y2, Y3 10﹣10000 (Hz), IVC3: 10﹣200000 (Hz).

D1: Start address of the high-speed pulse output. For IVC1 and IVC2L, only Y0 or Y1 can be designated. For IVC1L, Y0, Y1, and Y2 can be designated. For IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

The PLC output need to adopt transistor output mode.

D2: Start address of the rotating direction signal output.

If the pulse is transmitted in forward direction, ***D2*** is ON, otherwise ***D2*** is OFF.

- Function description

1. ***S1*** corresponds to the following current value registers as a relative position.

SD80	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC1)
SD81	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC1)
SD82	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC1)
SD83	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC1)
SD200	Current position value in the positioning instruction outputted through Y0 (MSB) (IVC2L, IVC1L, and IVC3)
SD201	Current position value in the positioning instruction outputted through Y0 (LSB) (IVC2L, IVC1L, and IVC3)
SD210	Current position value in the positioning instruction outputted through Y1 (MSB) (IVC2L ,IVC1L, and IVC3)
SD211	Current position value in the positioning instruction outputted through Y1 (LSB) (IVC2L ,IVC1L, and IVC3)
SD320	Current position value in the positioning instruction outputted through Y2 (MSB) (IVC1L and IVC3)
SD321	Current position value in the positioning instruction outputted through Y2 (LSB) (IVC1L and IVC3)
SD330	Current position value in the positioning instruction outputted through Y3 (MSB) (IVC3)
SD331	Current position value in the positioning instruction outputted through Y3 (LSB) (IVC3)
SD340	Current position value in the positioning instruction outputted through Y4 (MSB) (IVC3)
SD341	Current position value in the positioning instruction outputted through Y4 (LSB) (IVC3)
SD350	Current position value in the positioning instruction outputted through Y5 (MSB) (IVC3)
SD351	Current position value in the positioning instruction outputted through Y5 (LSB) (IVC3)
SD360	Current position value in the positioning instruction outputted through Y6 (MSB) (IVC3)
SD361	Current position value in the positioning instruction outputted through Y6 (LSB) (IVC3)
SD370	Current position value in the positioning instruction outputted through Y7 (MSB) (IVC3)
SD371	Current position value in the positioning instruction outputted through Y7 (LSB) (IVC3)

2. During the reverse rotation, the value of current value register decreases.

3. The rotating direction is determined by the positive or negative nature of ***S1***.

4. During the execution of the instruction, even if the contents of the operands are changed, these changes cannot be displayed in the current operation and can only take effect at the next execution of instruction.

5. During the execution of the instruction, the output decelerates to stop when the energy flow driven by the instruction turns OFF. The execution completion flag SM does not actat the moment.

6. If the high-speed pulse output flag (SM80 or SM81) is ON, the energy flow driven by the instruction is not driven by the instruction again after the energy flow turns OFF.

- Note

1. The min. output pulse frequency that can be outputted actually, is determined by the formula below:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ indicates the max. speed while T indicates the acceleration/deceleration time with the unit of ms. The calculation result $F_{\min\_acc}$ is the limit value of min. output frequency.

2. For the pulse output frequency, the calculated frequency is output even if a value less than the calculated one is assigned. The frequency of the starting section of ACC and end part of DEC cannot be less than the above calculation result. If the max. speed is lower than the above calculation result, there is not pulse output.

3. Crawling speed needs to be larger than zero and less than one tenth of the max. speed.

4. For details, refer to Chapter 11 "User Guide for Positioning Function".

6.21.5 ABS: Current value reading instruction

LAD: ├──┤ ├──[ABS (S) (D1) (D2)]	Applicable model	IVC2L IVC1
	Influenced flag bit	
IL: ABS (S) (D1) (D2)	Step length	8

Operand	Type	Applicable soft element														Indexing
S	BOOL	X	Y	M	S											
D1	BOOL		Y	M	S											
D2	DINT		KnY	KnM	KnS					D	SD	C			R	√

- Operand description

S: The input point from the servo device

The input points occupy three consecutive input Xs(S, S＋1, and S＋2) or other bit elements.

D1: The output points transmitted to the servo device

The output points occupy three consecutive output Ys (D1, D1＋1, and D1＋2) or other bit elements.

D2: Current value data (32-bit) read from the servo device

The current value data occupy two word elements: ***D2*** (MSB) and ***D2***+1 (LSB). Because the ABS value must be written into SD80 or SD82 (32-bit signed integer), this operand can be designated as SD80 or SD82 register directly.

- Function description

1. The PLCs and servo amplifier need to be powered on simultaneously or the latter needs to be powered on firstly, thus ensuring the PLC enters the running state after the servo amplifier is powered on.

2. Current value data read from the servo device can be set as the designated range of other word elements, but it is necessary to transmit this value to SD80 or SD82.

3. The energy flow of the ABS instruction, which should be in switch-on state after the ABS value is read out. When the ABS read operation is done, if the energy flow of the instruction is disconnected, the servo ON signal turns OFF and the operation cannot be executed.

4. SM82, SM83 corresponds to the output flag of Y0 and Y1, the flag is cleared when output is done.

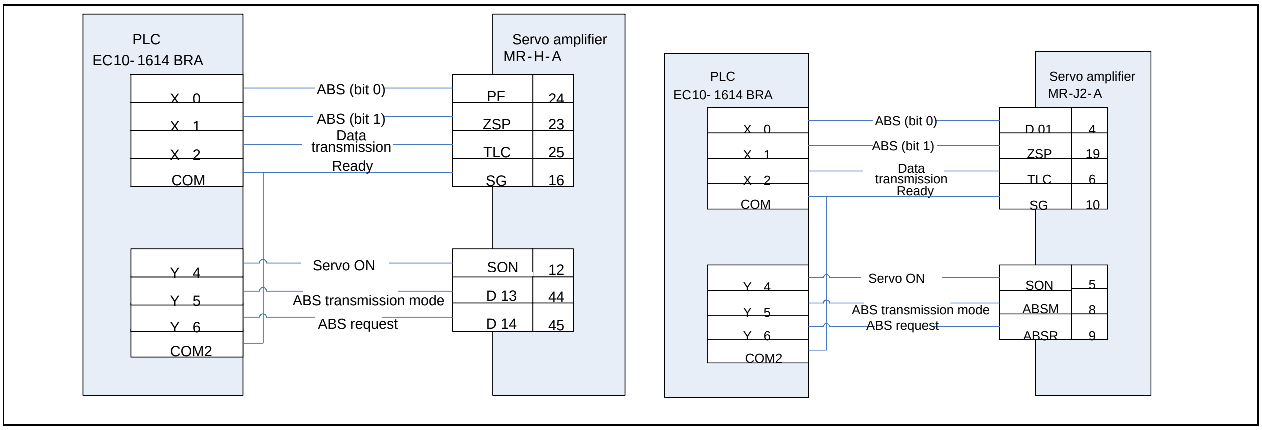
5. When the ABS instruction energy flow is switched on and the servo is on, the ABS instruction sends the transmission mode signal.

6. Carrying out the 32+6-bit data communication at the same time when the data transmission ready signal and the ABS request signal coincide with each other.

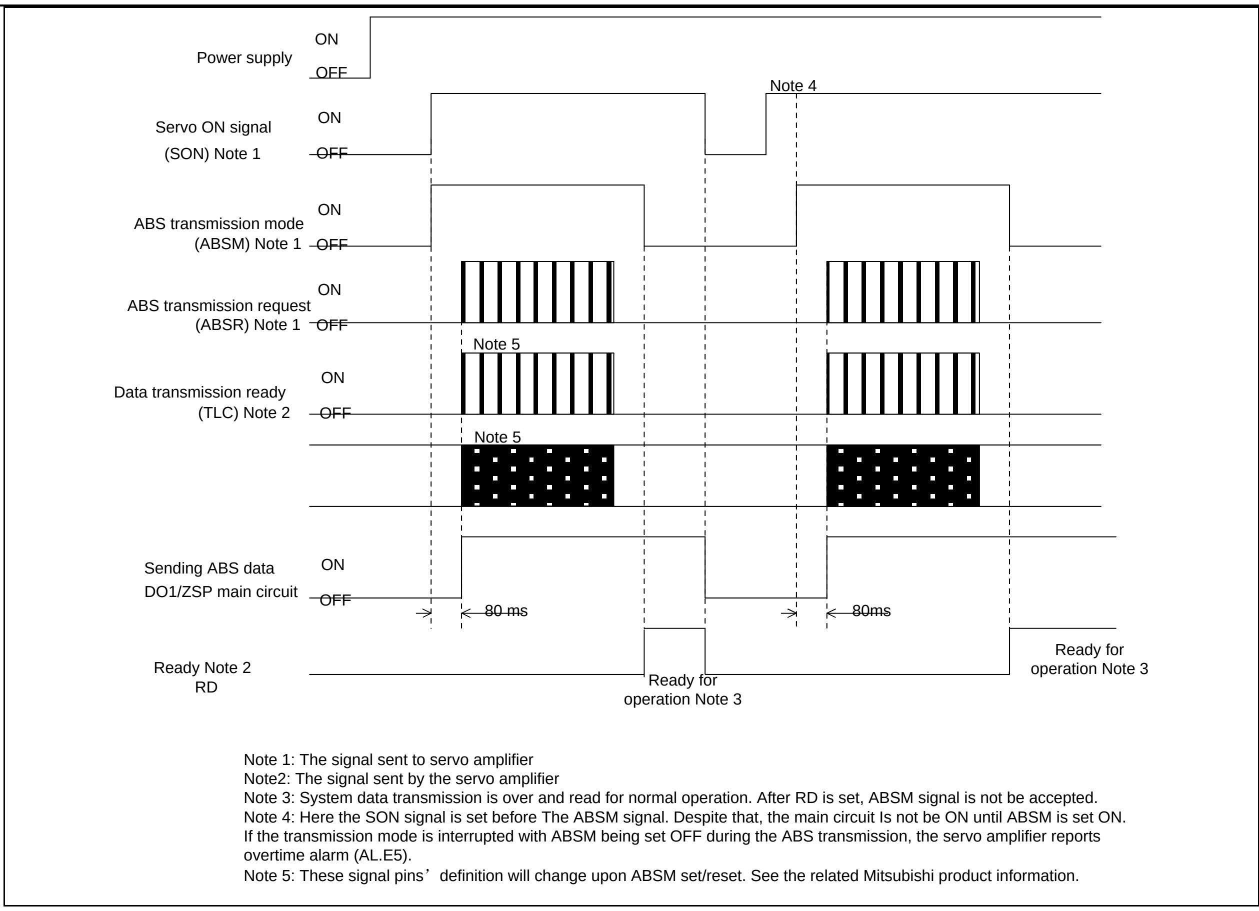
7. Data are transmitted through the ABS 2-bit (bit0 and bit1) circuit.

8. In case of data transmission timeout, the system reports No. 79 error while for checksum error, the system reports No. 80 error.

- The wiring connection for the I/O signals of the ABS instruction is as shown in the following figure.



- Sequence chart



- Note

  The ABS instruction supports the Mitsubishi MR–J2 and MR–J2S servo amplifiers, and uses its dedicated data transmission protocol to read the current value data of absolute position. The ABS instruction is 32-bit dedicated instruction. For the servo amplifiers of other brands, it is necessary to adopt communication or other designated modes to obtain the current value data of absolute position. When the ABS instruction is executed, the related I/O points are processed accordingly. Thus, the ABS instruction is applicable only to Mitsubishi servo amplifiers.

6.21.6 DSZR: Instruction for zero return with DOG

LAD: ├─┤ ├─[DSZR *(S1)* *(S2)* *(D1)* *(D2)*]	**Applicable model**	**IVC2L IVC3 IVC1L**
	Influenced flag bit	
IL: DSZR *(S1)* *(S2)* *(D1)* *(D2)*	**Step length**	**9**

Operand	Type	Applicable soft element													Indexing
S1	BOOL		X	Y	M	S									
S2	BOOL		X												
D1	BOOL			Y											
D2	BOOL			Y	M	S									

- Operand description

  S1: Specifying the soft element number of DOG. When an input soft element is designated, the position offset of the zero point increases due to the influence of the PLC calculation cycle.

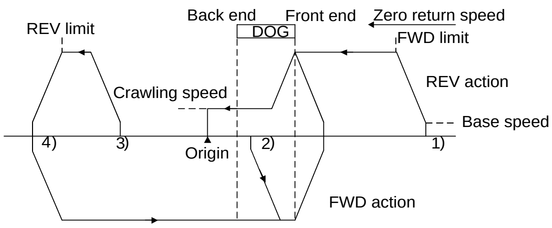
  S2: Specifying the soft element number of the input zero signal. Range: X0 – X7.

  D1: Specifying the pulse number of the output pulses. For IVC1 and IVC2L, only Y0 or Y1 can be designated. For IVC1L, Y0, Y1, and Y2 can be designated. For IVC3, Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 can be designated.

D2: Specifying the output object number of the rotating direction signal.

- Function description

The ZRN instruction that permits the use of DOG signal and zero signal. There is FWD and REV limits. The ZRN action depends on the ZRN position. A clearing signal is sent after the instruction ends.



1. When the starting position is before passing DOG:

1)Starting the ZRN action through executing the ZRN instruction.

2) Moving toward the ZRN direction atzero return speed.

3) Starting to decelerate to the crawling speed once the front end of DOG is detected.

4) Stopping once the first zero signal is detected again after detecting the back end of DOG.

2. When the starting position is within DOG:

1)Starting the ZRN action through executing the ZRN instruction.

2) Moving toward the opposite direction ofthe ZRN direction atzero return speed.

3) Decelerating to stop afterdetecting the front end of DOG.(leaving DOG)

4) Moving toward the ZRN direction at zero return speed. (entering DOG again)

5) Starting to decelerate to the crawling speed once the front end of DOG is detected.

6) Stopping once the first zero signal is detected after detecting the back end of DOG.

3. When the starting position is in DOG OFF (after passing DOG):

1)Starting the ZRN action through executing the ZRN instruction.

2) Moving toward the ZRN direction atzero return speed.

3)Decelerating to stop after detecting the reverse limit.

4) Moving toward the opposite direction ofthe ZRN direction at zero return speed.

5) Decelerating to stop after detecting the front end of DOG.(detecting (leave) DOG)

6) Moving toward the ZRN direction at zero return speed.

7) Starting to decelerate to the crawling speed once the front end of DOG is detected.

8) Stopping once the first zero signal is detected after detecting the back end of DOG.

4. When the starting position is in limit switch position (FWD or REV limit):

1)Starting the ZRN action through executing the ZRN instruction.

2) Moving toward the opposite direction ofthe ZRN direction at zero return speed.

3) Decelerating to stop after detecting the front end of DOG.(detecting (leave) DOG)

4) Moving toward the ZRN direction at zero return speed. (entering DOG again)

5) Starting to decelerate to the crawling speed once the front end of DOG is detected.

6) Stopping once the first zero signal is detected after detecting the back end of DOG.

- Note

1. PLSY, PLS, and positioning instructions can output the high-speed pulses through Y0 and Y1. It is not allowed to use these instructions for high-speed pulse output on the same port at the same time.

2. The min. output pulse frequency that can be outputted actually, is determined by the formula below:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ indicates the max. speed, set via SD85 and SD86 while T indicates the acceleration/deceleration time, set via SD87, withthe unit of ms. The calculation result $F_{\min\_acc}$ is the limit value of min. output frequency.

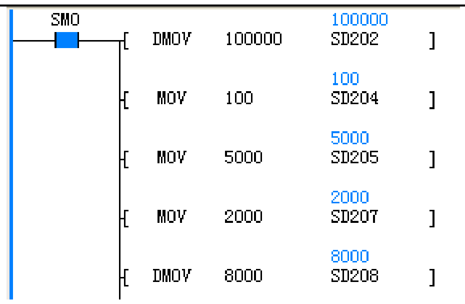
3. For the pulse output frequency, the calculated frequency is output even if a value less than the calculated one is assigned. The frequency of the starting section of ACC and end part of DEC cannot be less than the above calculation result. If the max. speed is lower than the above calculation result, there is not pulse output.

4. Crawling speed needs to be larger than zero and less than one tenth of the max. speed.

5. For details, refer to Chapter 11 "User Guide for Positioning Function".

Application instance

Parameters such as the max. speed, base speed, acceleration/deceleration time, zero return speed, and crawling speed can adopt default values or can be reset through the soft element assignment.



Executingthe DSZR instruction:

6.21.7 DVIT: Interrupt positioning instruction

LAD:	Applicable model	IVC2L IVC1L
├─┤ ├─[DVIT *(S1)* *(S2)* *(D1)* *(D2)*]	Influenced flag bit	
IL: DVIT *(S1)* *(S2)* *(D1)* *(D2)*	**Step length**	**11**

Operand	Type	Applicable soft element														Indexing
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: Specifying the number of output pulses after the interrupt (relative address).

  S2: Specifying the output pulse frequency. IVC2L: 10–100000 (Hz), IVC1L: Y0, Y1 10–100000 (Hz), Y2, Y3 10–10000 (Hz)

  D1:Specifying the output number of the output pulses.For IVC2L, Y0 and Y1 can be designated. For IVC1L, Y0, Y1, and Y2 can be designated.

  D2: Specifying the output object number of the rotation direction signal.

- Function description

  1. When the interrupt is generated, the pulse is output at the specified frequency and the specified number of pulses.SM260 is valid for interrupt input function; SD240 is the designation of interrupt input function; SM287 is the logic inversion soft element of the interrupt input signal. The logic inversion determines whether the interrupt soft element generates an interrupt by ON or OFF.

  2. The designation method of the interrupt input is as below: SM260 is set to ON.

  3. Designating a input number (X0–X7) as the interrupt input in SD240, or designating the user interrupt instruction soft elements, in which the LSB of SD240 correspond to the interrupt input used by the pulse output Y0, and the MSB corresponds to the interrupt input used by the pulse output Y1.

Set value	Setting content	
0	Designating X0 as the interrupt input signal	
1	Designating X1 as the interrupt input signal	
……	……	
7	Designating X7 as the interrupt input signal	
8×1	Designating the user interrupt instruction soft element[×1] as the interrupt input signal	
	Pulse output soft element	User interrupt instruction soft element
	Y0	SM289
	Y1	SM299

- Note

  1. PLSY, PLS, and positioning instructions can output the high-speed pulses through Y0 and Y1. It is not allowed to use these instructions for high-speed pulse output on the same port at the same time.

  2. The min. output pulse frequency that can be outputted actually, is determined by the formula below:

$$F_{min\_acc} = \sqrt{\frac{F_{max} \times 500}{T}}$$

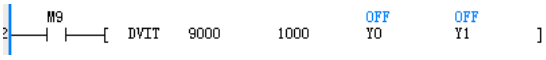
  In the above formula, F_{max} indicates the max. speed, set via SD85 and SD86 while T indicates the acceleration/deceleration time, set via SD87, withthe unit of ms. The calculation result F_{min_acc} is the limit value of min. output frequency.

3. For the pulse output frequency, the calculated frequency is output even if a value less than the calculated one is assigned. The frequency of the starting section of ACC and end part of DEC cannot be less than the above calculation result. If the max. speed is lower than the above calculation result, there is not pulse output.

4. When the number of output pulses is less than the number of pulses required for deceleration, the frequency can be decelerated.

5. For details, refer to Chapter 11 "User Guide for Positioning Function".

- Application instance



6.21.8 STOPDV: Pulse output stop instruction

LAD: STOPDV (S1) (S2) (S3) (D)	**Applicable model**	**IVC3**	
	Influenced flag bit		
IL: STOPDV (*S1*) (*S2*) (*S3*) (*D*)	**Step length**	**12**	

Operand	Type	Applicable soft element													Indexing
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: Number of output pulses after the instruction is executed (relative address).

S2: Base speed during the deceleration.

S3: Time for deceleration from the original output speed to the base speed.

D: Number of the output point corresponding to the high-speed pulses. Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 are optional.

- Function description

1. Executing the STOPDV instruction. It can started during the execution process of the PLSY, PLS, and positioning instructions, and stops the output action of the specified axis.

2. When the drive energy flow of the instruction is ON, the pulse output stops after running the specified number of pulses. When the specified number of pulses is 0, the instruction stops its output actionimmediately. When the specified number of pulses is greater than 0, the instruction continues the original output action first, then decelerates to the base speed, and stops the output action when the base speed is reached.

3. The base speed and acceleration/deceleration time are also set in the special data register of the output axis. The execution of the instruction does 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 set instruction operand instead of using the configuration in the special data register.

4. The direction signal of the output axis does not need to be specified, and the direction signals specified in the original PLSY, PLS, and positioning instructions are automatically recognized. The ON/OFF state of the direction signal is not changed during the execution of the instruction.

- Note

1. The min. output pulse frequency that can be outputted actually, is determined by the formula below:

$$F_{\min\_acc} = \sqrt{\frac{F_{\max} \times 500}{T}}$$

In the above formula, $F_{\max}$ indicates the max. speed, set via SD85 and SD86 while T indicates the acceleration/deceleration time, set via SD87, withthe unit of ms. The calculation result $F_{\min\_acc}$ is the limit value of min. output frequency.

2. For the pulse output frequency ***S2***, the calculated frequency is output even if a value less than the calculated one is assigned.The frequency of the starting section of ACC and end part of DEC cannot be less than the above calculation result. If the max. speed is lower than the above calculation result, there is not pulse output.

3. When the number of output pulses is less than the number of pulses required for deceleration, the frequency can be decelerated.

4. The output point number D corresponding to the high-speed pulses can specify Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7.

5. When the instruction is executed, if the output axis is already in the stop state, no operation is performed; if the output axis is executing the LIN, CW, or CCW instructions, no operation is performed.

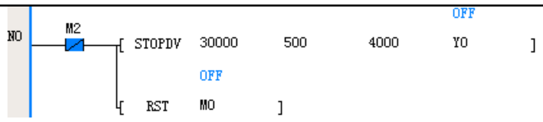
6. For details, refer to Chapter 11 "User Guide for Positioning Function".

- Application instance

In the main program, you can try out the PLSY instruction to drive Y0. The energy flow is controlled by the M0 element:

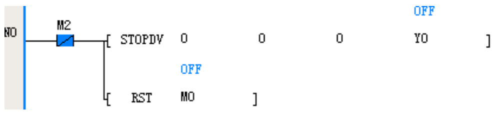
Setting the interrupt source for the interrupt subprogram, for example:

Adding the following statement to the interrupt subprogram:



In the above instruction, when X6 is set to ON, Y0 outputs some pulses and then decelerates to stop. If X6 is set to ON until Y0 stops completely, 30,000 pulses are outputted, which is not affected by the scan cycle.

For example, operand 1 in the STOPDV instruction in INT_1, is 0, the diagram is as shown below:



When the interrupt source event occurs (hereby referring to the rising edge of X6), Y0 stops immediately and is notaffected by the scan cycle.

Note that when the STOPDV instruction is called, it is necessary to cut off the energy flow of the high-speed instruction executed for Y0 in the main function to prevent the high-speed output from being restarted by scanning the instruction in the main function after Y0 is stopped.

6.21.9 LIN: Linear trajectory interpolation instruction

LAD: LIN (S) (D1) (D2) (D3) (D4)	Applicable model	IVC3
	Influenced flag bit	
IL: LIN (S) (D1) (D2) (D3) (D4)	Step length	12

Operand	Type	Applicable soft element													Indexing
S	DINT								D						
D1	BOOL			Y											
D2	BOOL			Y											
D3	BOOL			Y											
D4	BOOL			Y											

- Operand description

S: The start address of the parameter table storage area.

D1: The output point number corresponding to the X-axis pulse signal (or positive pulse signal). Only Y0 can be specified.

D2: The output point number corresponding to the X-axis direction signal (or negative pulse signal). Only Y1 can be specified.

D3: The output point number corresponding to the Y-axis pulse signal (or positive pulse signal). Only Y2 can be specified.

D4: The output point number corresponding to the Y-axis direction signal (or negative pulse signal). Only Y3 can be specified.

- Function description

1. Moving to the target position along a straight trajectory at the specified vector speed.

2. Parameter table definition

D element	Content
S	Reserved

<table>
<tr><th>D element</th><th colspan="3">Content</th></tr>
<tr><td rowspan="6">S+1</td><td colspan="3">Action mode and output logic relationship (configuration code in decimal)</td></tr>
<tr><td>Configuration</td><td>Mode</td><td>Output logic</td></tr>
<tr><td>00</td><td>Incremental type</td><td>Pulse+direction (Forward: ON/reverse: OFF)</td></tr>
<tr><td>01</td><td>Incremental type</td><td>Forward pulse+reverse pulse</td></tr>
<tr><td>10</td><td>Absolute value type</td><td>Pulse+direction (Forward: ON/reverse: OFF)</td></tr>
<tr><td>11</td><td>Absolute value type</td><td>Forward pulse+reverse pulse</td></tr>
<tr><td>S+2</td><td colspan="3">Resultant speed initial speed Fmin (Hz) (MSB)</td></tr>
<tr><td>S+3</td><td colspan="3">Resultant speed initial speed Fmin (Hz) (LSB)</td></tr>
<tr><td>S+4</td><td colspan="3">Resultant speed max. speed Fmax (Hz) (MSB)</td></tr>
<tr><td>S+5</td><td colspan="3">Resultant speed max. speed Fmax (Hz) (LSB)</td></tr>
<tr><td>S+6</td><td colspan="3">Acceleration/deceleration time T (ms) (MSB)</td></tr>
<tr><td>S+7</td><td colspan="3">Acceleration/deceleration time T (ms) (LSB)</td></tr>
<tr><td>S+8</td><td colspan="3">X-axis target position (moving distance) (MSB)</td></tr>
<tr><td>S+9</td><td colspan="3">X-axis target position (moving distance) (LSB)</td></tr>
<tr><td>S+10</td><td colspan="3">Y-axis target position (moving distance) (MSB)</td></tr>
<tr><td>S+11</td><td colspan="3">Y-axis target position (moving distance) (LSB)</td></tr>
</table>

Among which:

(1) In the incremental mode, the trajectory target adopts the relative address, which refers to the moving distance from the current position to the X and Y axes during the target period.

(2) In the absolute value mode, the trajectory target adopts an absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes.

- Note

1. The output point numbers D1 and D3 corresponding to the two output axis pulse signals (or forward pulse signals) in the instruction must be used in groups. When the output group can only designatedas Y0 and Y2, Y1 and Y3 are used in combination with Y0 and Y2 respectively to provide the direction signal or negative pulse output signal.

2. The output group (Y0 and Y2) can be specified as "Pulse + direction" mode or "Positive pulse + negative pulse" mode, the max. speed of the single axis is 200k while the resultant speedis a max.of 200K.

3. The setting range of the moving distance for each axis is -8388608-+8388607 pulses.

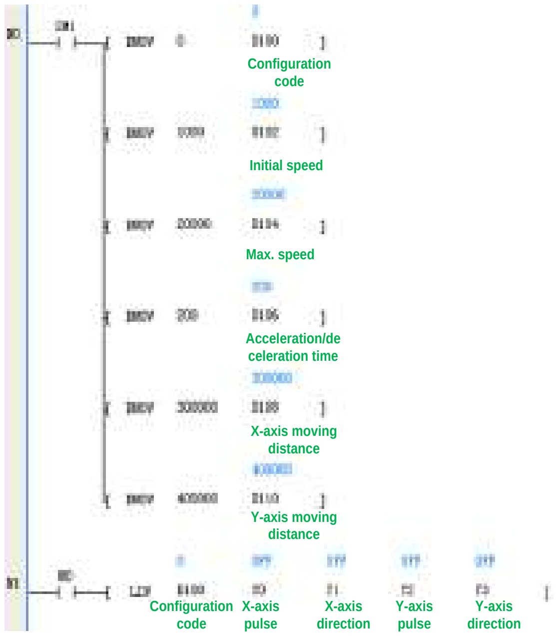
4. It is not allowed to use multiple PLSY, PLS, or positioning instructions on the same high-speed pulse output port at the same time.

5. The range of acceleration and deceleration time is 5-5000ms.

6. Only trapezoidal acceleration/deceleration is supported, and the deceleration time cannot be set alone.

7. The completion interrupt only supports one (Y0Y1) interrupt.

- Application instance



6.21.10 CW: Clockwise arc trajectory interpolation

<table>
<tr><td rowspan="2">LAD:
├─┤ ├──[CW (S) (D1) (D2) (D3) (D4)]</td><td>Applicable model</td><td>IVC3</td></tr>
<tr><td>Influenced flag bit</td><td></td></tr>
<tr><td>IL: CW (S) (D1) (D2) (D3) (D4)</td><td>Step length</td><td>12</td></tr>
</table>

Operand	Type	Applicable soft element													Indexing
S	DINT								D						
D1	BOOL			Y											
D2	BOOL			Y											
D3	BOOL			Y											
D4	BOOL			Y											

- Operand description

S: The start address of the parameter table storage area.

D1: The output point number corresponding to the X-axis pulse signal (or positive pulse signal). Only Y0 can be specified

D2: The output point number corresponding to the X-axis direction signal (or negative pulse signal). Only Y1 can be specified.

D3: The output point number corresponding to the Y-axis pulse signal (or positive pulse signal). Only Y2 can be specified.

D4: The output point number corresponding to the Y-axis direction signal (or negative pulse signal). Only Y3 can be specified.

- Function description

1. Moving to the target position along the arc trajectory in the clockwise direction at the specified line speed.

2. Parameter table definition

<table>
<tr><th>D element</th><th colspan="3">Content</th></tr>
<tr><td rowspan="4">S</td><td colspan="3">Arc formation method</td></tr>
<tr><td>Configuration</td><td colspan="2">Mode</td></tr>
<tr><td>0</td><td colspan="2">Center position designation</td></tr>
<tr><td>1</td><td colspan="2">Passing position designation</td></tr>
<tr><td rowspan="6">S+1</td><td colspan="3">Action mode and output logic relationship (configuration code in decimal)</td></tr>
<tr><td>Configuration</td><td>Mode</td><td>Output logic</td></tr>
<tr><td>00</td><td>Incremental type</td><td>Pulse+direction (Forward: ON/reverse: OFF)</td></tr>
<tr><td>01</td><td>Incremental type</td><td>Forward pulse+reverse pulse</td></tr>
<tr><td>10</td><td>Absolute value type</td><td>Pulse+direction (Forward: ON/reverse: OFF)</td></tr>
<tr><td>11</td><td>Absolute value type</td><td>Forward pulse+reverse pulse</td></tr>
<tr><td>S+2</td><td colspan="3">Resultant speed initial speed (MSB)</td></tr>
<tr><td>S+3</td><td colspan="3">Resultant speed initial speed (LSB)</td></tr>
<tr><td>S+4</td><td colspan="3">Resultant speed (MSB)</td></tr>
<tr><td>S+5</td><td colspan="3">Resultant speed (LSB)</td></tr>
<tr><td>S+6</td><td colspan="3">Acceleration/deceleration time T (ms) (MSB)</td></tr>
<tr><td>S+7</td><td colspan="3">Acceleration/deceleration time T (ms) (LSB)</td></tr>
<tr><td>S+8</td><td colspan="3">X-axis moving distance (target position) (MSB)</td></tr>
<tr><td>S+9</td><td colspan="3">X-axis moving distance (target position) (LSB)</td></tr>
<tr><td>S+10</td><td colspan="3">Y-axis moving distance (target position) (MSB)</td></tr>
<tr><td>S+11</td><td colspan="3">Y-axis moving distance (target position) (LSB)</td></tr>
<tr><td>S+12</td><td colspan="3">X-axis coordinate of the circle center position/passing point (MSB)</td></tr>
<tr><td>S+13</td><td colspan="3">X-axis coordinate of the circle center position/passing point (LSB)</td></tr>
<tr><td>S+14</td><td colspan="3">Y-axis coordinate of the circle center position/passing point (MSB)</td></tr>
<tr><td>S+15</td><td colspan="3">Y-axis coordinate of the circle center position/passing point (LSB)</td></tr>
</table>

Among which:

(1) In the incremental mode, the trajectory target adopts the relative address, which refers to the moving distance from the current position to the X and Y axes during the target period.

(2) In the absolute value mode, the trajectory target adopts an absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes.

- Note

1. The output point numbers D1 and D3 corresponding to the two output axis pulse signals (or forward pulse signals) in the instruction must be used in groups. When the output group can only designatedas Y0 and Y2, Y1 and Y3 are used in combination with Y0 and Y2 respectively to provide the direction signal or negative pulse output signal.

2. The output group (Y0 and Y2) can be specified as "Pulse + direction" mode or "Positive pulse+ negative pulse" mode, and the max. speed is 200k.

3. The setting range of the moving distance for each axis is -8388608-8388607 pulses.

4. It is not allowed to use multiple PLSY, PLS, or positioning instructions on the same high-speed pulse output port at the same time.

5. The coordinates of the passing point specified in the passing position mode refer to the passing position of the entire circle, but the user-defined trajectory path may not pass through this point.

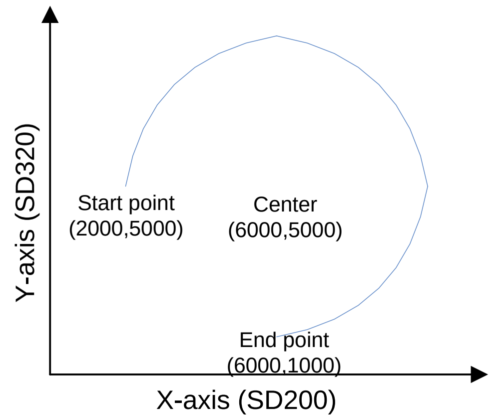
6. The range of acceleration and deceleration time is 5-5000ms.

7. Only trapezoidal acceleration/deceleration is supported, and the deceleration time cannot be set alone.

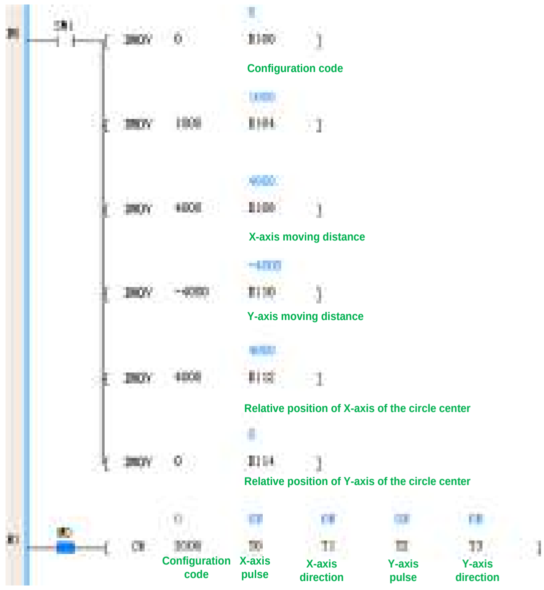
8. The completion interrupt only supports one (Y0Y1) interrupt.

- Application instance

Assuming the current position value of the SD element is (2000, 5000), and you want to draw the arc shown in the following figure:



You can adopt the incremental type, the displacement of the end point to the start point is (4000, -4000), and the displacement of the center to the start point is (4000, 0). You can program like this:



6.21.11 CCW: Counterclockwise arc trajectory interpolation instruction

LAD: ├┤ ├──[CCW (S) (D1) (D2) (D3) (D4)]	**Applicable model**	**IVC3**
	Influenced flag bit	
IL: CCW (S) (D1) (D2) (D3) (D4)	**Step length**	**12**

Operand	Type	Applicable soft element													Indexing
S	DINT								D						
D1	BOOL			Y											
D2	BOOL			Y											
D3	BOOL			Y											
D4	BOOL			Y											

- Operand description

S: The start address of the parameter table storage area.

D1: The output point number corresponding to the X-axis pulse signal (or positive pulse signal). Only Y0 can be specified.

D2: The output point number corresponding to the X-axis direction signal (or negative pulse signal). Only Y1 can be specified.

D3: The output point number corresponding to the Y-axis pulse signal (or positive pulse signal). Only Y2 can be specified.

D4: The output point number corresponding to the Y-axis direction signal (or negative pulse signal). Only Y3 can be specified.

- Function description

1. Moving to the target position along the arc trajectory in the counter clock wise direction at the specified line speed.

2. Parameter table definition

<table>
<tr><th>D element</th><th colspan="3">Content</th></tr>
<tr><td rowspan="4">S</td><td colspan="3">Arc formation method</td></tr>
<tr><td>Configuration</td><td colspan="2">Mode</td></tr>
<tr><td>0</td><td colspan="2">Center position designation</td></tr>
<tr><td>1</td><td colspan="2">Passing position designation</td></tr>
<tr><td rowspan="6">S+1</td><td colspan="3">Action mode and output logic relationship (configuration code in decimal)</td></tr>
<tr><td>Configuration</td><td>Mode</td><td>Output logic</td></tr>
<tr><td>00</td><td>Incremental type</td><td>Pulse+direction (Forward: ON/reverse: OFF)</td></tr>
<tr><td>01</td><td>Incremental type</td><td>Forward pulse+reverse pulse</td></tr>
<tr><td>10</td><td>Absolute value type</td><td>Pulse+direction (Forward: ON/reverse: OFF)</td></tr>
<tr><td>11</td><td>Absolute value type</td><td>Forward pulse+reverse pulse</td></tr>
<tr><td>S+2</td><td colspan="3">Resultant speed initial speed (MSB)</td></tr>
<tr><td>S+3</td><td colspan="3">Resultant speed initial speed (LSB)</td></tr>
<tr><td>S+4</td><td colspan="3">Resultant speed (MSB)</td></tr>
<tr><td>S+5</td><td colspan="3">Resultant speed (LSB)</td></tr>
<tr><td>S+6</td><td colspan="3">Acceleration/deceleration time T (ms) (MSB)</td></tr>
<tr><td>S+7</td><td colspan="3">Acceleration/deceleration time T (ms) (LSB)</td></tr>
<tr><td>S+8</td><td colspan="3">X-axis moving distance (target position) (MSB)</td></tr>
<tr><td>S+9</td><td colspan="3">X-axis moving distance (target position) (LSB)</td></tr>
<tr><td>S+10</td><td colspan="3">Y-axis moving distance (target position) (MSB)</td></tr>
<tr><td>S+11</td><td colspan="3">Y-axis moving distance (target position) (LSB)</td></tr>
<tr><td>S+12</td><td colspan="3">X-axis coordinate of the circle center position/passing point (MSB)</td></tr>
<tr><td>S+13</td><td colspan="3">X-axis coordinate of the circle center position/passing point (LSB)</td></tr>
<tr><td>S+14</td><td colspan="3">Y-axis coordinate of the circle center position/passing point (MSB)</td></tr>
<tr><td>S+15</td><td colspan="3">Y-axis coordinate of the circle center position/passing point (LSB)</td></tr>
</table>

Among which:

(1) In the incremental mode, the trajectory target adopts the relative address, which refers to the moving distance from the current position to the X and Y axes during the target period.

(2) In the absolute value mode, the trajectory target adopts an absolute address, which refers to the absolute position coordinates of the target position on the X and Y axes.

- Note

1. The output point numbers D1 and D3 corresponding to the two output axis pulse signals (or forward pulse signals) in the instruction must be used in groups. When the output group can only designatedas Y0 and Y2, Y1 and Y3 are used in combination with Y0 and Y2 respectively to provide the direction signal or negative pulse output signal.

2. The output group (Y0 and Y2) can be specified as "Pulse + direction" mode or "Positive pulse+ negative pulse" mode, and the max. speed is 200k.

3. The setting range of the moving distance for each axis is -8388608-8388607 pulses.

4. It is not allowed to use multiple PLSY, PLS, or positioning instructions on the same high-speed pulse output port at the same time.

5. The coordinates of the passing point specified in the passing position mode refer to the passing position of the entire circle, and the user-defined trajectory path may not pass through this point.

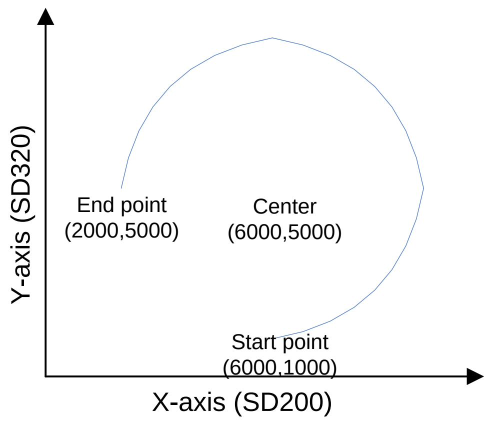
6. The range of acceleration and deceleration time is 5-5000ms.

7. Only trapezoidal acceleration/deceleration is supported, and the deceleration time cannot be set alone.

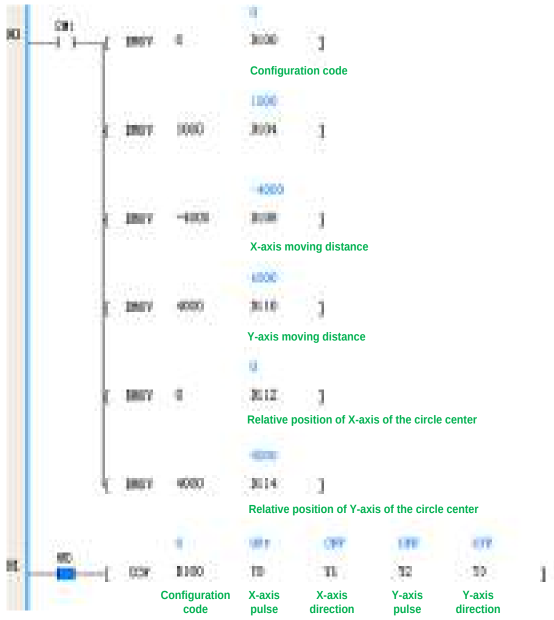
8. The completion interrupt only supports one (Y0Y1) interrupt.

- Application instance

  Assuming the current position value of the SD element is (6000, 1000), and you want to draw the arc shown in the following figure:



You can adopt the incremental type, the displacement of the end point to the start point is (4000, -4000), and the displacement of the center to the start point is (0, 4000). You can program like this:



6.21.12 MOVELINK: Synchronous control instruction

LAD: ─┤ ├──[MOVELINK (S1) (S2) (S3) (S4) (S5) (S6)]	Applicable model	
	Influenced flag bit	
IL: MOVELINK (*S1*) (*S2*) (*S3*) (*S4*) (*S5*) (*S6*)	Step length	17

Operand	Type	Applicable soft element													Indexing
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		
S4	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		
S5	BOOL			Y											
S6	BOOL			Y							C				

- Operand description

_S1_: Total number of pulses generated by the slave axis when it follows the spindle axis.

_S2_: Total number of pulses generated by the spindle axis when it is followed by the slave axis.

_S3_: Number of pulses generated by the spindle axis before reaching the constant speed synchronization

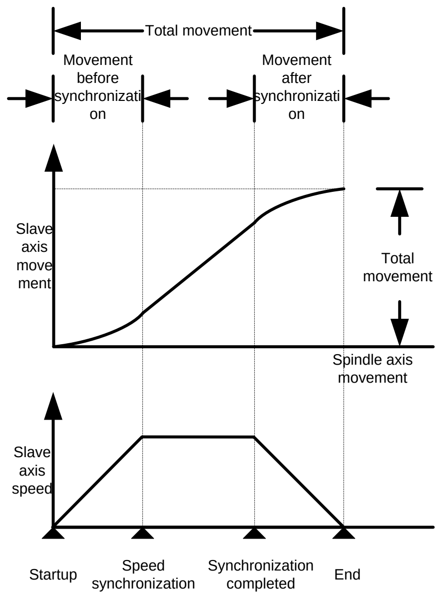
_S4_: Number of pulses generated by the spindle axisafter ending the constant speed synchronization

_S5_: Output point number corresponding to the slave axis pulse signal. Range: Y4 and Y5.

_S6_: Output point number corresponding to the spindle axis pulse signalor high-speed counter number. Range: Y0–Y7, C236–C255, and C301–C306.

- Function description

1. This instruction implements a simple synchronization function for two-axis motion. The slave axis measures and follows the speed and position of the spindle axis. The spindle axis can be used as a high-speed output or input port of the module.



2. As shown in the figure above, after this instruction is started, the slave axis starts from the stop state and moves by following the spindle axis after a period of acceleration and deceleration. When the spindle axis outputs the specified number of pulses **_S3_**, the speed of the slave axis is the same as that of the spindle axis, and the constant speed synchronization starts. When there are S4 pulses to be generated by the spindle axis before it reaches the destination position, the constant speed synchronization is stopped, and the slave axis starts to decelerate the speed. When the spindle axis reaches the target position, the slave axis stops.

3. When the spindle axis is defined as an output axis, operand S6 is set to an output point number Y0–Y7;

when defined as an input axis, operand ***S6*** is set toC236–C255 or C301–C306.

4. When the energy flow of the instruction is turned on, the slave axis starts running. This instruction supports the calling mode and subprogram, and the interrupt program calling methodin the main program.

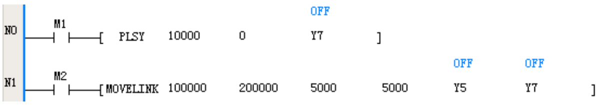
5. The direction signal of the output axis does not need to be specified, and the ON/OFF state of the direction signal is not changed during the following process.

- Note

1. It is not allowed to use multiple PLSY, PLS, or positioning instructions on the same high-speed pulse output port at the same time.

2. The output point number D corresponding to the high-speed pulses can be specified as: Y0, Y2, Y4, Y5, Y6, Y7. Y1 and Y3 are used in combination with Y0 and Y2 respectively to provide the direction signal or negative pulse output signal.

- Application instance



6.21.13 GEARBOX: Electronic gear instruction

LAD: GEARBOX (D1) (S1) (D2) (S2)		**Applicable model**	
		Influenced flag bit	
IL: GEARBOX (D1) (S1) (D2) (S2)		**Step length**	**9**

Operand	Type	Applicable soft element													Indexing
D1	BOOL			Y											
S1	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		√
D2	BOOL			Y											
S2	DINT	Constant	KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C		V		√

- Operand description

D1: The output point number corresponding to the spindle axis signal. Range: Y0–Y7.

S1: This operand together with ***S2*** determines the ratio of the electronic gear.

D2: The output point number corresponding to the slave axis pulse signal. Range: Y4 and Y5.

S2: This operand together with ***S1*** determines the ratio of the electronic gear.

The range of the electronic gear ratio (***S1***/***S2***): 1/10000–10000.

- Function description

1. Enabling the slave axis to follow the spindle axis according to the electronic gear ratio (S1/S2). When the spindle axis sends N pulses, the slave axis sends N × S1/S2 pulses. When the spindle axis outputs pulses at a frequency of F, the slave axis outputs pulses at a frequency of F × S1/S2.

2. When the energy flow of the instruction is turned on, the slave axis starts running.

3. The direction signal of the output axis does not need to be specified, and it does not change its ON/OFF state of the direction signalduring the follow-up process.

4. When the electronic gear ratio is less than 0, that is, one value of S1, S2 is positive and the other is negative, and the relevant SD elements of the slave axis are decreased progressively with the pulse output.

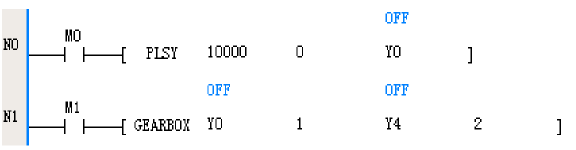
- Note

1. It is not allowed to use multiple PLSY, PLS, or positioning instructions on the same high-speed pulse output port at the same time.

2. If the electronic gear ratio is outside the specified range, there is no pulse output.

3. The output point number D corresponding to the high-speed pulses can be specified as: Y0, Y2, Y4, Y5, Y6, and Y7. Y1 and Y3 are used in combination with Y0 and Y2 respectively to provide the direction signal or negative pulse output signal.

- Application instance



6.22 Data processing instructions

6.22.1 MEAN: Mean instruction

LAD: ├──┤ ├──[MEAN (S1) (D) (S2)]	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: MEAN (*S1*) (*D*) (*S2*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
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: Number of the start word element that stores data on which the mean operation is to be performed

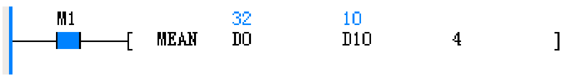
  S2: Number of pieces of data on which the mean operation is to be performed (1–64)

  D: Number of the word element that stores the obtained mean value

- Function description

  1. Storing the mean of S2 16-bit data starting from S1 in D, and rounding off the remainder.

- Application instance



LD M1

MEAN D0 D10 4

Taking the average value of four unit

sof data starting from D0, and storing the obtained value in D10 when M1 = ON. When D0=32, D1=10, D2=15, and D3=-14, D10=10.

6.22.2 WTOB: Byte-unit data separation instruction

LAD: ├──┤ ├──[WTOB (S1) (D) (S2)]	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: WTOB (*S1*) (*D*) (*S2*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

- Operand description

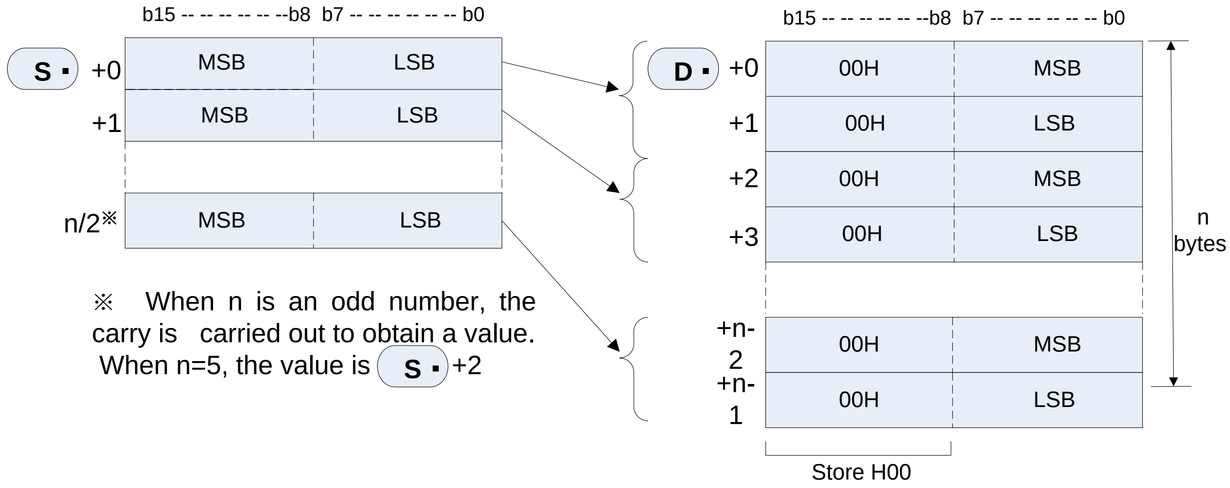
  S1: Number of the start soft element that stores data to be separated in byte unit

  S2: Number of byte data to be separated (S2 ⩾ 0)

  D: Number of the start soft element that stores the result that has been separated in byte unit

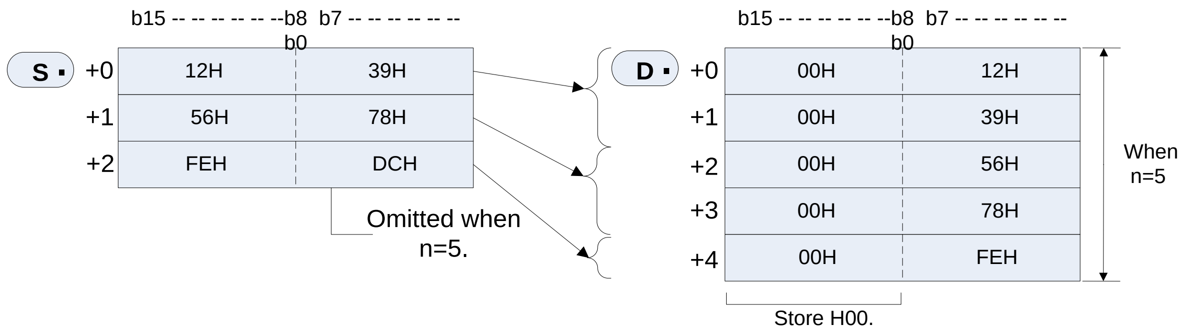
- Function description

  1. Separating the 16-bit data stored in S2/2 soft elements from S1 into S2 bytes, storing them in the low bytes of S2 soft elements starting from D, and clearing the high bytes.



2. When S2 is odd, only the high byte (8 bits) is the object data in the last data of the separated source.

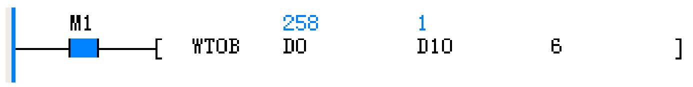
For example, when n=5, the data of the low bytes of S－S+2 are stored in D－D+4.



3. When S2=0, the instruction is not executed.

4. The source and destination operands cannot overlap.

- Application instance



LD M1

WTOB D0 D10 6

Separating three units of data starting from D0 into six units of data according to the high and low bytes, and storing the obtained data in six units starting from D10 when M1=ON. When D0=0x102, D1=0x304, and D2=0x506, D10=0x01, D11=0x02, D12=0x03, D13=0x04, D14=0x05, and D15=0x06.

6.22.3 BTOW: Byte-unit data combination instruction

LAD: BTOW (S1) (D) (S2)	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: BTOW (*S1*) (*D*) (*S2*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

- Operand description

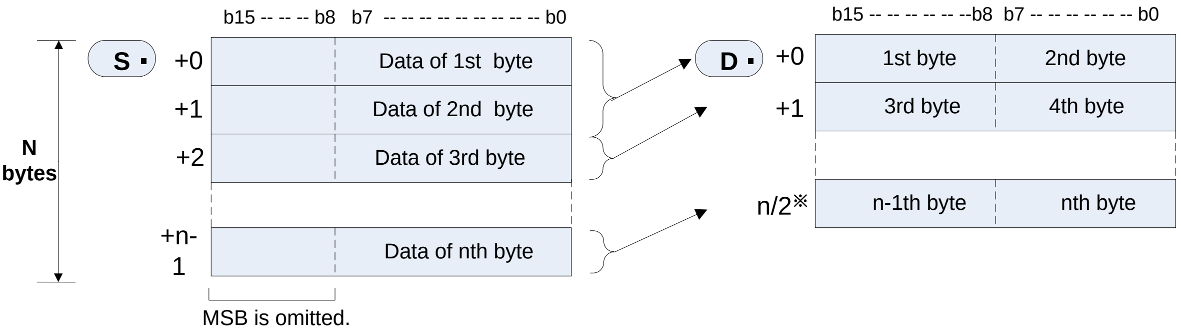
S1: Number of the start soft element stores data to be combined in byte unit

S2: Number of byte data to be combined (S2 ⩾ 0)

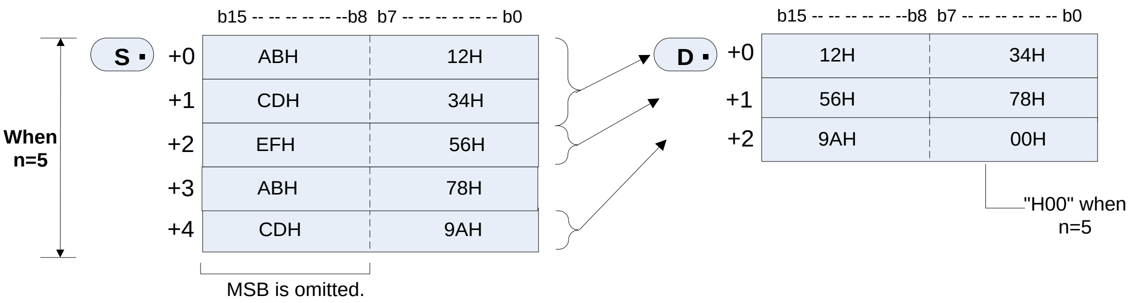
D: Number of the start soft element that stores the result that has been combined in byte unit

- Function description

1. Combining the least significant bytes (8 bits) of S2 16-bit data starting from S1 into 16-bit data, and storing these data in S2/2 soft elements starting from D. The most significant bytes (8 bits after S1) of the 16-bit data in the data source are ignored.



2. When S2 is odd, the least significant byte that is combined finally is cleared.



3. When S2=0, the instruction is not executed.

4. The source and destination operands cannot overlap.

- Application instance

LD M1

BTOW D0 D10 6

Combining six unitsof data starting from D0 into three units of data, and storing the obtained data in three units starting from D10 when M1=ON. When D0=0x01, D1=0x02, D2=0x03, D3=0x04, D4=0x05, and D5=0x06, D10=0x102, D11=0x304, and D12=0x506.

6.22.4 UNI: Instruction for combining 4bits of 16-bit data

LAD: UNI (S1) (D) (S2)	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: UNI (*S1*) (*D*) (*S2*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

- Operand description

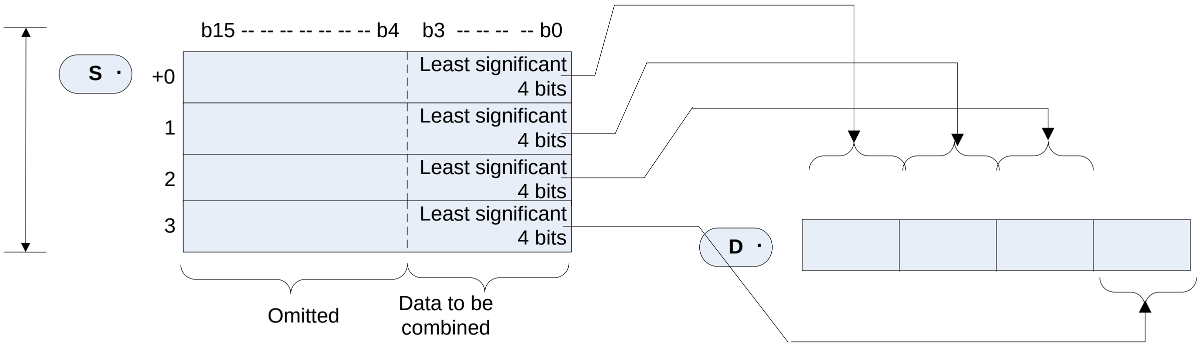
  S1: Number of the start soft element that stores data to be combined

  S2: Number of the combined data (when the range of ***S2*** is 0—4, and S2=0, there is no processing)

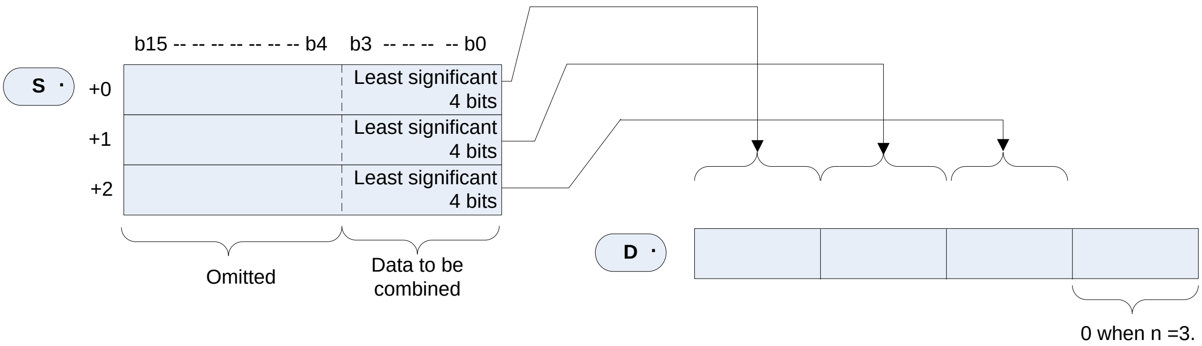
  D: Number of the soft element that stores the combined data

- Function description

  1. Storing S2 16-bit data starting from S1 in S2 soft elements starting from D.



  2. When S2 is ranging from 1 to 3, the ones of the LSB {4 ×(4 -S2)} of D are zero.



  3. When the range of ***S2*** is 1-4, and ***S2***=0, the instruction is not executed.

  4. The source and destination operands cannot overlap.

- Application instance

  LD M1

  UNI D0 D10 4

  Combining the lower four bits of four units of data starting from D0, and storing them in D10. When D0=0x01, D1=0x02, D2=0x03, and D3=0x04, D10=0x1234.

6.22.5 DIS: Instruction for separating 4bitsof 16-bit data

LAD: DIS (S1) (D) (S2)	Applicable model	IVC3
	Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: DIS (*S1*) (*D*) (*S2*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	INT								D	SD	C	T		R	√
S2	INT								D	SD	C	T		R	√
D	INT	Constant							D					R	√

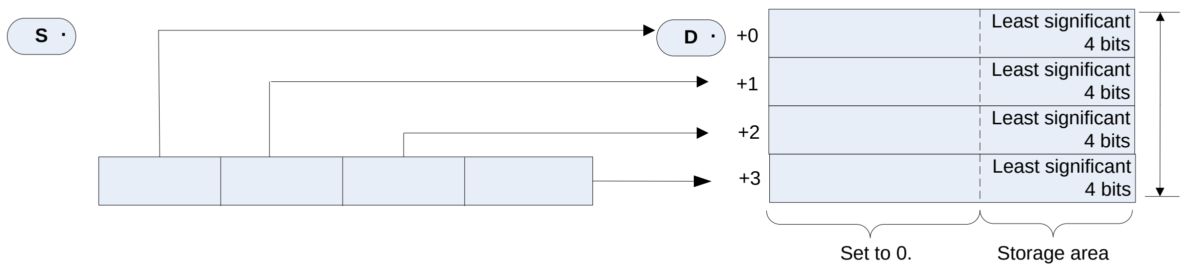
- Operand description

  S1: Number of the start soft element that stores data to be separated

  S2: Number of the separated data (when the range of ***S2*** is 0—4, and ***S2***=0, there is no processing)

  D: Number of the soft element that stores the separated data
- Function description

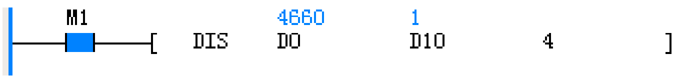
  1. Storing S2 16-bit data starting from S1 in S2 soft elements starting from D.



  2. The valid range of S2 is 1—4, and the rest of the data do not execute the instruction.

  3. The upper 12 bits of S2 soft elements starting from D are cleared.

  4. The source and destination operands cannot overlap.
- Application instance



LD M1

DIS D0 D10 4

Separating every 4 bits of the data in the D0 unit, and storing these data in four units starting from D10 when M1=ON. When D0=0x1234, D10=0x01, D11=0x02, D12=0x03, and D13=0x04.

6.22.6 ANS: Signal alarm set instruction

LAD: ANS (S1) (S2) (D)	**Applicable model**	**IVC3**
	Influenced flag bit	**Zero flag, carry flag, and borrow flag**
IL: ANS (*S1*) (*S2*) (*D*)	**Step length**	**7**

Operand	Type	Applicable soft element													Indexing
S1	INT											T			√
S2	INT	Constant							D					R	√
D	BOOL			S											√

- Operand description

  S1: Timing timer number for judging time, only applicable for 100ms timer with the range of T0-T209

  S2: Data for judging time (1—32767)

  D: Set signal alarm soft elements. Range: S900—S999
- Function description

  1. Setting D when the energy flow duration is greater than S2; resetting the timer S1 and not setting D when the energy flow duration of the instruction is less than S2; resetting S1 when the energy flow is invalid, S1 reset.

Address No.	Name	function
SM400	Enable the signal alarm	When SM400 is set to ON, the following SM401 and SD401 work
SM401	Signal alarm acts	SM401 is set to ON when there is any action in state S900-S999
SD401	Mini.number of the On state	Storing the mini.number of actions in S900-S999

- Application instance

LD M0

ANS T0 100 S901

Setting S901 to on when the energy flow is valid, and not be disconnected within 10 seconds.

6.22.7 ANR: Signal alarm reset instruction

LAD: [ANR]		Applicable model	IVC3
		Influenced flag bit	Zero flag, carry flag, and borrow flag
IL: ANR		Step length	1
Operand	Type	Applicable soft element	Indexing

- Operand description

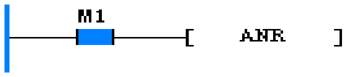
No operand.

- Function description

1. Resetting the running state of the signal alarms S900-S999 when the energy flow is valid; resetting the one with the mini. number if there are multiple state actions;resetting the one with the next mini. number, when the energy flow is valid again.

Address No.	Name	function
SM400	Enable the signal alarm is valid	When SM400 is set to ON, the following SM401 and SD401 work
SM401	Signal alarm acts	SM401 is set to ON when there is any action in state S900-S999
SD401	Mini.number of the on state	Storing the mini.number of actions in S900-S999

- Application instance



LD M1

ANR

When the energy flow is valid, if there are multiple Ss set by ANS, the one with the mini. number is reset.

6.23 Other instructions

6.23.1 RND: Instruction for generating random numbers

LAD: [RND (D)]															Applicable model	IVC3
															Influenced flag bit	Zero flag bit
IL: RND ***(D)***															Step length	3
Operand	Type	Applicable soft element														Indexing
D	WORD		KnX	KnY	KnM	KnS	KnLM	KnSM	D	SD	C	T		Z	R	√

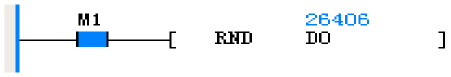
- Operand description

D: Number of the start soft element that stores the random number

- Function description

1. When a pseudo-random number ranging from 0 to 32767 is generated,its value is stored in ***D*** as a random number; if the generated random number is 0, the zero flag bit (SM180) is set.

- Application instance



LD M1

RND D0

Generating a random number,storing it in D0, and D0=26406 when M1=ON.

6.23.2 DUTY: Instruction for generating timed pulses

LAD: [DUTY (S1) (S2) (D)]	**Applicable model**	**IVC3**	
	Influenced flag bit		
IL: DUTY *(S1) (S2) (D)*	**Step length**	**7**	

Operand	Type	Applicable soft element															Indexing
S1	WORD	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: ON scanning times

S2: OFF scanning times

D: Destination address of timed pulse output

- Function description

1. The timed pulse output unit D changes with the ON and OFF scanning times (specified in S1 and S2, respectively).

2. SM unit. Range: SM430-SM434.

Destination address of timed pulse output	Soft elements for counting the number of scans
SM430	SD430
SM431	SD431
SM432	SD432
SM433	SD433
SM434	SD434

3. This instruction can be used for 5 times, but the same timing clock output destination address cannot be used in multiple DUTY instructions.

- Application instance

When M1=ON, 10 scans are ON, and 10 scans are OFF in SM430, and the count value of the scan times is saved in D430 at the same time.

- Note

When the operation starts at the rising edge of the instruction, the energy flow does not stop even if it is cut off. It stops in STOP or upon power outage.

Chapter 7 SFC Tutor

This chapter introduces the basic programming concepts, basic and complex programming methods, and note ofSFC.

7.1 Introduction to SFC

7.1.1 What is SFC

The Sequential Function Chart, or SFC, is a programming language that developed and got popular in recent years. SFC can turn a PLC programming project into a structured flow chart. By using the programming elements and language structure compliant with IEC61131-3 standard, it divides a complex systematic process into sequential multi-step procedures and transitions between the procedures, and thus implements sequential control.

The SFC programming is direct and sequential. Each procedure and conversion condition after decomposition is a relatively simple program process, which isideal for the sequential control application. These advantages explain why it has been widely used.

7.1.2 What is SFC of IVC series PLCs

The SFC of IVC series PLCs is a programming language used by INVT's IVC series PLC products. In addition to the standard SFC functions, one or more LAD program blocks can be built in the SFC of IVC series PLCs.

The program written with SFC of IVC series PLCs can be converted into corresponding LAD or IL program.

The SFC of IVC series PLCs can also support a maximum of 20 independent processes. These independent processes can be run independently, that is to say, the step state within each process are scanned and transferred separately by process. Jumping among the independent procedures is enabled.

7.1.3 Basic concept of SFC

The SFC has two basic concepts: step state and transfer. Other concepts, such as jump, branch and multiple independent processes, are derived from these two basic concepts.

- Stepstate

  1. Definition of step state

  A step state is actually an independent program, representing a working state or an operation in the sequence control process. Putting multiple step states together in a organic way can form a complete SFC program.

  2. Execution of step state

  In a SFC program, each step state is represented by a fixed S element.

  A step state is valid when it is being executed. For a valid step state, the state of its corresponding S element is ON, and the PLC scans and executes all of the instruction sequences within it. A step state is invalid when it is not executed. For an invalid step state, the state of its corresponding S elementis OFF, and the PLC does not scan and execute its instruction sequences.

- Transfer

  The sequence control process is actually a series of step state switching processes. A PLC that is executing a step state leaves the current step state, and enter and execute a new step state if certain logic conditions are met. This switching process is called the transfer of the step state.

  The logical condition that triggers the transfer is called the step transfer condition.

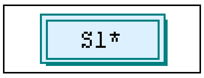
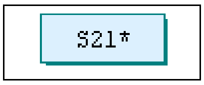
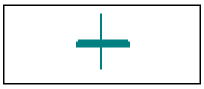
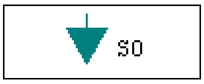
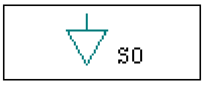
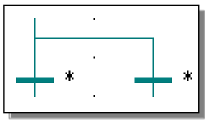
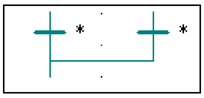
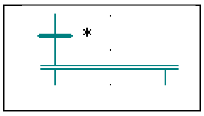
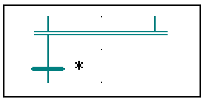
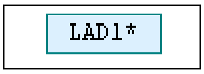
7.1.4 Programming symbols and their usage

- Programming symbols

  The IVC series PLC SFC programming language consists of the following basic programming symbols:

Table 7-1 Programming symbols

Symbol name	Symbol	Description

Symbol name	Symbol	Description
Initial step		An initial step state, numbered as Sn. The "n" cannot be repeated. The execution of a SFC program needs to start with an initial step symbol. The address range of S soft element corresponding to an initial step symbol is S0–S19
Normal step		A normal step state, numbered as Sn. The "n" cannot be repeated. The address range of S soft element corresponding to a normal step symbol is S20--S991
Transfer		A transfer. It can be built-in with a transfer condition (an embedded LAD). You can compile the transfer condition so that the state of S element connected with this transfer is set when the condition is met, and enters the next step state. The transfer symbol needs to be used between steps
Jump		A jump symbol, used after the transfer symbol. It can set the specified S element to ON when the transfer conditions are met. It is used to cycle or jump among the step state.
Reset		A reset symbol, used after the transfer symbol. It can set the specified S element to OFF when the transfer conditions are met. It is used to end the SFC program.
Selection branch		Multiple independent transfer conditions, used after a step symbol. When the transfer condition of one branch is met, the last step state ends, and the next step state of the corresponding branch starts.It is used to select one of multiple step branches. After selecting one branch, no other branches can be selected.
Selection merge		A merge of step branches, connected at the junction of the selection branches. When the transfer condition of one branch is met, the last step state ends, and the next step state takes effect.
Parallel branch		Connected after a step symbol, and the subsequent parallel branches share the same transfer condition. When the transfer condition is established, the subsequent parallel branches are validated and executed at the same time.
Parallel merge		A merge of parallel branches, connected at the junction of the parallel branches. When the multiple parallel step branches are executed and the transfer conditions are met, the next step state takes effect.
LAD block		The LAD block is used to represent LAD instructions other than the SFC flow. It can be used to start the initial step and other general operations.

- Usage of programming symbols

  1. An initial step symbol can be used alone. It cannot be preceded by other symbols, and can be followed only by the transfer symbols.

  2. A LAD block cannot be connected with other symbols.

  3. A normal step symbol can only be connected with a transfer symbol. The normal step symbol cannot exist alone in the diagram.

  4. The reset and jump symbols need to be preceded by transfer symbols and followed by nothing.

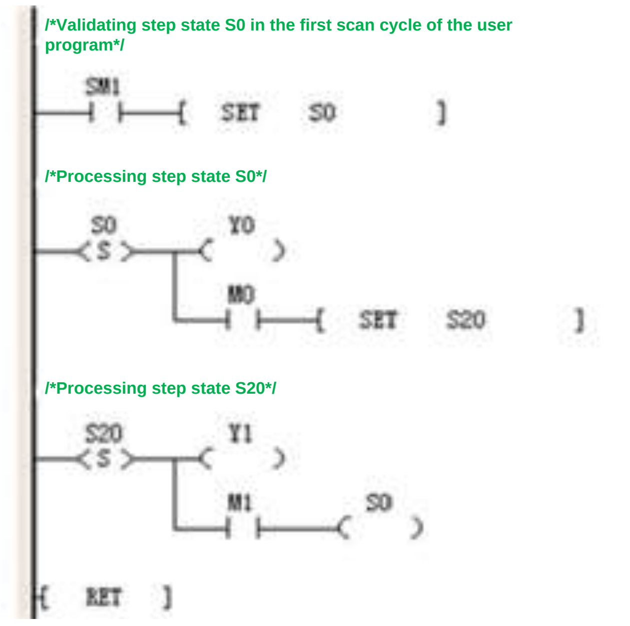
  5. Neither a transfer symbol nor a jump symbol can exist alone in a program.

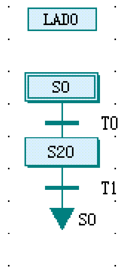
7.1.5 SFC program structure

The process structure of a SFC is classified into three types: simple sequential structure, selection branch structure and parallel branch structure. Besides, the jump structure is also a special form of the selection structure.

- Simple sequential structure

  The following figure shows a simple structured SFC program and its LAD counterpart.





In a simple structured SFC program, when the step transfer conditions are met, the program is sequentially transferred from the current step state to the next step state without any branch structure.At the last step, when the transfer conditions are met, the SFC program section either ends or transfers to the initial step state.

1. LAD block

The LAD block is used to start the SFC program section. To be specific, setting the S element of the initial step symbol to ON. In the preceding figure, the program uses the power-on startup mode.

The LAD block can also be used in other general program sections except the SFC program.

2. Initial step state

As shown in above figure, the initial step state is started by a LAD block. The range of S elements for initialstep stateis 0－19.

3. Normal step symbol

The normal step symbol is used for programming in sequential processes. The range of S elements for normalstep state is 20－991 (applicable for IVC2 series) or 20－1023 (applicable for IVC1 series).

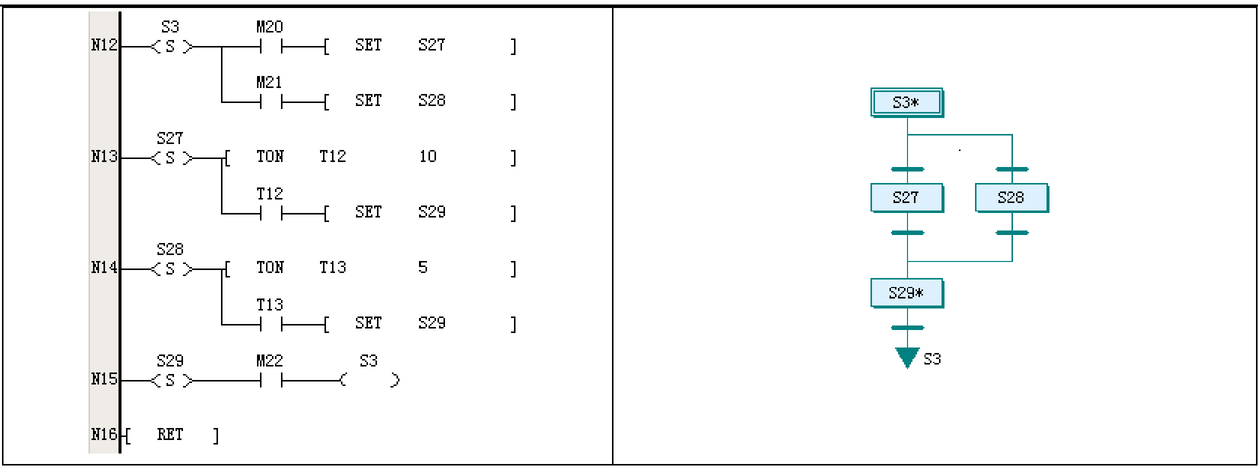
4. Transfer or reset

The last transfer symbol of the program shown in the above figure is connected with a jump symbol, which leads the program to jump to the initial step state. This is a cyclic program.

However, the last transfer symbol of the program can be connected with a reset symbol, which can reset the last step state. After the reset, the program ends, andwaits for the next round of execution.

- Selection branch structure

The following figure shows a selection branch structure, with LAD on the left and SFC on the right.



1. Selection branch

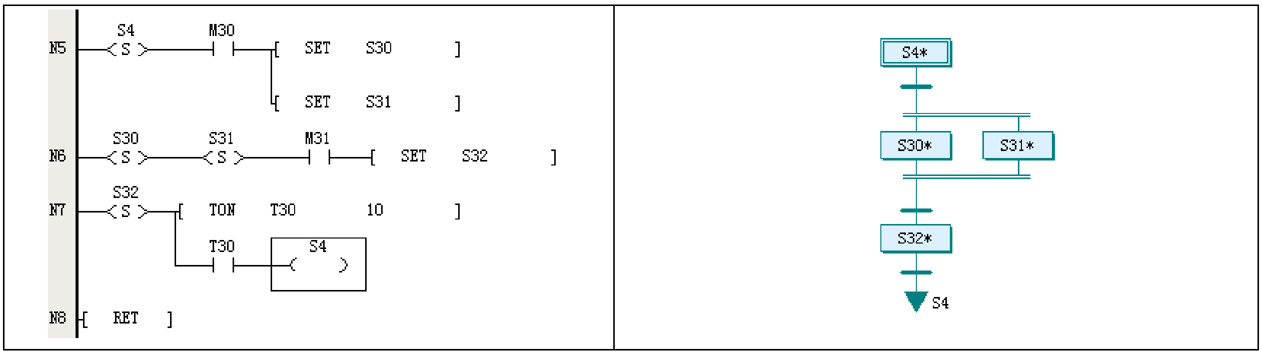
The step state of a branch is validated when its corresponding transfer conditions are met. You need to ensure that the transfer conditions of different branches are all exclusive, so as to make sure that each time only one branch is selected.As shown in the preceding figure, S27 and S28 in row N12 of LAD program are transferred from conditions M20 and M21 respectively. The conditions M20 and M21 must not be set at the same time in order to ensure that S27 and S28 are not be selected at the same time.

2. Selection merge

The selection merge is the structure where all selection branches merge to the same step state. The transfer conditions areset respectively. As shown in the preceding figure, the transfer condition in the step state S27 in row N13 is that time is up for T12,while that in the step state S28 in row N14 is that time is up for T13. However, the results are the same: step state S29 starts.

- Parallel branch structure

The following figure shows a parallel branch structure, with LAD on the left and SFC on the right.



1. Parallel branch

When the transfer conditions of the parallel branch structure are met, each step state connected to the parallel branch structure is simultaneously activated. This is also a common sequential control structure, which enables the PLC to process multiple procedures at the same time. As shownin row N5 program of the preceding figure, S30 and S31 are validated at the same time when transfer conditionM30 is met and M30 is set.

2. Parallel merge

When the transfer conditions of the parallel merge structure are met, each step state connected to the parallel merge structure becomes invalid at the same time. As shownin row N6 program of the preceding figure, when the program is running both S30 and S31 at the same time, and M31 is set, the program starts S32 and ends S30 and S31.

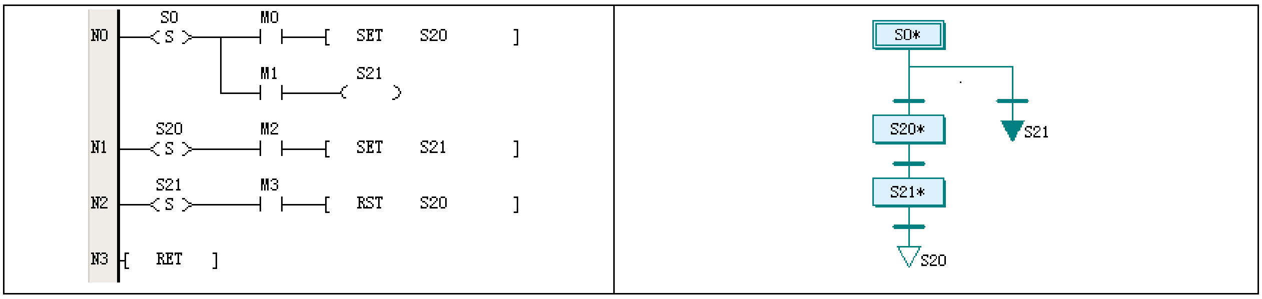
The transfer condition for the parallel mergeis that the transfer action can be executed only when all individual steps are finished before merging.

- Jump structure

The jump structure is often used for the following purposes: to omit certain step states, to cyclically return to the initial step state or the normal step state, andto jump to other processes.

1. Omitting certain step states

In a process, when sequential execution is not required under certain transfer conditions, the program can jumps directly to the needed step state and omits the unnecessary step states, as shown in the following figure, with LAD on and left and SFC on theright.



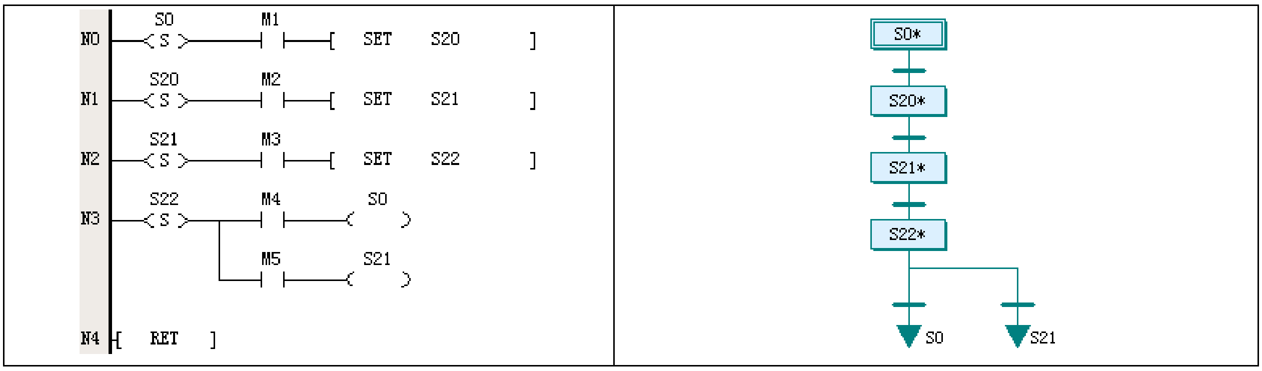
In the SFC program shown in the preceding figure, the jump symbol S21 is used as the jump, while the step state S20 is skipped. In fact, a branch structure is selected before the jump is performed.

In the LAD, the second branch in row N0 is a jump instruction, which adopts the form of an OUT coil instead of an form of the SET instruction that issequentially transferred. When running in the step state S0 and M1 is ON, the program jumps to the step state S21.

2. Cycling

In a process, when it is necessary to cycle a part or all of the step states under certain transfer conditions, you can use a jump symbol to implement the cycle function. At the end of this process, a part of the step states can be cycled if the program jumps to the previous normal step symbol, or all step states can be cycled if the program jumps to thei nitial step symbol.

The following figure shows a program that realizes the above two cycle structures at the same time, with LAD on the left and SFC on theright.



In the SFC, when the step state S22 is valid and one of the transfer conditions is met, the program jumps to S21 and re-run the step state S21. Under another transfer condition, the program jumps to the initial step state S0 and re-run allstep states.

In LAD, these two kinds of jumps are realized in row N3, where you can see the OUT coil of the jump instruction.

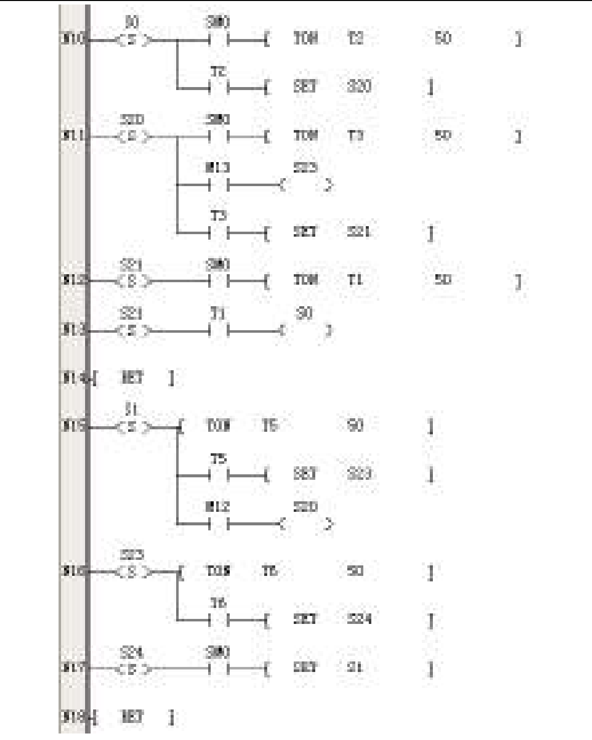
3. Jumping among different independent processes

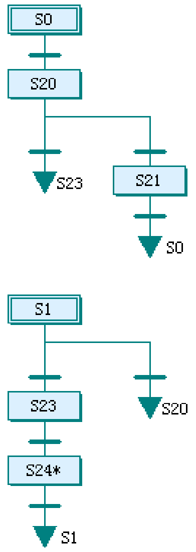
The SFC of IVC series PLCs support multiple independent processes at the same time, and jumping among these processes is enabled. You can set certain transfer conditions in an independent process, and when the conditions are met, it is directly transferred to another independent process, or it jumps to a random step state (initial ornormal) of another independent process.

Note

Jumping among multiple independent processes complicates the PLC program. You need to use it with prudence.

The following figure shows a program that realizes a jump from one independent process to another, with LAD on the left and SFC on the right.





In the SFC, when the step state S0 in the first process is valid, the program can jump to the step state S23 in the second process

under certain transfer conditions; while the step state S1 in the second process is valid, the program can also jump to the step state S20 in the first process under certain transfer conditions.

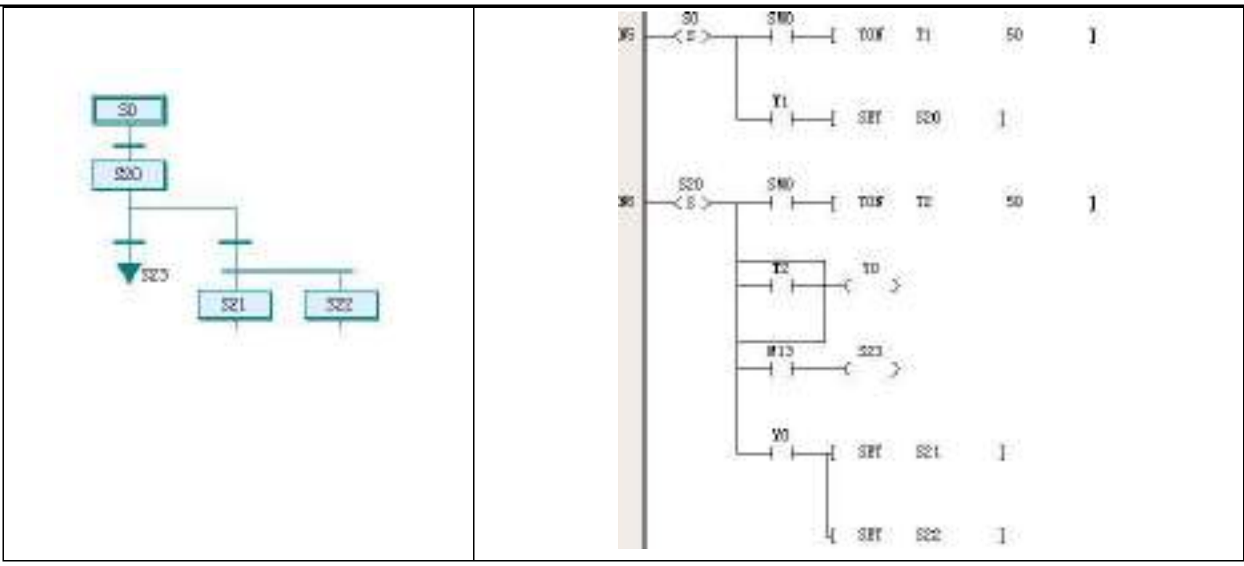
As shown in the preceding figure, the jump is based on a selection branch structure. When the program jumps to another process, all step states in the original process become invalid. As shown in the example, if the program jumps to the step state S23 in the second process from the step state S20 in the first process, the S20 is set to OFF and all step states (S0, S20, and S21) in thefirst process become invalid (OFF).

7.1.6 Execution of SFC program

The similarity between the execution of a SFC program and that of a LAD program is that they both carry out the continuously cyclic scanning from up to down and from left to right.

On the other hand, their difference lies in: in a SFC program, the step states' validity can change according to the sequence conditions, and only internal instruction sequences in the valid step states can be scanned and executed. While in the main program of a normal LAD, all programs are scanned and executed in each scan cycle.

As shown in the following figure, the program on the right is the LAD counterpart of the SFC program on the left. When the step state S20 is valid, the T2 timer is scanned and starts timing. The step states S21 and S22 are not be executed before T2 timer reaches the preset value, and S23 is not executed when M13 is OFF.



The state of S elements is switch between ON and OFF according to the transfer conditions, thus making the program transfer from one step state to another. When a S element changes from ON to OFF, the output elements of the corresponding step state are reset or cleared. For details, refer to section 5.3.1 "STL: SFC state loading instruction".

Note

1. The SFC program of IVC series PLCs usually contains SFC and LAD blocks. The LAD blocks are used to handle operations outside the process, including starting the SFC. The LAD blocks are not controlled by the S elements, and the program rows of these LAD blocks scanned by the PLCs are executed in every scan cycle.

2. The state change of the S element affects the built-in instructions of the corresponding step states, and the switch-over between two step states takes some time, so it is necessary to observe certain rules on the operation of certain soft elements and usage of the instructions during the SFC programming.

7.2 Relationship between SFC and LAD

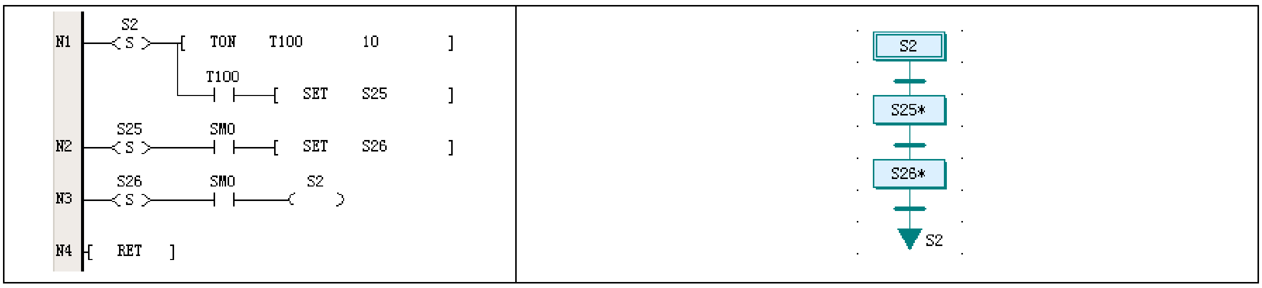
A SFC program can take the form of a LAD, which can help you understand the practical significance of the SFC program structure.

In a LAD, the SFC symbols are replaced with various corresponding SFC instructions, while the corresponding processes are represented by various structures.

7.2.1 STL instruction and step states

In a LAD, a step state is started by a STL instruction.All step states are represented by S elements.

The following diagram on the left is a LAD program of a simple sequential structure, and the diagram on the right is it scorresponding SFC program.



As shown in the LAD program, the step state S2 starts with a STL instruction, and the following TON instruction is the internal instruction sequence of S2. An internal instruction of the step state can be made up of multiple statements, which is actually a relatively complete program section, and almost consistent with the normal LAD counterpart.

The difference between an initial step state and a normal step state only lies in the range in which the S elements are selected.

For details about the STL instruction, refer to section 5.3.1 "STL: SFC state loadinginstruction". You need to note that when the step state changes from ON to OFF, the destination operands corresponding to the built-in instructions (OUT, TON, TOF, PWM, HCNT, PLSY, PLSR, DHSCS, SPD, DHSCI, DHSCR, DHSZ, DHST, DHSP, and BOUT)are cleared.

Note

Because the PLCs run in continuous scan cycles, after a step state transfer, those built-in statements of the original step state are not be affected by the transfer of ON to OFF until the next scan. For details, refer to section 7.4.1 "Common programming errors".

7.2.2 SET instruction

As shown in the preceding diagram, the transfer symbols in the SFC program on the right are realized through the SFC state transfer instruction.

The transfer conditions consist of the NO contacts before the SET statements. The NO contacts are controlled by built-in instruction statements or through the external operations.

Setting the specified step state to be valid when the energy flow of the SFC state transfer instructionis valid, and setting the current valid step state to be invalid to complete the state transition.

7.2.3 RET instruction and SFC program segment

As shown in the preceding diagram, the SFC program on the right starts with a S2 initial step symbol, and returns to S2 after passing two ordinary step symbols. While in the LAD, the end of a segment of the SFC program needs to be marked with the RET instruction.

The RET instruction can be used only in the main program.

7.2.4 OUT instruction and RST instruction

As shown in the preceding diagram, the jump to S2 is realized in LAD by the N3 row, which uses an OUT instruction. The destination operand of the OUT(jump) instruction can be in any independent procedure.

If the reset symbol S26 is used, line N3 in the LAD is a RST instruction, which can reset the last step state S26.

7.2.5 SFC selection branch, parallel branch and merge

For details about LAD counter part of SFC selection branches, refer to selection branch structure of section 7.1.5 "SFC program structure".

For details about LAD counterpart of SFC parallel branches, refer to parallel branch structure of section 7.1.5 "SFC program structure".

7.3 How to program With SFC

1. Analyzing the work flow and determining the program structure

The structure of a SFC program is classified into three types: simple sequential structure, selection branch structure and parallel branch structure. Besides, the jump structure is also a special form of the selection branch structure.To program with SFC, the first step to do is to determine the structure of the flow. For example, a single object passing through a sequential flow is a simple sequential structure. Multiple objects with different parameters to be processed asynchronously needs a selection branch structure. While a cooperation of multiple independent mechanical elements may need a parallel branch structure.

2. Determining the major procedures and transfer conditions to draw a draft of the flow chart

After determining the structure, you need to figure out the major procedures and transfer conditions. By dividing the work flow into smaller operation stages, you can get the procedures. Ending each procedure with a transfer condition,and then you can get a draft of the work flow.

3. Making a SFC program according to a draft of the flow chart

Using the SFC programming language in Auto Station to make a SFC program out of a draft of the flow chart. By now you have got an executable PLC program, but the program needs to be improved.

4. Making a list of I/O pointsand determining the object of each procedure and actual transfer conditions

Generally, the input points are mostly transfer conditions while the output points are mostly operation objects. In addition, with the list, you can further modify the SFC.

5. Inputting the steps and transfer conditions

In the SFC programming interface, you can right click a SFC symbol and select **Embedded LAD** item in the shortcut menu. Then you are able to open the built-in LAD editing workspace of this symbol, and input the LAD programs and transfer condition.

6. Programming the LAD program blocks

You need to remember to program some program blocks that provide general functions, such as start, stop and alarm functions of the sequence process. Such program blocks need to be placed in LAD blocks.

Note

The start and stop operations are crucial for personal and equipment safety. Considering the special features of SFC program, you need to make sure that all outputs that should be stopped are shut down when the PLCs are stopped.

7.4 Notesin SFC programming

The STL statement has some special characteristics, and the PLCs periodically scan the instruction statements by their display order.Because of these reasons, there are several important notes in SFC programming.

7.4.1 Common programming errors

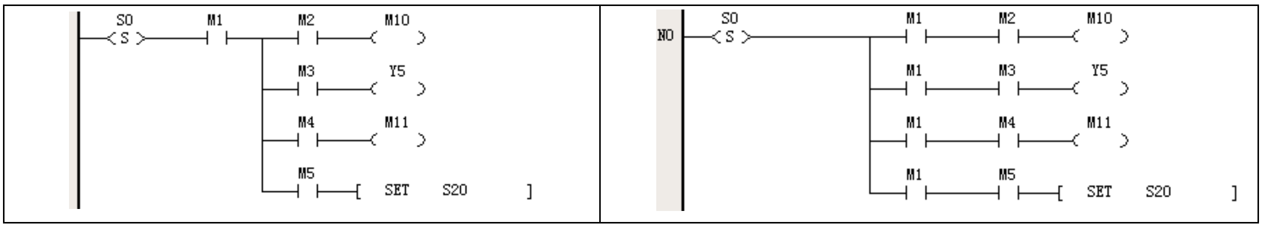
1. Reusing the stepstate symbols

In the same PLC program, each step state symbol used for sequential control programming corresponds to a unique S element and cannot be reused.

You need to note this requirement when editing a SFC program using the LAD editor.

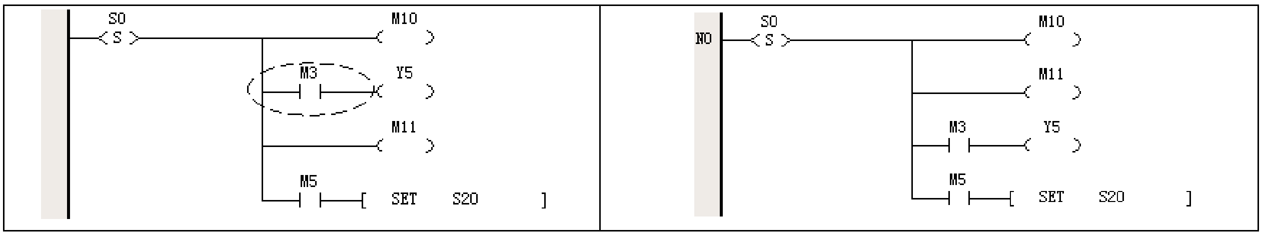
2. Setting branches after a transfer condition

Setting conditioned branches after a transfer condition is disabled in SFC programming, as shown in the left diagram below. Because M1 has become a transfer condition and cannot be branched. Instead, you can change it into the right diagram below, which can be compiled correctly.



3. Connecting output coils to internal bus after a NC or NO contact instruction

When a NO or NC contact instruction is used in a branch, the output coils in the subsequent branches are prohibited to connect the internal bus, as shown in the left diagram below. Instead, you can change the branch order into the one shown in the right diagram below, which can be compiled correctly.

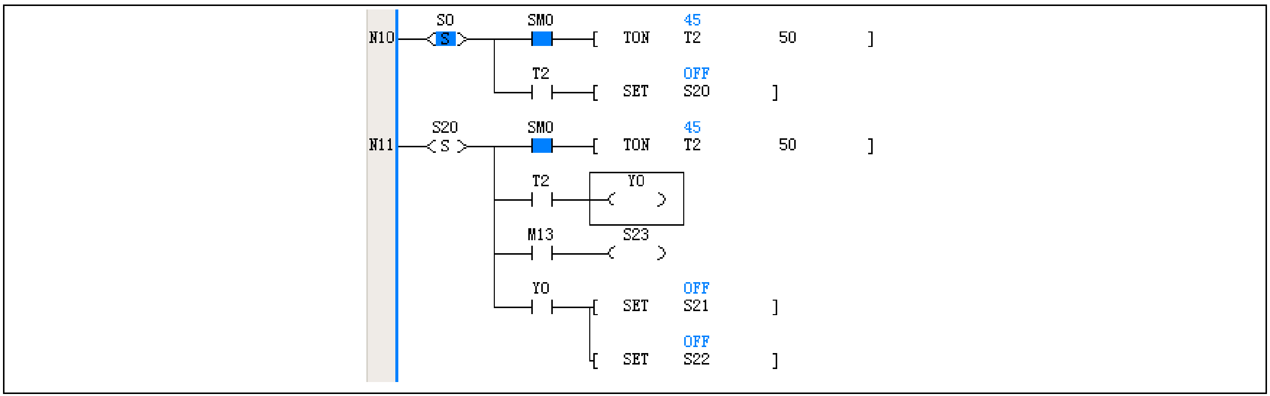


4. Reusing the same soft element in the neighboring step states

When the PLCs execute the program, the instructions are cyclically scanned by their display orders. When the program shifts from the previous step state to the next step state, the sequence of instructions in the previous step state just ends the scan, and the sequence of instructions in the next step state has also been turned on to form a control output.

Therefore, when the STL instruction is turned from OFF to OFF, certain internal elements of the instruction are reset (For details, refer to section 5.3.1 "STL: SFC state loading instruction"), but the reset can only be carried out during the next scan cycle. That is to say, at the moment of transferring the step state, the internal elements of the last step state retains their original data and states until the step state is scanned in the next cycle.

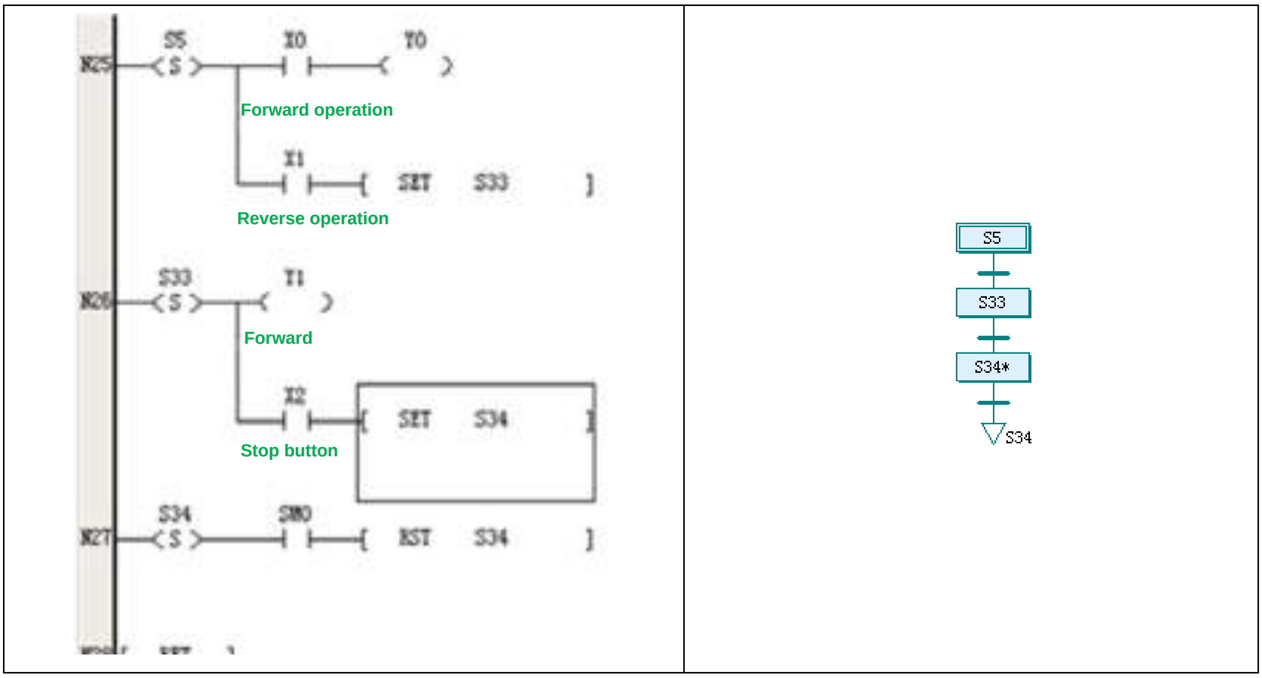
As shown in the following diagram, the two neighboring step states use the same timer: T2. When the step state is shifted from S0 to S20,the T2 element keeps its timing value and state, rendering the step state S20 unable to perform the timing operation as it is originally designed by the user. The program will jump directly to S21 and S22. Therefore, you need to note that reusing soft elements in a program is not prohibited, but it is best not to reuse them in the neighboring step states, otherwise unexpected results may occur.



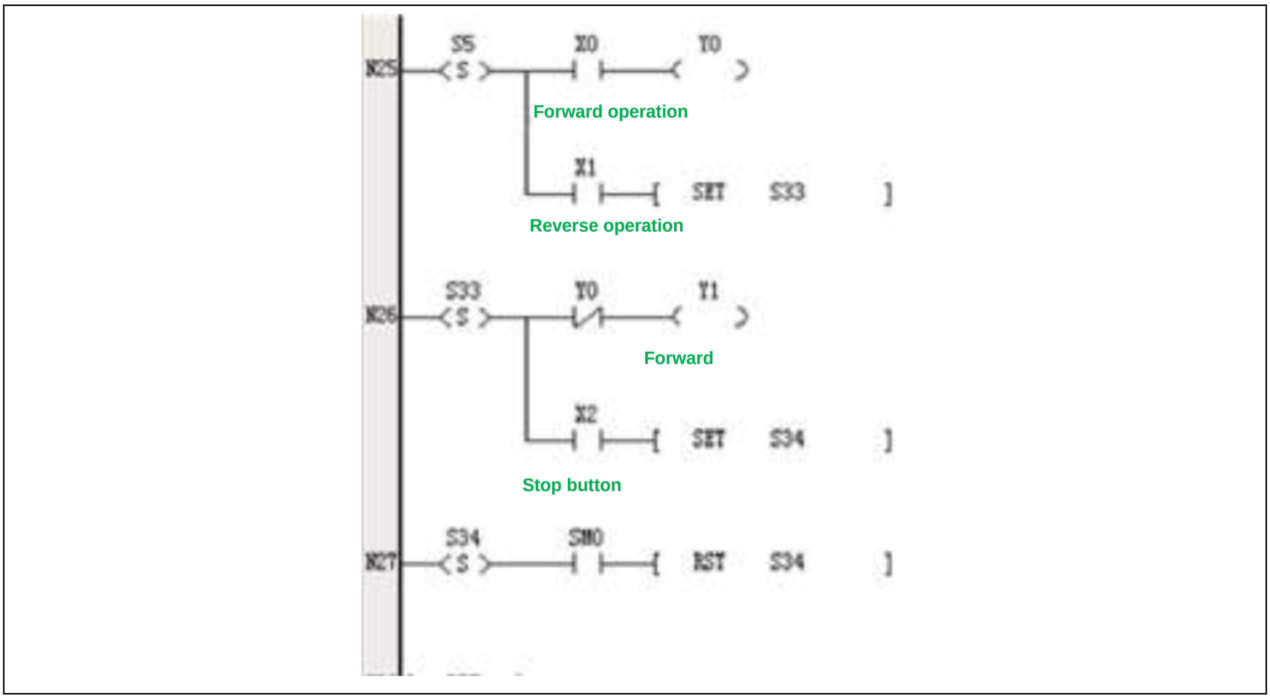
5. Failing to interlock the soft elements

During the SFC programming, certain soft elements may become contradictory to each other under some special transfer conditions of the step states. At this time, inter-locking isnecessary.

Taking the following forward and backward operation program as an example, where Y0 and Y1 are respectively forward and backward control output. X0 is forward operation, X1 is backward operation, and X2 is stop button. It is required that Y0 and Y1 are interlocked, that is to say, they cannot be ON at the same time.However, in this example, when the device runs forward, and X1 is turned ON to shift the step state from S5 to S33, Y0 and Y1 are both ON in one program scan cycle.



Therefore, you need to add the interlock statements to the program by adding a Y0 NC contact before the Y1 output coil, as shown in the following diagram.

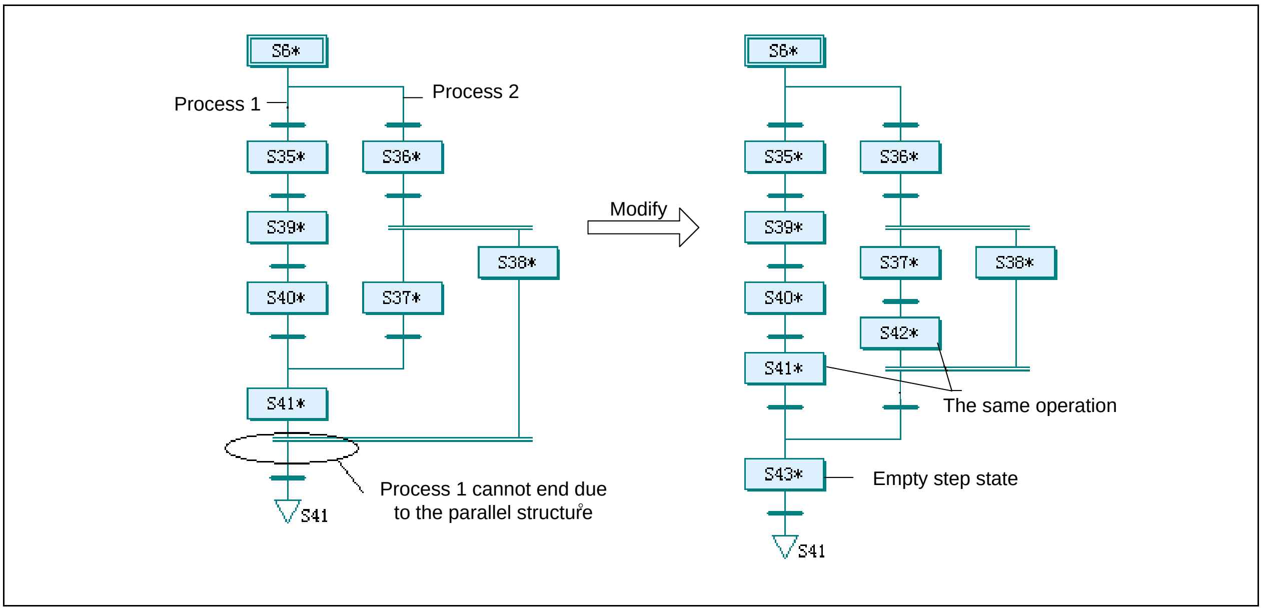


6. Confusing jumps with transfers

Jumps are mostly used between different processes and non-neigh boring step states, while transfers are used between the neighboring step states. It is prohibited to change an output coil into a SET instruction statement where a jump should be used, or change a SET instruction statement into an output coil where a transfer should be used.

7. Selecting a parallel branch structure as the branch transfer point, causing the failure to end the process

A branch is selected from multiple ones. If parallel branches are included, the process may never be ended, as shown in the following figure. In the program on the left,when process 1 runs to the step state S41, the transfer condition is a parallel branch, and therefore the system does not run process 2,causing the failure to implement the transfer condition, and thus causing a process error.



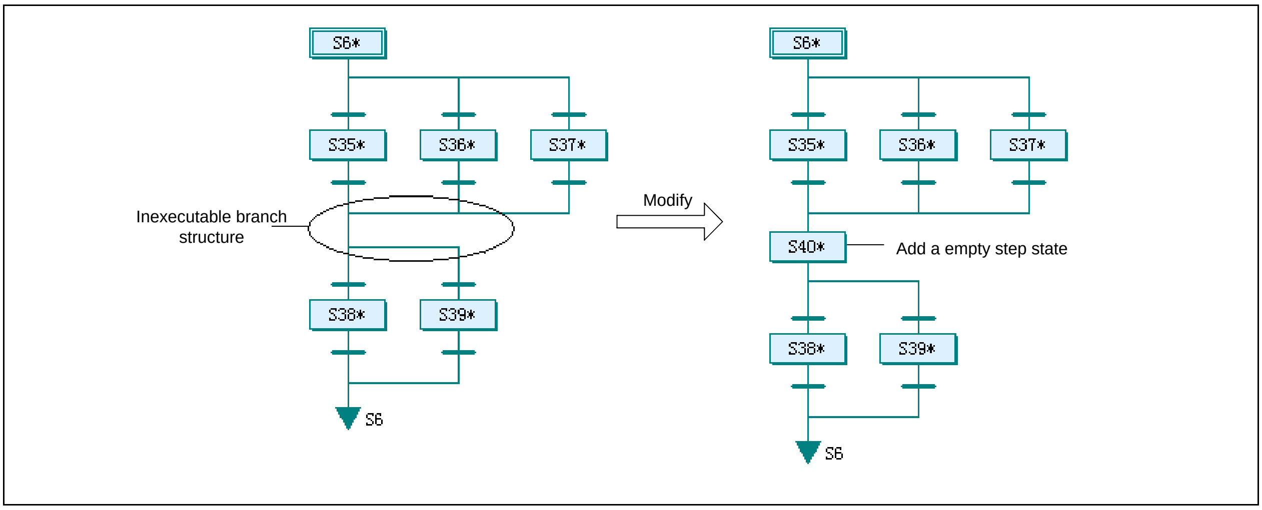
The modification method is shown in the program on the right, you need to add a step state S42 whose function is the same as S41. Then you need to add an empty step state S43 that serves as a programming structural element without actual function. The transfer conditions of S38, S41, and S43 need to be designed by the programmers, for example, all of them can adopt the transfer conditions of the original S41.

7.4.2 Programming tricks

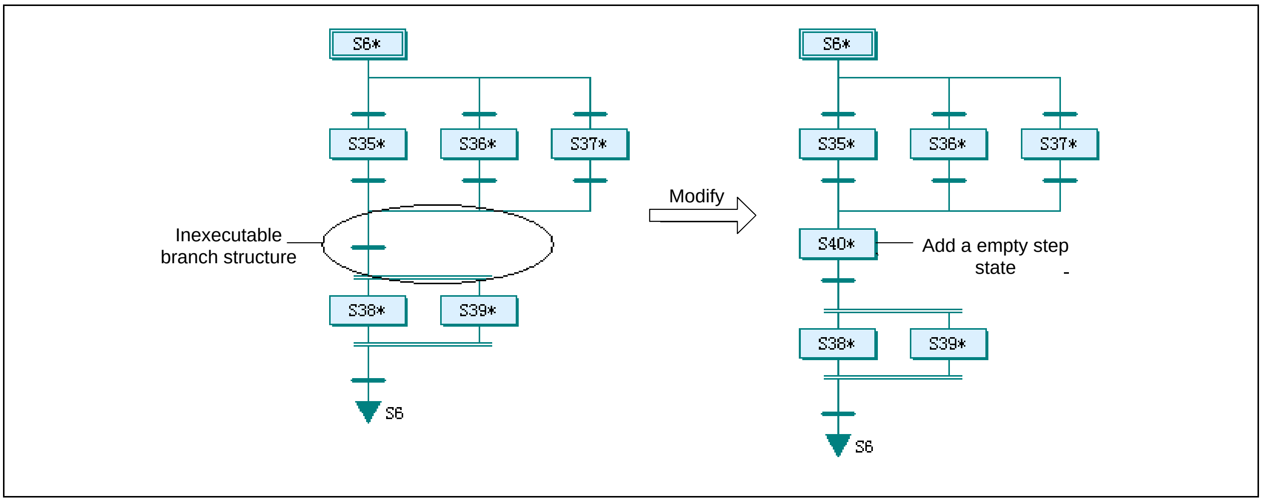
1. Making use of empty step states

You can use empty step states to deal with the branches with grammatical problems. The empty step states do not provide actual operation, but a necessary node in structure before the next transfer. You can see the following example.

In the left diagram below, the selection merge is connected immediately with another selection branch structure, which is prohibited. You can modify it by adding an empty step stateas shown in the right diagram.



In the left diagram below, the selection merge is connected immediately with a parallel branch structure, which is prohibited too. You can also modify it by adding an empty step state as shown in the right diagram.

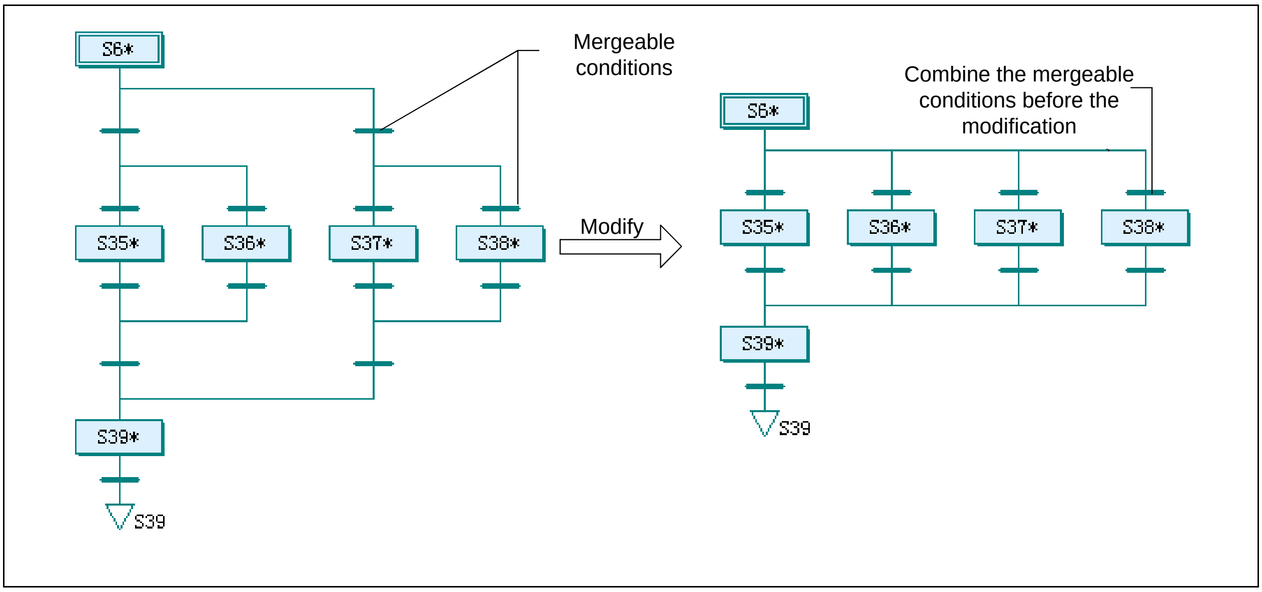


You can address other tricky structures, such as parallel merge connected with parallel branches, or parallel branches connected with selection branches, by adding an empty step state.

2. Merging branches and transfer conditions

Some branches that seem complicated are actually caused by improper designand can be merged or simplified appropriately.

As shown in the following diagram, the designer set a selection branch first, and then set two selection branches. In fact, only one selection branch of four branches can achieve the same, and the two levels of transfer symbols of the original design can be merged into one.



3. Using the function for saving data at power outage

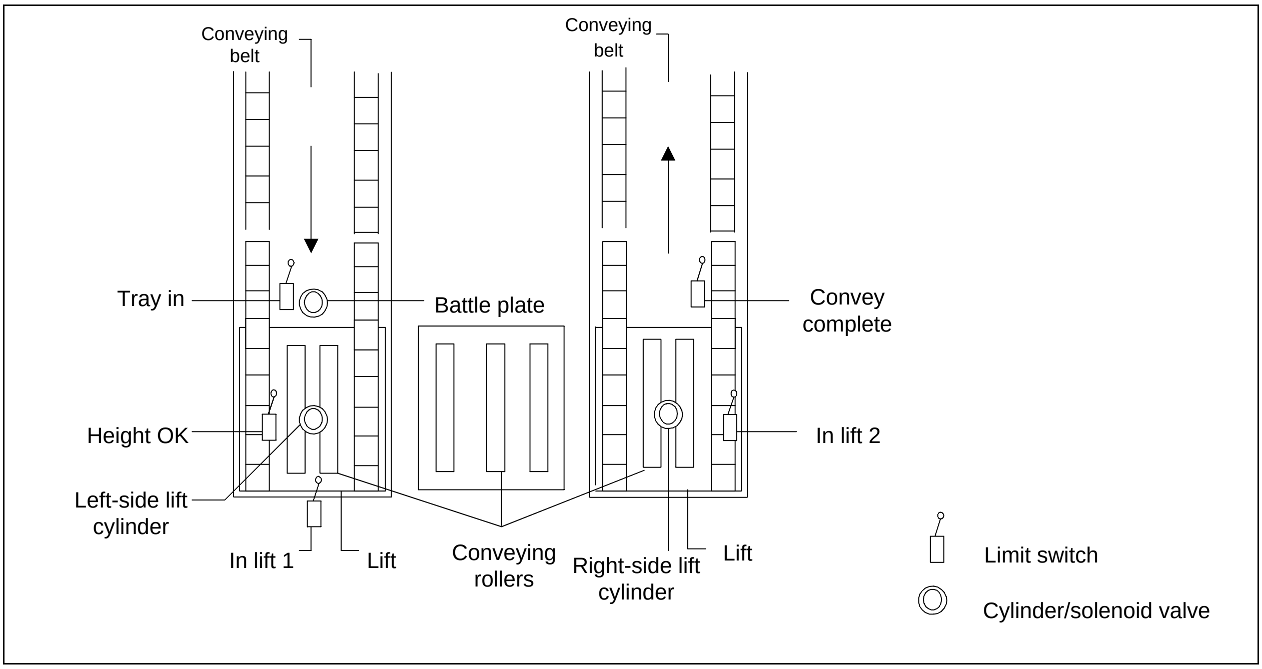
The values of S elements can be kept by the function for saving data at power outage. In this way, after the power on, the program can rerun from the step state when the power outage occurred.

7.5 Examples of SFC programming

The examples in this section are just illustrations of SFC programming, with simplified operations and conditions. The device configuration is conceptual and for study only. You cannot apply the example programs to actual use.

7.5.1 Simple sequential structure

The following example is a workpiece tray lifting and conveying machine. This machine uses cylinder lifting devices and conveying rollers to convey the workpiece tray from one conveying belt to another. The following figure shows a top view of the machine.



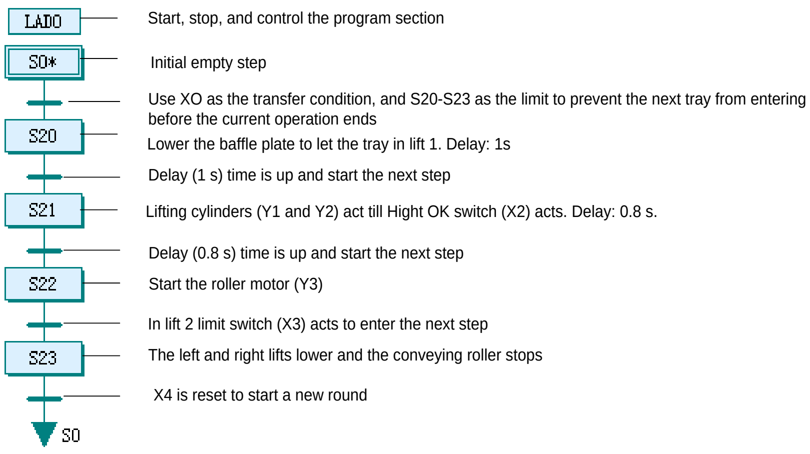
After the machine is started, the workpiece tray is conveyed along the convey or belt at the left sideto the entrance of the machine and triggers the "Tray in" limit switch. If no other tray is occupying the machine, the "Baffle plate" lowers down to let the work piecetray enter the machine. When the tray has completely entered the left lift, it triggers the "In lift 1" limit switch, the lift raises the tray until the "Height OK" limit switch is triggered. The rollers then act to convey the tray to the lift on the rightside until the "In lift 2" limit switch is triggered. The lift then lowers to put the tray to the conveying belt on the right.When the "Convey complete" limit switch is reset, a complete lifting and conveying process is over and the machine is ready for the next round.

The I/O points are listed in the following table.

SN	Address	Monitored objects		SN	Address	Monitored objects
1	X0	Tray in limit switch		8	Y0	Cylinder solenoid valve for the baffle plate
2	X1	In lift 1 limit switch		9	Y1	Cylinder solenoid valve for the left lift
3	X2	Height OK limit switch		10	Y2	Cylinder solenoid valve for the right lift
4	X3	In lift 2 limit switch		11	Y3	Conveying roller motor contactor
5	X4	Convey complete switch		12	Y4	Motor contactor for the left conveying belt
6	X5	Start switch		13	Y5	Motor contactor for the right conveying belt
7	X6	Auxiliary signal of emergency switch				

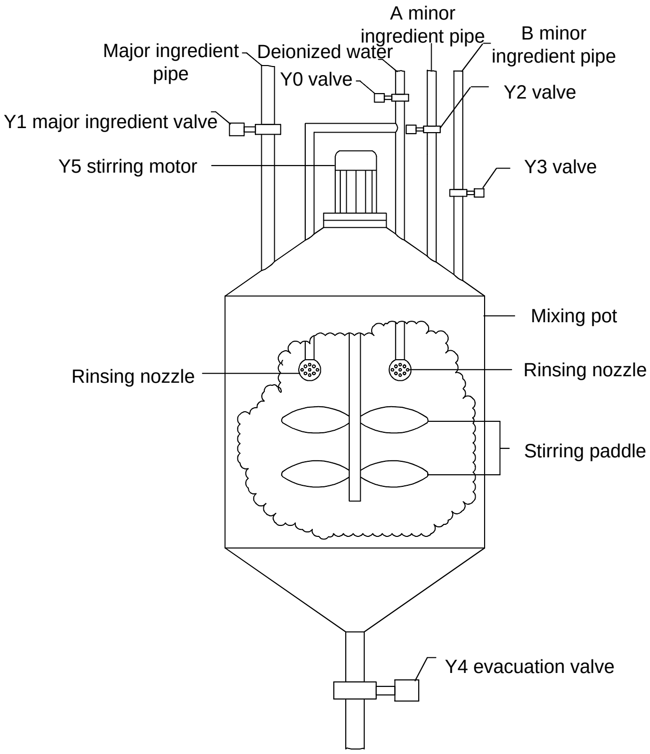
This is a simple sequential process. Each tray is conveyed in several successive procedures, with no other choices or parallel procedures, and no parallel flow. Writing the program with SFC would be faster and clearer than the conventional logic design method.

The following diagrams are the SFC program and its LAD counterpart.



7.5.2 Selection branch structure

The following example is a material mixing process flow. Through this flow, two kinds of products, namely A and B, are produced. The following figure shows the illustration of the manufacturing device.



To start the operation, the first step is to select the product type, A or B through the touch screen for the next batch of products, and then start production. The second step is to add the major ingredients until the weight of the added ingredient reaches 2000kg. The third step is to add minor ingredients, namely adding A minor ingredients for type A product or B minor ingredients for type B product until the added A or B minor ingredient reaches 500kg respectively. The forth step is to mix the ingredients for 20 minutes. The fifth step is to evacuate the materials until the remaining material is less than 20kg and the delay time is up. Once these steps are complete, the machine is ready for the next round.

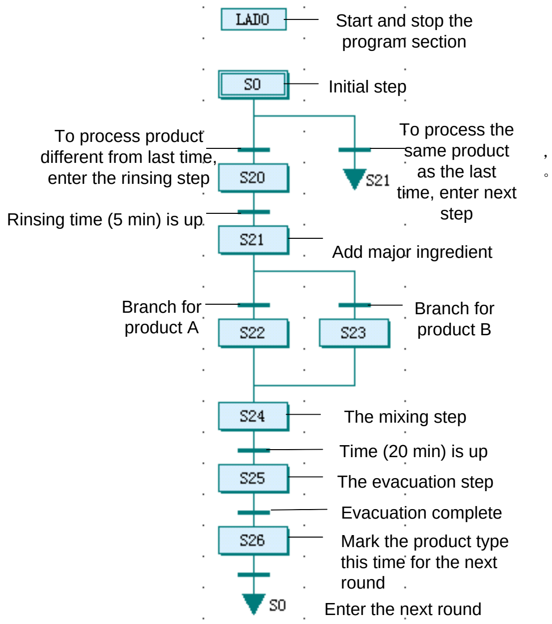
If it is the first time for the machine to start production, or the product type of the previous batch is different from what is going to be produced, you need to open the deionized water valve and evacuation valve to rinse the machine for 5 minutes before adding the major ingredients.

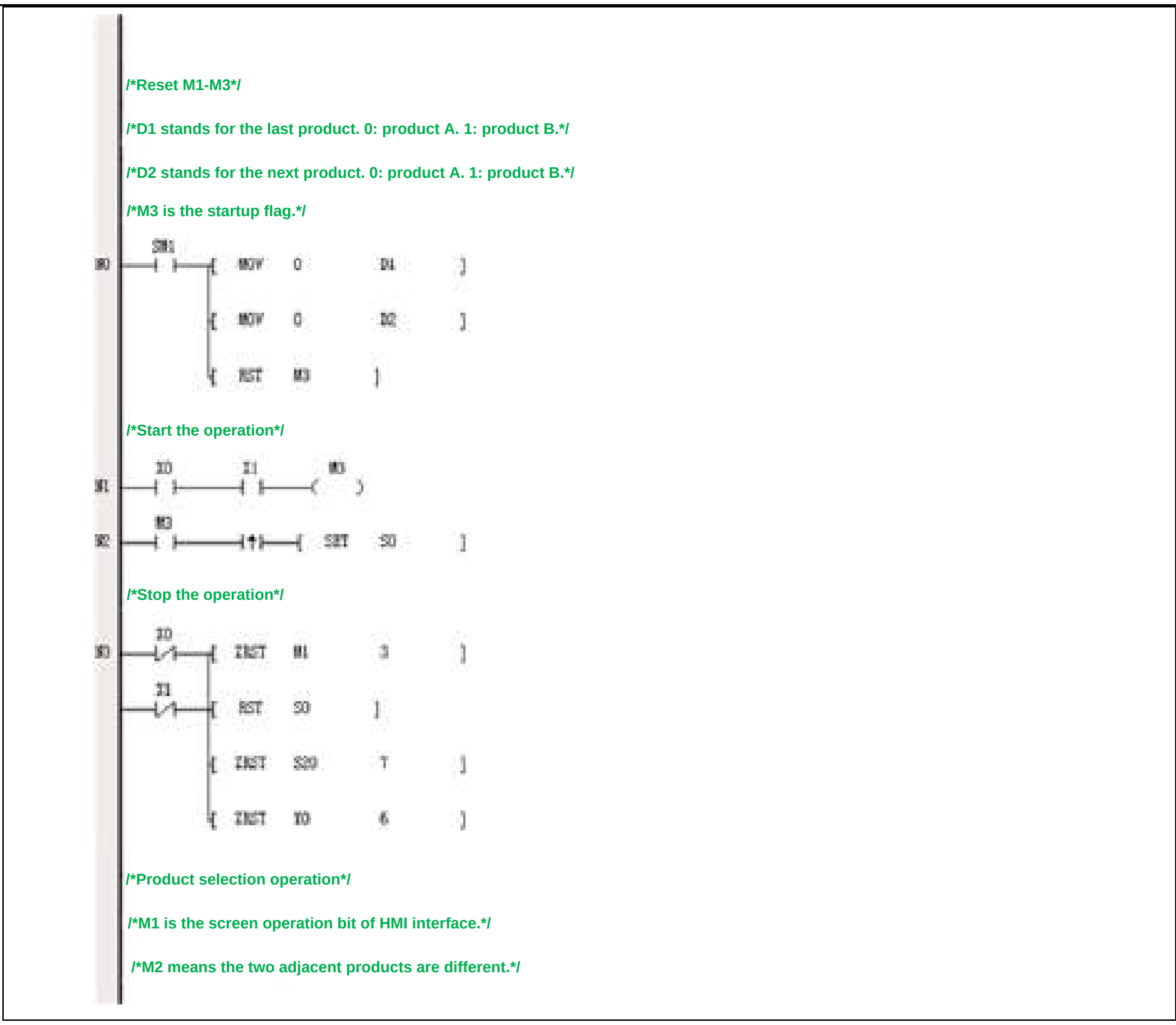
The I/O points are listed in the following table.

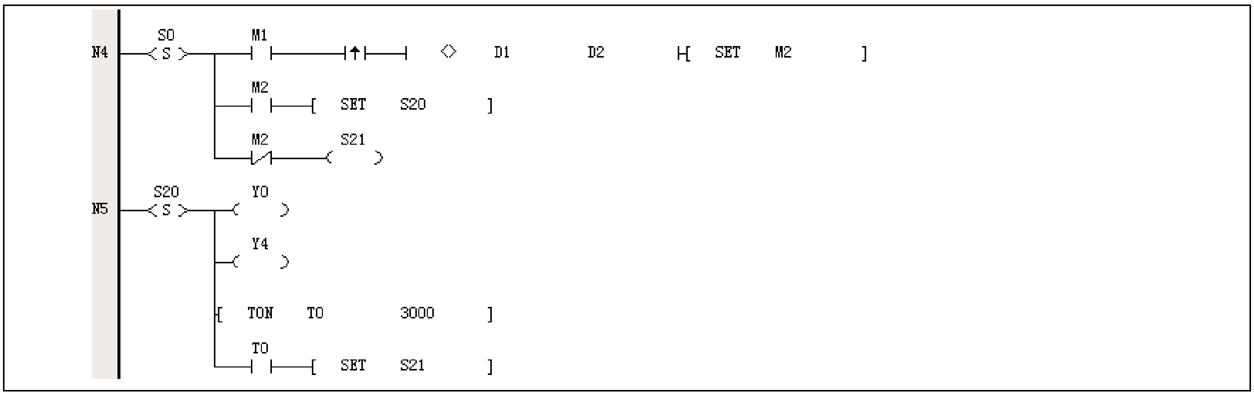
SN	Address	Monitored objects		SN	Address	Monitored objects
1	X0	Deionized water valve open		10	X11	Evacuation valve open
2	X1	Deionized water valve closed		11	X12	Evacuation valve closed
3	X2	Major ingredient valve open		12	Y0	Solenoid valve for deionized water
4	X3	Major ingredient valve closed		13	Y1	Solenoid valve for major ingredient
5	X4	Minor ingredient A valve open		14	Y2	Solenoid valve for minor ingredient A
6	X5	Minor ingredient A valve closed		15	Y3	Solenoid valve for minor ingredient B
7	X6	Minor ingredient B valve open		16	Y4	Solenoid valve for evacuation
8	X7	Minor ingredient B valve closed		17	Y5	Mixing motor contactor
9	X10	Mixing motor running				

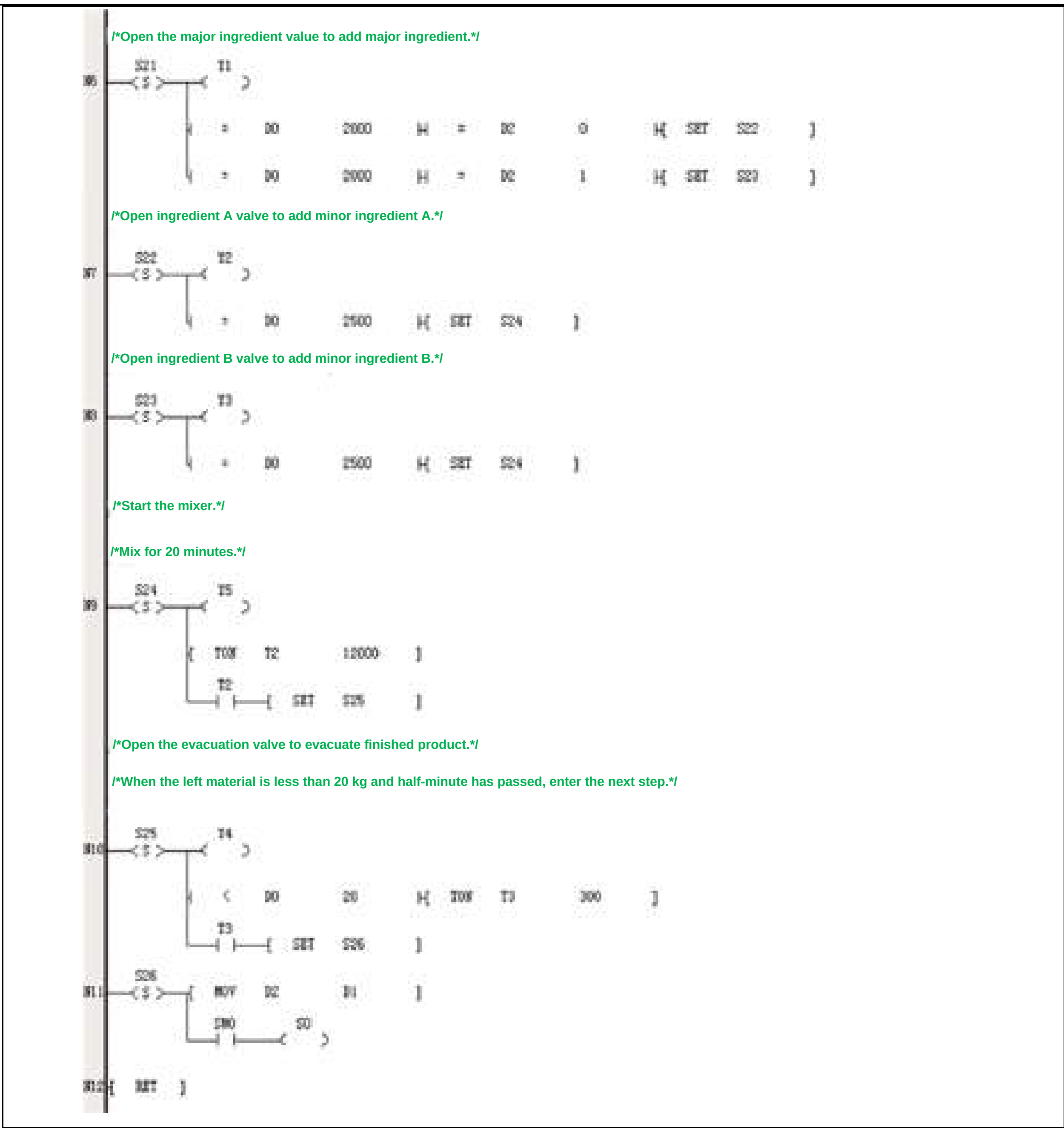
Obviously this is a selection branch structured flow. You can select only one type of product, A or B, in a round of product production. You can switch the type of product only when a round of production is completed.Meanwhile, there is a structure for selection and jump in the process, namely the rinsing step.

The following diagrams are the SFC program and its LAD counter part.



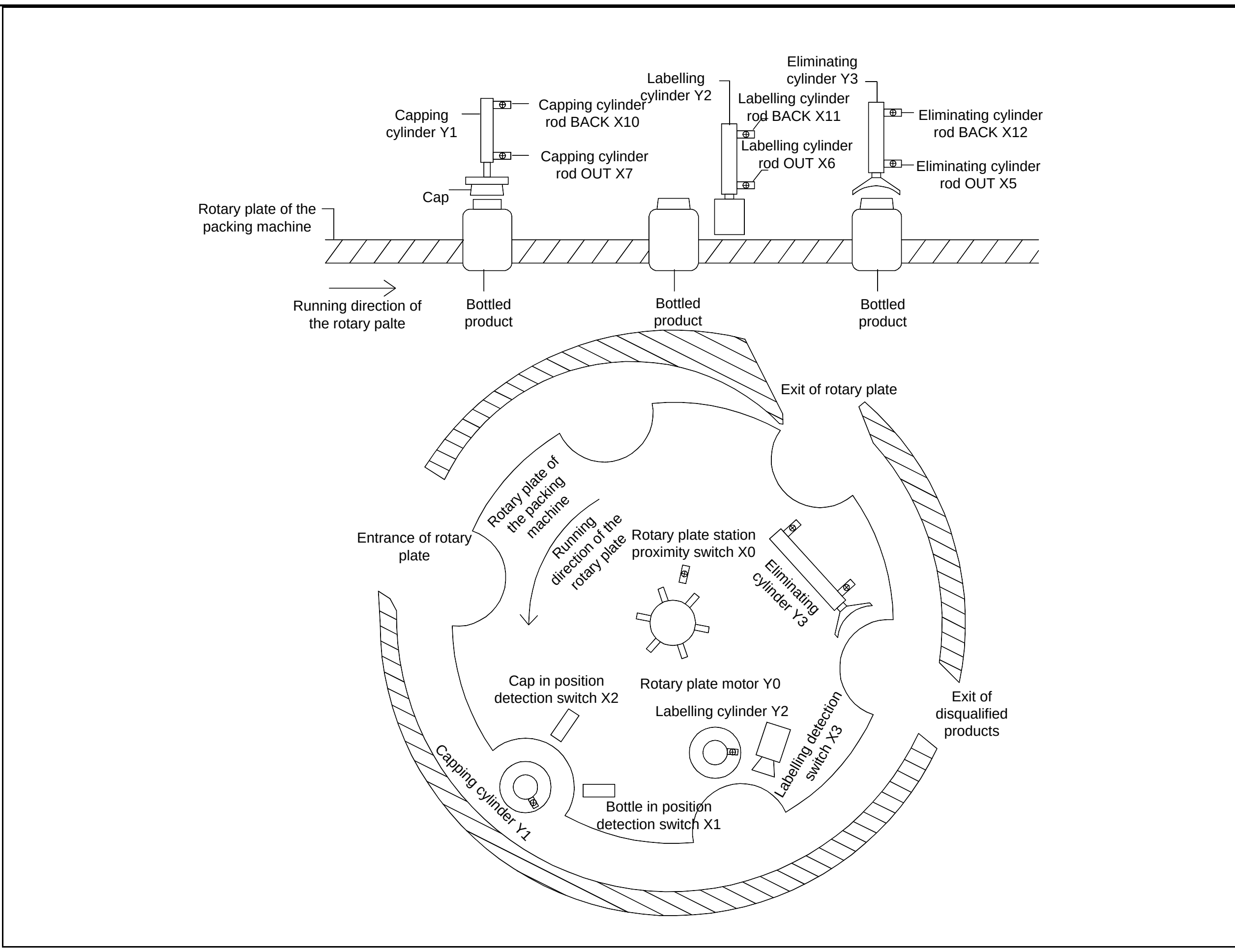

/*Reset M1-M3*/
/*D1 stands for the last product. 0: product A. 1: product B.*/
/*D2 stands for the next product. 0: product A. 1: product B.*/
/*M3 is the startup flag.*/
MOV 0 D1
MOV 0 D2
RST M3
/*Start the operation*/
X0
X1
M3
SET S0
/*Stop the operation*/
X0
X1
ZRST
RST S0
ZRST
ZRST
/*Product selection operation*/
/*M1 is the screen operation bit of HMI interface.*/
/*M2 means the two adjacent products are different.*/


N4
S0
M1
D1
D2
SET M2
M2
SET S20
M2
S21
N5
S20
Y0
Y4
TON T0 3000
T0
SET S21



7.5.3 Parallel branch structure

The following example is a bottle packager. This packager seals the bottles and sticks labels to them. Meanwhile, it examines the bottle caps and labels, so that the flawed products are eliminated in the third procedure, while the qualified products continue to the next work flow.If no bottle is sent from the last work flow, the packager does not conduct any sealing or labeling. These three procedures are carried out at the same time, and each bottle moves from one position to another each time the rotary plate rotates.The following figure shows the illustration of the packager.



During the operation, the rotary plate rotates one step each time, which is detected by the X0 limit switch. The rotary plate stays at each step long enough for all the three procedures, driven by cylinders, finished. The cylinder rod OUT signal and cylinder rod BACK signal are monitored respectively.

The I/O points are listed in the following table.

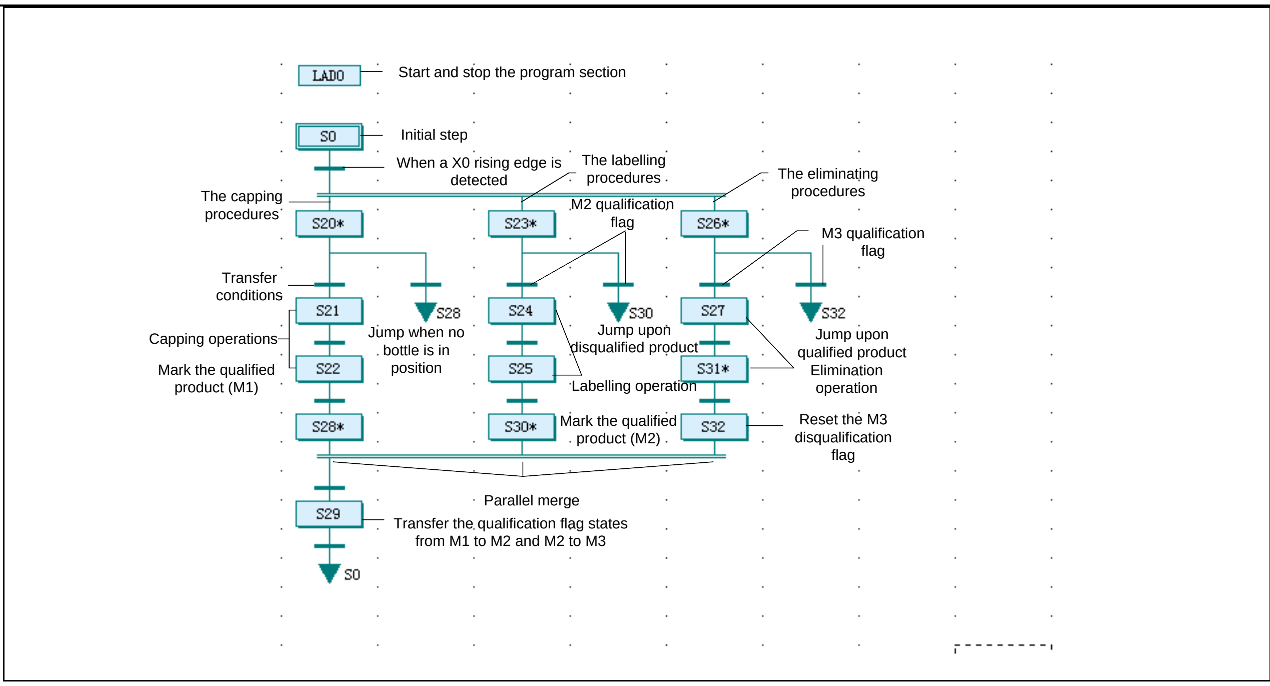
SN	Address	Monitored objects	SN	Address	Monitored objects
1	X0	Rotary plate station proximity switch	8	X10	Capping cylinder rod BACK
2	X1	Bottle in position detection switch	9	X11	Labelling cylinder rod BACK
3	X2	Cap in position detection switch	10	X12	Eliminating cylinder rod BACK
4	X3	Labelling detection switch	11	Y0	Rotary plate motor
5	X5	Eliminating cylinder rod OUT	12	Y1	Capping cylinder
6	X6	Labelling cylinder rod OUT	13	Y2	Labelling cylinder
7	X7	Capping cylinder rod OUT	14	Y3	Eliminating cylinder

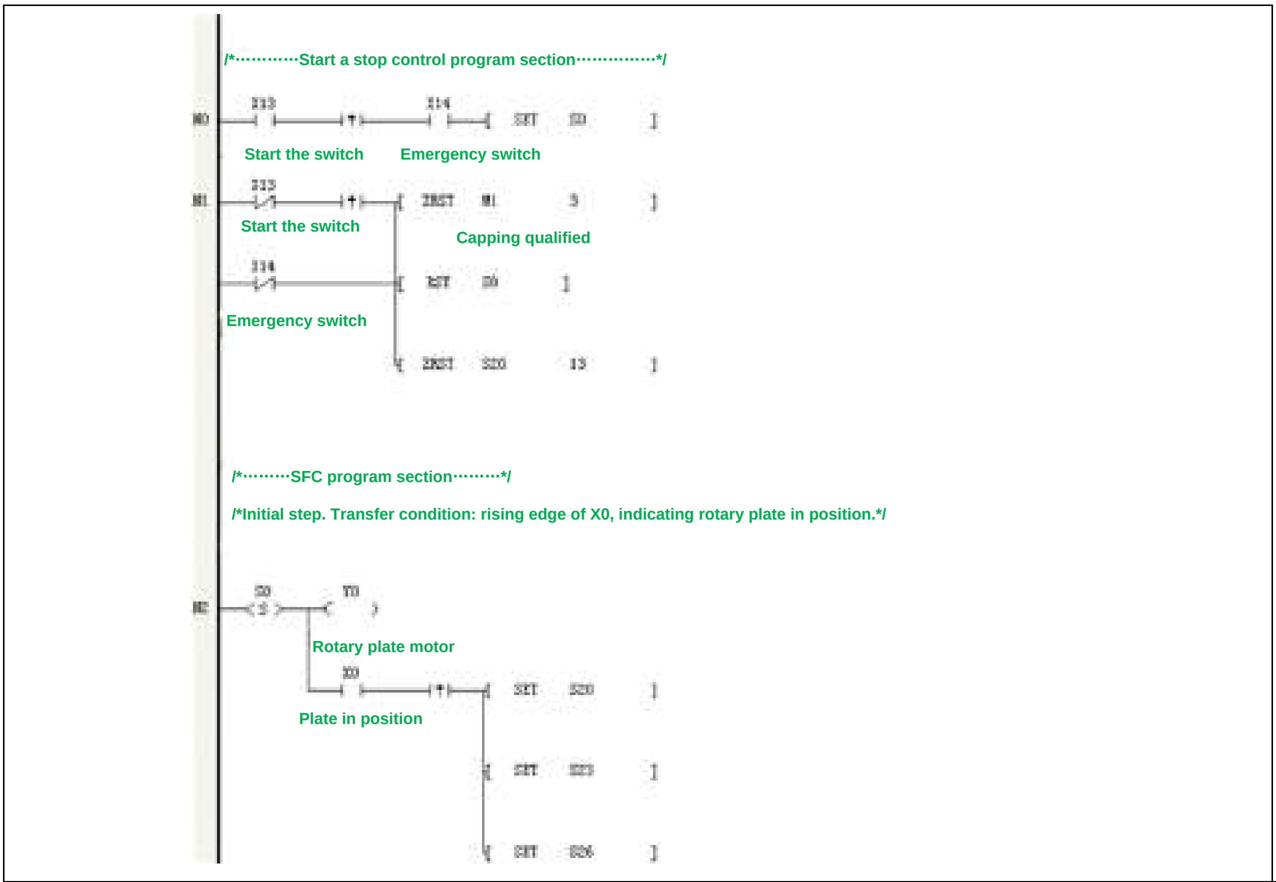
It is obvious that this is a parallel branch structured flow. After the rotating of the rotary plate, operations are performed in three stations. After all the operations are performed in the three stations, the device performs the subsequent operations.The following diagrams are the SFC program and its LAD counter part.

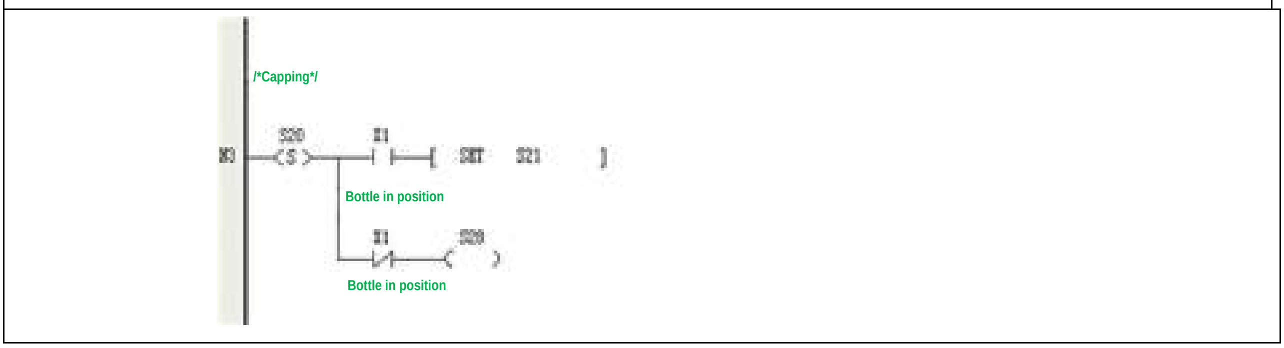
In the program, M1-M3 are the qualification flags for the procedures of capping, labeling and eliminating respectively.

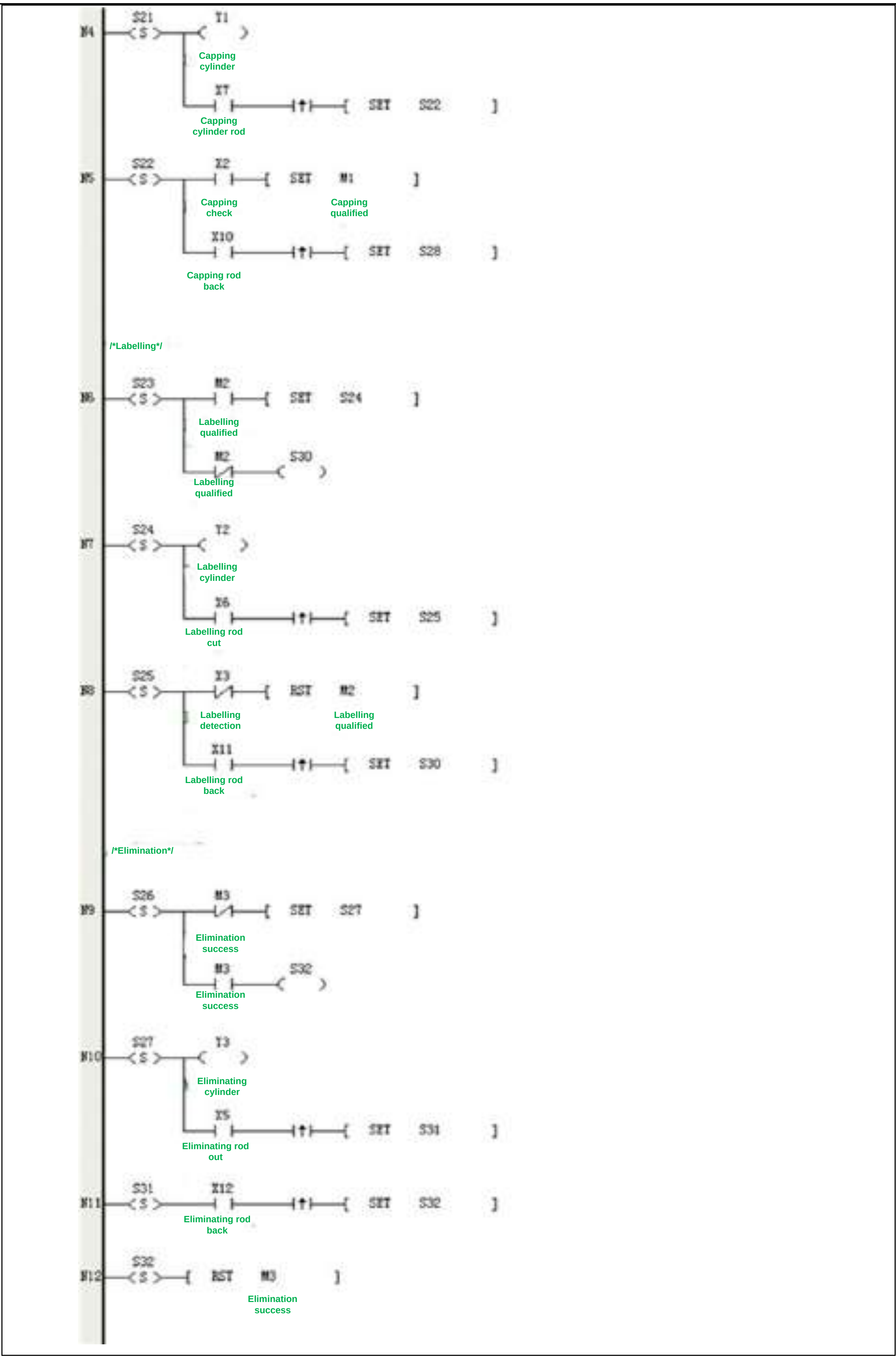
When the capping procedure runs to S22, X2 checks whether the capping is qualified or not. If yes, the corresponding qualification flag M1 is set. When the labeling procedure runs to S25, X3 checks whether the labeling is qualified or not. If not, M2 is reset. After all the procedures are complete, at step S29, the M2 state is transferred to M3, and M1 state is transferred to M2.

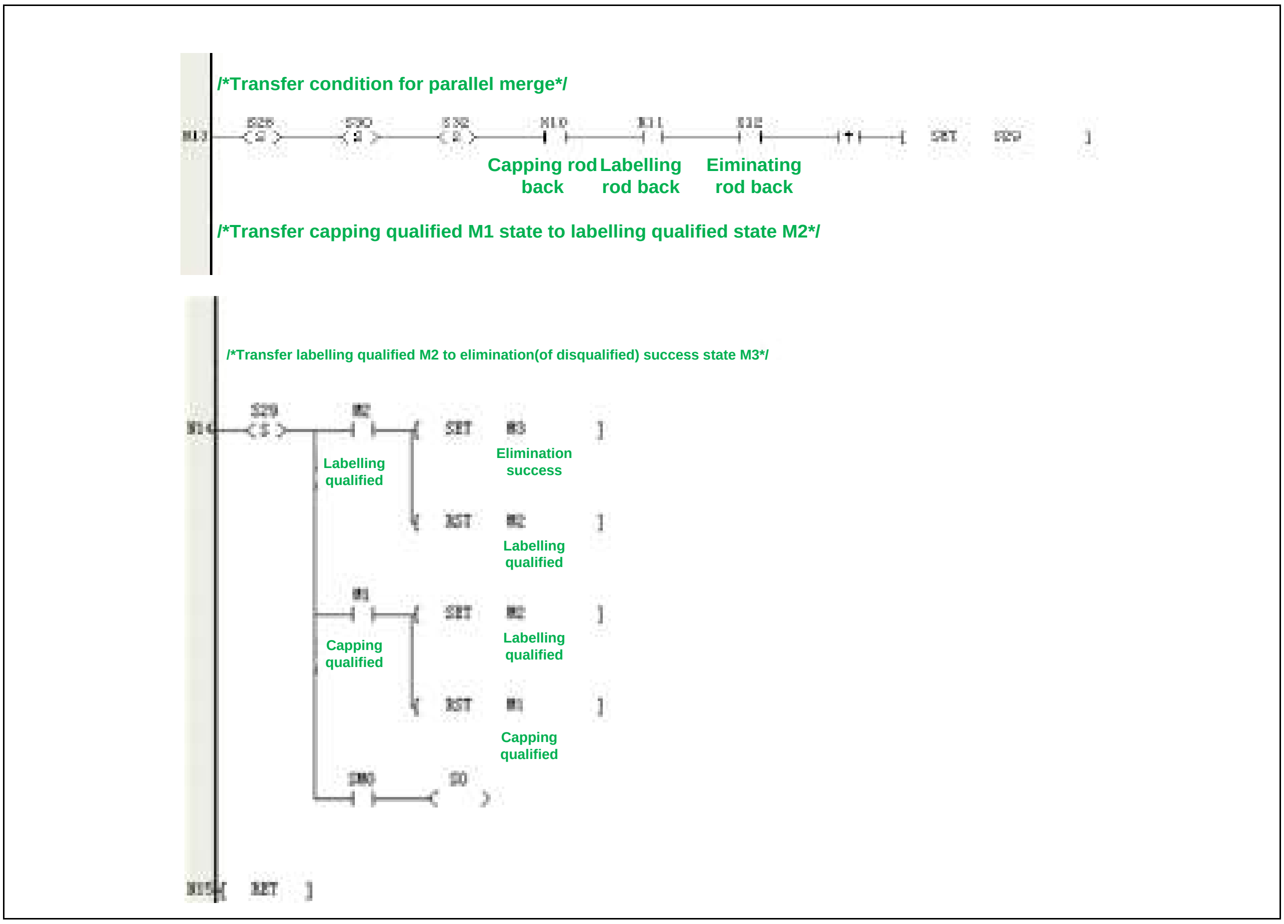
The capping procedure acts according to X1 state. If X1 indicates that no bottle is in position, the capping does not proceed. The labeling procedure acts according to M2 state. If M2 is OFF, it indicates that the bottle in position is disqualified, and the labeling does not proceed. The eliminating procedure acts according to M3. If M3 is ON, it indicates that the bottle is qualified, and the elimination does not proceed, and vice versa. In both cases, M3 is reset in S32 to prepare for the next procedure.


LAD0
Start and stop the program section
S0
Initial step
When a X0 rising edge is detected
The labelling procedures
The eliminating procedures
The capping procedures
S20*
S23*
M2 qualification flag
S26*
M3 qualification flag
Transfer conditions
S21
S28
S24
S30
S27
S32
Capping operations
Jump when no bottle is in position
Jump upon disqualified product
Jump upon qualified product
Elimination operation
S22
S25
S31*
Labelling operation
Mark the qualified product (M1)
S28*
S30*
Mark the qualified product (M2)
S32
Reset the M3 disqualification flag
Parallel merge
S29
Transfer the qualification flag states from M1 to M2 and M2 to M3
S0


/*…………Start a stop control program section……………*/
Start the switch
Emergency switch
Start the switch
Capping qualified
Emergency switch
/*………SFC program section………*/
/*Initial step. Transfer condition: rising edge of X0, indicating rotary plate in position.*/
Rotary plate motor
Plate in position


/*Capping*/
Bottle in position
Bottle in position


M4
S21
S
T1
Capping cylinder
X7
SET S22
Capping cylinder rod
M5
S22
X2
SET M1
Capping check
Capping qualified
X10
SET S28
Capping rod back
/*Labelling*/
M6
S23
M2
SET S24
Labelling qualified
M2
S30
Labelling qualified
M7
S24
T2
Labelling cylinder
X6
SET S25
Labelling rod cut
M8
S25
X3
RST M2
Labelling detection
Labelling qualified
X11
SET S30
Labelling rod back
/*Elimination*/
M9
S26
M3
SET S27
Elimination success
M3
S32
Elimination success
M10
S27
T3
Eliminating cylinder
X5
SET S31
Eliminating rod out
M11
S31
X12
SET S32
Eliminating rod back
M12
S32
RST M3
Elimination success


/*Transfer condition for parallel merge*/
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*/
Labelling qualified
Elimination success
Labelling qualified
Capping qualified
Labelling qualified
Capping qualified

Chapter 8 Operating guide for high-speed input function

This chapter detailedly describes the usages and notes about the high-speedinput function, including high-speed counters and pulse capture.

8.1 High-speed counter

8.1.1 Configuration

The built-in high-speed counter for IVC series micro-PLCs are configured as follows:

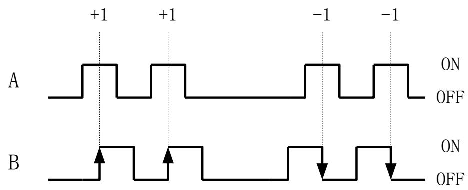
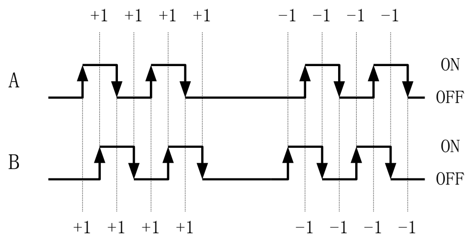
- Configuration table of high-speed counter

Counter \ Input point		X0	X1	X2	X3	X4	X5	X6	X7	Max. frequency (kHz)				
										IVC3	IVC2L	IVC1	IVC1L	IVC1S
Single-phase one point input mode	C236	Up/Down												
	C237		Up/Down									50		
	C238			Up/Down										
	C239				Up/Down									
	C240					Up/Down								
	C241						Up/Down					10		10
	C301							Up/Down						
	C302								Up/Down			/		/
	C242	Up/Down		Reset								50		
	C243				Up/Down		Reset					10		
	C244	Up/Down		Reset				Start				50		
	C245				Up/Down		Reset		Start	200		10		10
Single-phase bidirectional count input	C246	Up	Down									50		
	C247	Up	Down	Reset								50		10
	C303				Up	Down						/		/
	C248				Up	Down	Reset					10		
	C249	Up	Down	Reset				Start				50		
	C250				Up	Down	Reset		Start	200		10		10
Two-phase count input mode	C251	A phase	B phase									30		5
	C304			A phase	B phase									
	C305					A phase	B phase							
	C306							A phase	B phase			/		/
	C252	A phase	B phase	Reset								30		
	C253				A phase	B phase	Reset					5		
	C254	A phase	B phase	Reset				Start				30		
	C255				A phase	B phase	Reset		Start	200		5		5
SPD instruction		Input point	Input point	Input point	Input point	Input point	Input point	Input point	Input point	200		10		10
Pulse capture function		Input point	Input point	Input point	Input point	Input point	Input point	Input point	Input point			10		

External interrupt number	0/10	1/11	2/12	3/13	4/14	5/15	6/16	7/17	10

In the modes listed in the preceding table, the high-speed counters act according to certain input and handle high-speed action according to interrupts. The counting action is unrelated to the PLC scan cycle.

This kind of high-speed counters are of the 32-bit bi-directional counting type. According to their different counting up/down switch over methods, they can be divided into the following four categories:

Counting mode	Counting action
Single-phase one port count input	Counters C236–C245 and C301–C302 count down/up when SM236–SM245 and SM301–SM302 are ON/OFF.
Single-phase bidirectional count input	Corresponding to the action of counting up or down input, the counters C246–C250 and C303 are automatically incremented/decremented. You can know the current count direction of the corresponding counter through SM246–SM250 and SM303. The counter counts up when the SM element is OFF, or counts down when OFF.
Two-phase count input	When SM100–SM104 is set to OFF, counters C251–C255 and C304–C306 are automatically incremented and decremented according to the two-phase input. You can know the current count direction of the corresponding counter through SM251–SM255 and SM304–SM306. The counter counts up when the SM element is OFF, or counts down when OFF. The count direction is defined as follows: 
Two-phase quad frequency count input	When SM100–SM104 is set to ON, counters C251–C255 and C304–C306 are automatically quadrupled up and down according to the two-phase input. You can know the current count direction of the corresponding counter through SM251–SM255 and SM304–SM306. The counter counts up when the SM element is OFF, or counts down when OFF. The count direction is defined as follows: 

8.1.2 Relationship between high-speed counter and SM auxiliary relay

- Special auxiliary relay for controlling count direction

Type	Counter SN	Up/down control
Single-phase one point count input	C236	SM236
	C237	SM237
	C238	SM238
	C239	SM239
	C240	SM240
	C241	SM241
	C242	SM242
	C243	SM243
	C244	SM244
	C245	SM245
	C301	SM301
	C302	SM302

- Special auxiliary relay for monitoring count direction

Type	Counter SN	Up/down monitor
Single-phase bidirectional count input	C246	SM246
	C247	SM247
	C248	SM248
	C249	SM249

- Special auxiliary relay for controlling quad-frequency

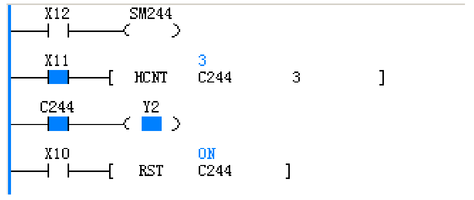
Type	Counter SN	Quad-frequency setting
Two-phase count input	C251	SM100
	C252	SM100
	C253	SM102
	C254	SM100
	C255	SM102
	C304	SM101
	C305	SM103
	C306	SM104

Type	Counter SN	Up/down monitor
	C250	SM250
	C303	SM303
Two-phase count input	C251	SM251
	C252	SM252
	C253	SM253
	C254	SM254
	C255	SM255
	C304	SM304
	C305	SM305
	C306	SM306

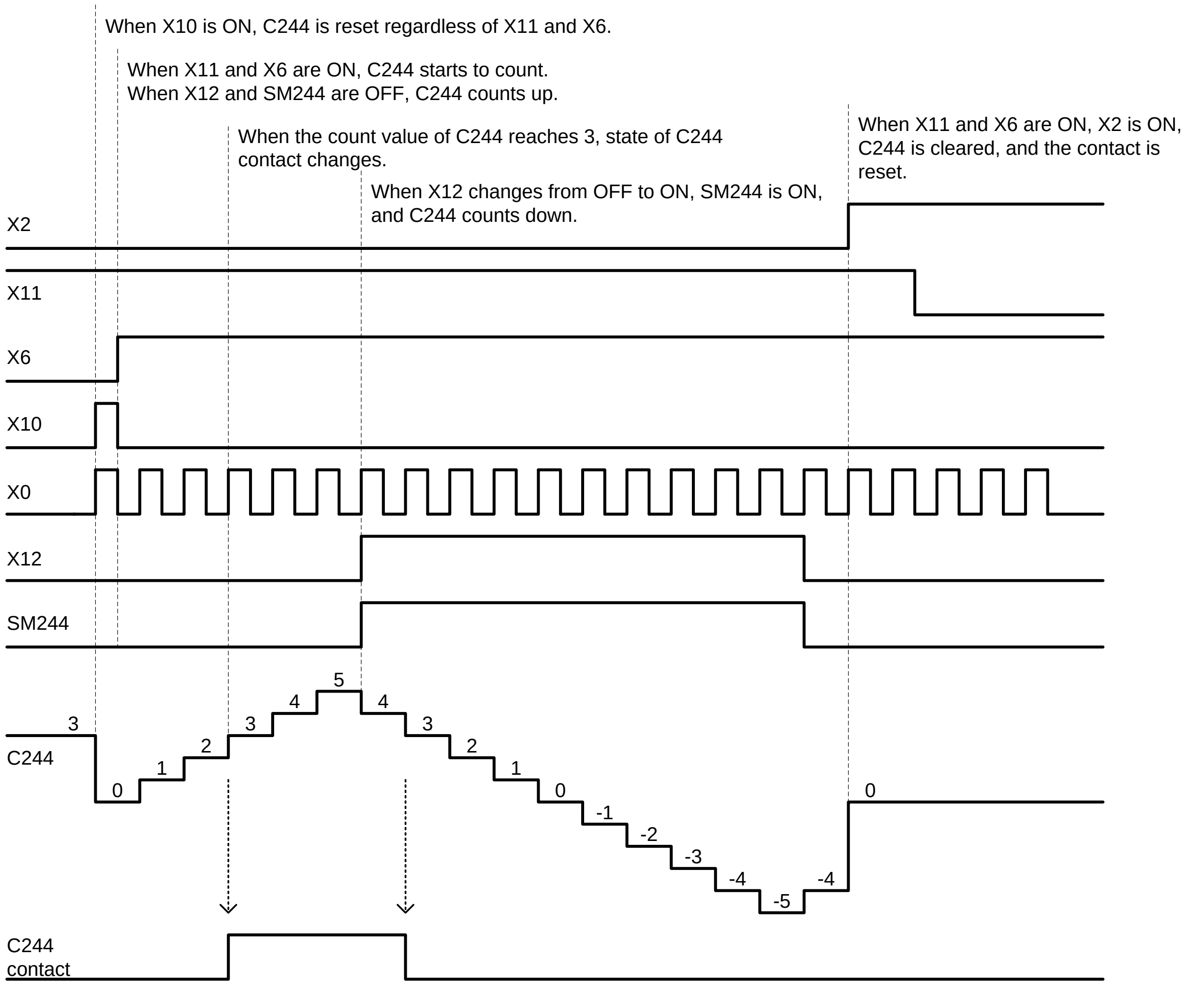
8.1.3 How to use the high-speed counters

- How to use the single-phase one point input high-speed counters

Characteristics: A single-phase one point input high-speed counter starts to count only when the pulse input changes from OFF to ON, and the count direction is determined by its corresponding special auxiliary relay SM. An example of the action is shown in the following figure.



The time sequence chart of the contact actions in the program is shown in the following figure.

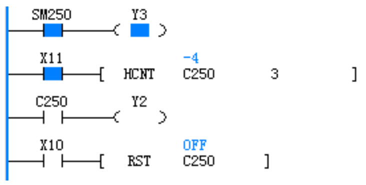


Note:
1. Counter input points: X0.
2. High-Speed counters, when used in DHSCS, DHSCR, DHSZ, DHSP, and DHST instructions, can trigger operations or controls free from the scan cycle.

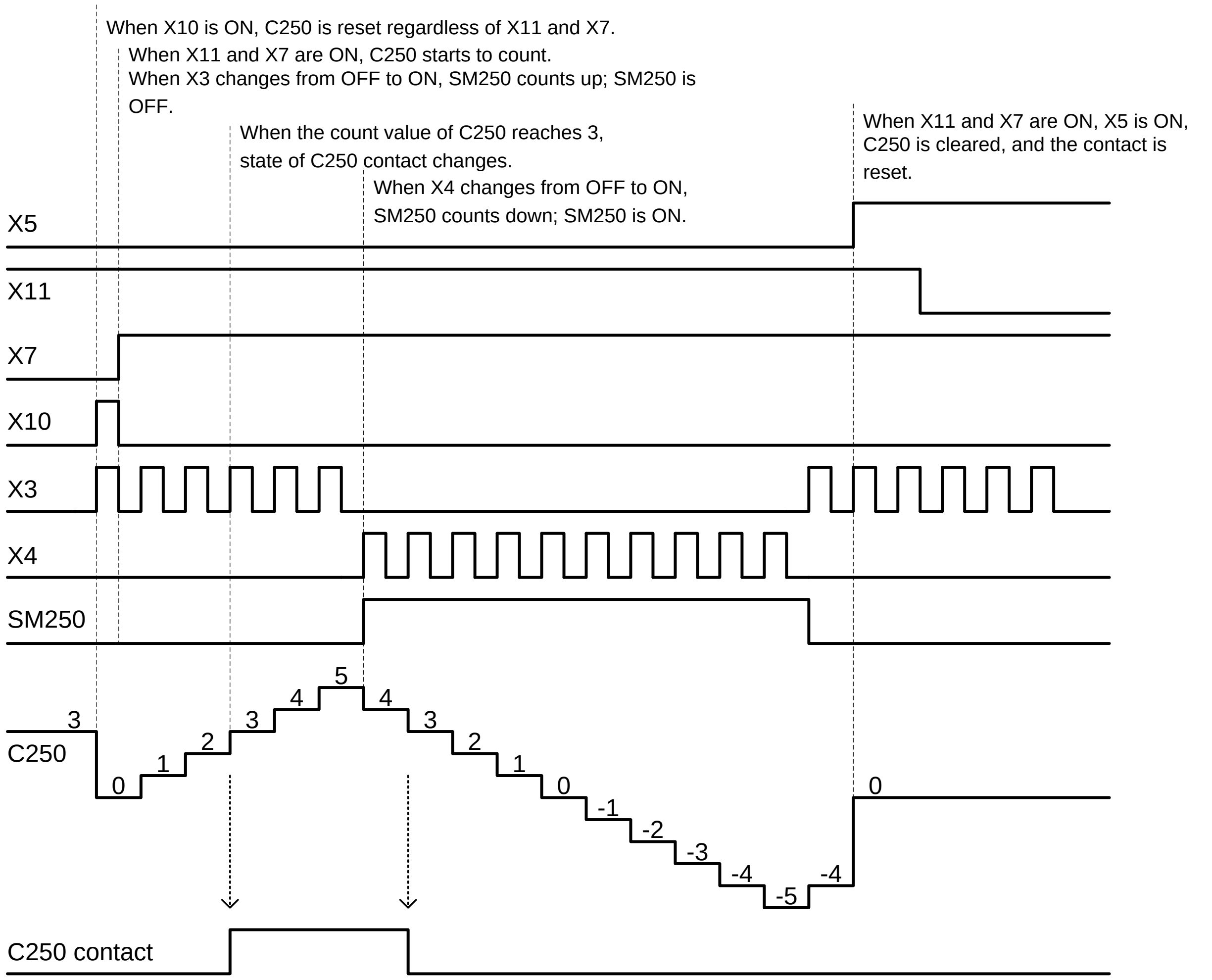
- How to use the single-phase bidirectional count inputhigh-speed counters

Characteristics: A single-phase bidirectional count input high-speed counter starts to count only when the pulse input changes from OFF to ON, and the count direction is determined by their corresponding two input points. The state of the high-speed counter is determined by its corresponding special auxiliary relay SM.

An example of the action is shown in the following figure.



The time sequence chart of the contact actions in the program is shown in the following figure.



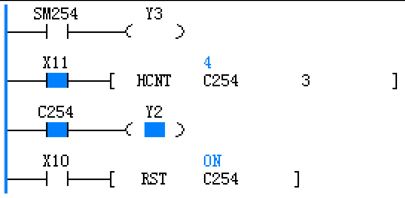
Note:

1. Counter input points: X3 and X4.
2. High-Speed counters, when used in DHSCS, DHSCR, DHSZ, DHSP, and DHST instructions, can trigger operations or controls free from the scan cycle.

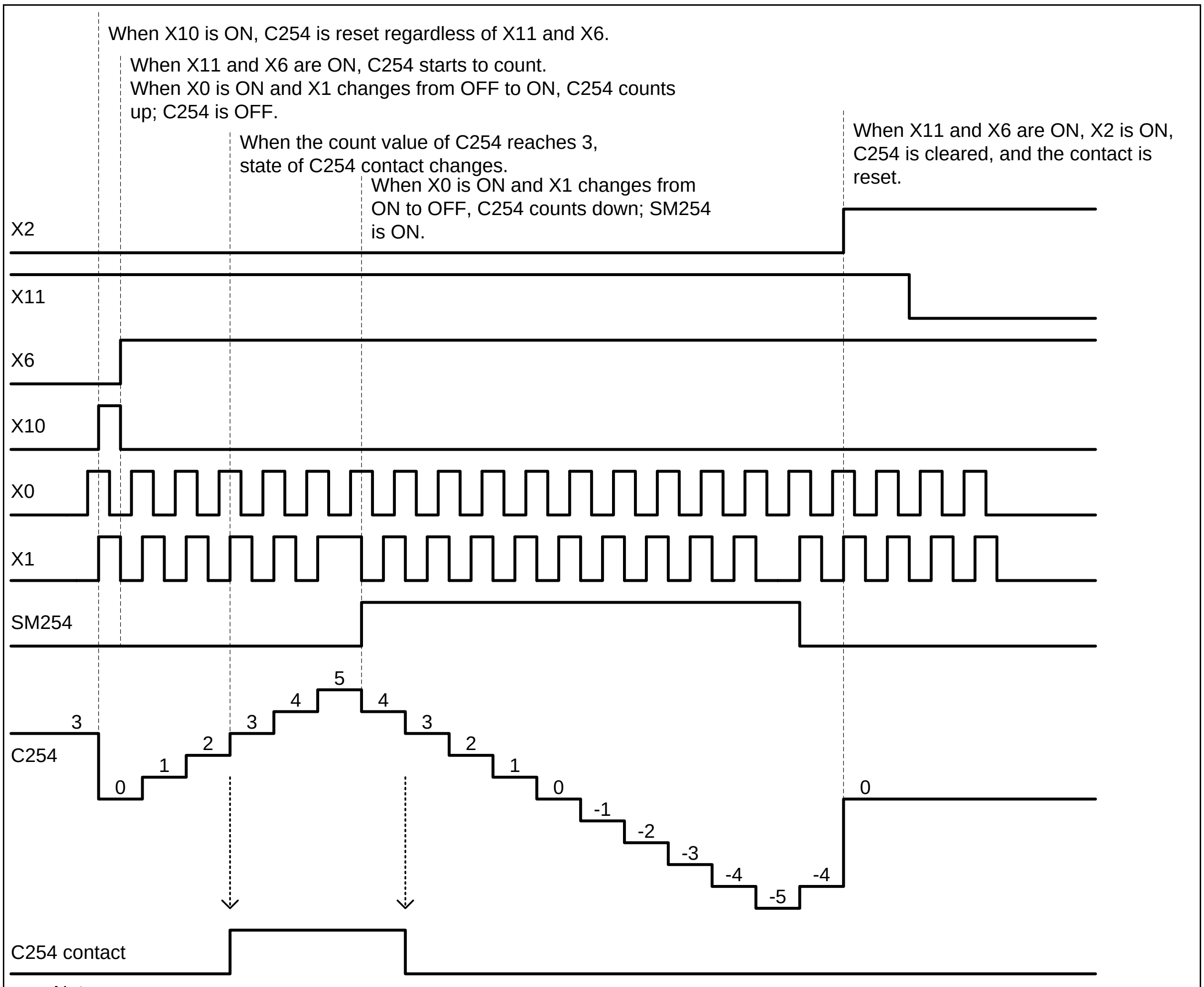
- How to use the two-phase count input high-speed counters

Characteristics: A two-phase count input high-speed counter starts to count only when the pulse input changes from OFF to ON, and the count direction is determined by the phase difference between the two input points. The state of the high-speed counter is determined by its corresponding special auxiliary relay SM.

An example of the action is shown in the following figure.



The time sequence chart of the contact actions in the program is shown in the following figure.

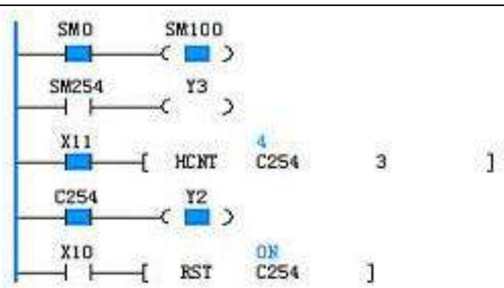


Note:
1. Counter input points: X0 and X1.
2. High-Speed counters, when used in DHSCS, DHSCR, DHSZ, DHSP, and DHST instructions, can trigger operations or controls free from the scan cycle.

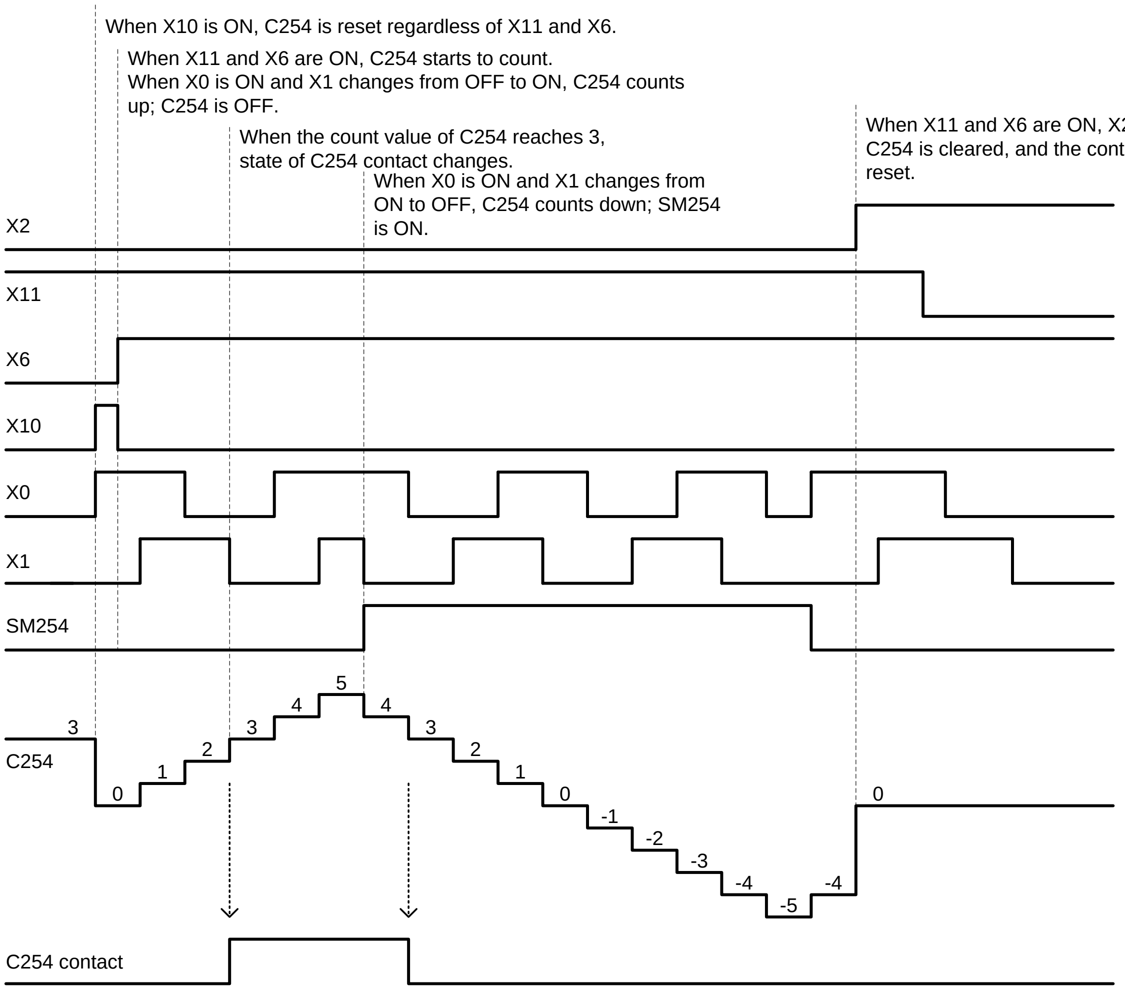
- How to use the two-phase quad frequency count input high-speed counters

Characteristics: A two-phase quad frequency count input high-speed counter starts to count when the pulse input changes from OFF to ON or ON to OFF, and the count direction is determined by the phase difference between the two input points. The state of the high-speed counter is determined by its corresponding special auxiliary relay SM.

An example of the action is shown in the following figure.



The time sequence chart of the contact actions in the program is shown in the following figure.



Note:
1. Counter input points: X0 and X1.
2. High-Speed counters, when used in DHSCS, DHSCR, DHSZ, DHSP, and DHST instructions, can trigger operations or controls free from the scan cycle.

8.1.4 Notes about the IVC2L and IVC1 series high-speed counters

- Classification of high-speed counters

C236, C237, C246 and C251 can be used as both hardware counters and software counters, which depends on the modes in which they are used. All the other high-speed counters are software counters in any case.

- Max. frequency sum

The max. combined frequency cannot exceed 80kHz when multiple high-speed counters (hardware counting mode) are used simultaneously or the high-speed counters (hardware counting mode) and the SPD instruction are used at the same time.

Total input frequency when multiple software high-speed counters, or when high-speed counters and the SPD instruction, are used at the same time, is shown in the following table.

conditions of usage	Total input frequency
DHSCS, DHSCR, DHSCI, DHSZ, DHSP, and DHSTare not used	≤80 kHz
DHSCS, DHSCR, DHSCI, DHSP, DHST are used	≤30 kHz
DHSZ is used	≤20 kHz

- Max. frequency of hardware counters

Counters C236, C237, C246 and C251 are the only four potential hardware counters. Among which:

- C236, C237 and C246 are single-phase counters with max. counting frequency of 50kHz.

- C251 is a two-phase counterwith max. counting frequency of 30kHz.

- **Max.**frequency **of software counters**

  The high speed counters used in DHSCS, DHSCR, DHSCI, DHSP or DHST instructions works in the software counter mode. The max. input frequency for the single-phase counters and two-phase counters are 10kHz and 5kHz respectively.

  When used in the DHSZ instruction, the max. input frequency for the single-phase counters and two-phase counters are 5 kHz and 4 kHz respectively.

8.2 External pulse capture function

The input points that provide the external pulse capture function are X0–X7. The corresponding SM soft elements arelisted below:

Input hardware point	SM element
X0	SM90
X1	SM91
X2	SM92
X3	SM93
X4	SM94
X5	SM95
X6	SM96
X7	SM97

Note

1. When the external input points change from OFF to ON, the SM soft elements of the corresponding portsare set to ON.
2. SM90–SM97 are cleared when the user programstarts.
3. When using the pulse capture, the counters need to observe the sum limit of the input pulse frequency of each PLC series, otherwise an abnormality may occur.
4. When the high-speed counter or SPD instruction corresponding to HCNT are used on the same input point, the pulse capture becomes invalid after the first scan cycle, regardless of whether the instructionis valid or not.

8.3 Notes on High-speed input application

The input points X0–X7 are used as input signals in functions such as high-speed counter, SPD instruction, pulse capture, and external interrupt. However, these functions cannot be used at the same time, because many different functions may use the same one or multiple input points. Therefore, during the PLC programming, only one of the several functions that aninput point can provide is available. If the X0–X7 input pointsare used repeatedly in the user program, the user program cannot be compiled.

Among the functions of high-speed counter, SPD instruction, pulse capture and external interrupt, the functions that input points X0–X7 can provide respectively are listed in the following table.

<table>
<tr><th colspan="2" rowspan="2">Input point / Counter</th><th rowspan="2">X0</th><th rowspan="2">X1</th><th rowspan="2">X2</th><th rowspan="2">X3</th><th rowspan="2">X4</th><th rowspan="2">X5</th><th rowspan="2">X6</th><th rowspan="2">X7</th><th colspan="4">Max. frequency (kHz)</th></tr>
<tr><th>IVC3</th><th>IVC2L</th><th>IVC1</th><th>IVC1L</th></tr>
<tr><td rowspan="7">Single-phase one point input mode</td><td>C236</td><td>Up/down</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td rowspan="7">200</td><td colspan="3" rowspan="2">50</td></tr>
<tr><td>C237</td><td></td><td>Up/down</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>C238</td><td></td><td></td><td>Up/down</td><td></td><td></td><td></td><td></td><td></td><td colspan="3" rowspan="4">10</td></tr>
<tr><td>C239</td><td></td><td></td><td></td><td>Up/down</td><td></td><td></td><td></td><td></td></tr>
<tr><td>C240</td><td></td><td></td><td></td><td></td><td>Up/down</td><td></td><td></td><td></td></tr>
<tr><td>C241</td><td></td><td></td><td></td><td></td><td></td><td>Up/down</td><td></td><td></td></tr>
<tr><td>C301</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Up/down</td><td></td><td colspan="3">/</td></tr>
</table>

	C302								Up/down		
	C242	Up/down		Reset							10
	C243				Up/down		Reset				
	C244	Up/down		Reset				Start			
	C245				Up/down		Reset		Start		
Single-phase bidirectional count input	C246	Up	Down							200	50
	C247	Up	Down	Reset							10
	C303				Up	Down					/
	C248				Up	Down	Reset				10
	C249	Up	Down	Reset				Start			
	C250				Up	Down	Reset		Start		
Two-phase count input mode	C251	A phase	B phase							200	30
	C304			A phase	B phase						/
	C305					A phase	B phase				
	C306							A phase	B phase		
	C252	A phase	B phase	Reset							5
	C253				A phase	B phase	Reset				
	C254	A phase	B phase	Reset				Start			
	C255				A phase	B phase	Reset		Start		
SPD instruction		Input point	Input point	Input point	Input point	Input point	Input point	Input point	Input point	200	10
Pulse capture function		Input point	Input point	Input point	Input point	Input point	Input point	Input point	Input point	/	/
External interrupt number (rising/falling edge)		0/10	1/11	2/12	3/13	4/14	5/15	6/16	7/17	/	/

Chapter 9 Using Interrupts

This chapter detailedly describes the mechanisms, processing procedures, and usages of various interrupts.

9.1 Interrupt program

When an interrupt event occurs, the normal scan cycle is interrupted and the interrupt program is called and executed with priority,which is called the interrupt processing mechanism. For the time-critical and event-triggered control tasks, you often need to adopt this special system processing mechanism.

The system provides many kinds of programmable interrupt resources. Each kind of interrupt resource can trigger a type of interrupt events, and each type of interrupt eventsis independently numbered.

In order to deal with a certain interrupt event, you must compile a processing program, that is, an interrupt program,

which is an independent POU that constitutes the user program.

An event number needs to be designated for each interrupt program in order to link the user interrupt program with the interrupt events designated with the event number. When responding to the interrupt request of the interrupt event, the system calls the corresponding user interrupt program based on the interrupt event number.

The following list shows the interrupt resources provided by the IVC series micro-PLCs:

Event number	Interrupt event	Enabling SM	Event number	Interrupt event	Interrupt enabling SM
0	X0 input rising edge interrupt	SM40	30	COM 0 character transmission interrupt	SM48
1	X1 input rising edge interrupt	SM41	31	COM 0 character receiving interrupt	SM49
2	X2 input rising edge interrupt	SM42	32	COM 0 frame transmission interrupt	SM50
3	X3 input rising edge interrupt	SM43	33	COM 0 frame receiving interrupt	SM51
4	X4 input rising edge interrupt	SM44	34	COM 1 character transmission interrupt	SM52
5	X5 input rising edge interrupt	SM45	35	COM 1 character receiving interrupt	SM53
6	X6 input rising edge interrupt	SM46	36	COM 1 frame transmission interrupt	SM54
7	X7 input rising edge interrupt	SM47	37	COM 1 frame receiving interrupt	SM55
8	COM2 frame transmission interrupt	SM59	38	COM 2 character transmission interrupt	SM57
9	COM2 character receiving interrupt	SM60	39	COM 2 character receiving interrupt	SM58
10	X0 input falling edge interrupt	SM40	40	High-speed counter interrupt 6	SM65
11	X0 input falling edge interrupt	SM41	41	High-speed counter interrupt 7	SM65
12	X0 input falling edge interrupt	SM42	42	PTO (Y2) output completioninterrupt	SM72
13	X0 input falling edge interrupt	SM43	43	PTO (Y3) output completioninterrupt	SM73
14	X0 input falling edge interrupt	SM44	44	PTO (Y4) output completioninterrupt	SM74
15	X0 input falling edge interrupt	SM45	45	PTO (Y5) output completioninterrupt	SM75
16	X0 input falling edge interrupt	SM46	46	PTO (Y6) output completioninterrupt	SM76
17	X0 input falling edge interrupt	SM47	47	PTO (Y7) output completioninterrupt	SM77

Event number	Interrupt event	Enabling SM		Event number	Interrupt event	Interrupt enabling SM
18	PTO (Y0) output completioninterrupt	SM63		50	Interpolation completion interrupt 1	SM69
19	PTO (Y1) output completion interrupt	SM64				
20	High-speed counter interrupt 0	SM65				
21	High-speed counter interrupt 1	SM65		53	Position-based interrupt 0 of high-speed output	SM61
22	High-speed counter interrupt 2	SM65		54	Position-based interrupt 1 of high-speed output	SM62
23	High-speed counter interrupt 3	SM65		55	Position-based interrupt 2 of high-speed output	SM105
24	High-speed counter interrupt 4	SM65		56	Position-based interrupt 3 of high-speed output	SM106
25	High-speed counter interrupt 5	SM65		57	Position-based interrupt 4 of high-speed output	SM107
26	Timed interrupt 0	Intervals of timed interrupt: SD66 Enable control: SM66		58	Position-based interrupt 5 of high-speed output	SM108
27	Timed interrupt 1	Intervals of timed interrupt: SD67 Enable control: SM67		59	Position-based interrupt 6 of high-speed output	SM98
28	Timed interrupt 2	Intervals of timed interrupt: SD68 Enable control: SM68		60	Position-based interrupt 7 of high-speed output	SM99
29	Power failure interrupt	SM56				

9.2 Processing interrupt event

1. When a certain interrupt event occurs, if it has been enabled, its corresponding event number is added to the interrupt request queue record, which is afirst-in-first-out queue with 8-record long.

2. Processing of the interrupt request by the system:

(1) If the system detects that any request in the interrupt queue, it stops the normal execution of the user program.

(2) The system reads the head record in the interrupt request queue, which records the number of the interrupt event that occurred first. The user interrupt program corresponding to the event number is called and executed.

(3) When the corresponding interrupt program has been executed (the interrupt return instruction has been executed), the interrupt request is processed, the corresponding head record of the request queue is deleted from the queue, and all the following records takes one step forward.

(4) The system detects again whether the interrupt request queue is empty. If it is not empty, the system repeats the above procedures until the queue is empty.

(5) When the interrupt request queue is empty, the system continues to execute the interrupted main program.

3. The system can handle only one interrupt request at one time. When the system is processing an interrupt request,a new interrupt event is added to the interrupt request queue rather than being responded immediately. The system processes it after all the requests ahead of it in the queue are processed.

4. When there are 8 records in the interrupt request queue, the system automatically masks the new interrupt eventso that no new request is added to the queue. The mask is not cancelled until all the requests in the queue have been processed and the interrupted main program has been executed.

📖 Note

1. The interrupts should be brief, or abnormalities may occur, including the mask of other interrupt events (missing of interruptrequests), system scan overtime and low execution efficiency of the main program.
2. It is prohibited to call other subprograms in the interrupt program.
3. If you want to refresh I/O immediately during the interrupt, you can use the REF instruction. You need to note that the execution time of the REF instruction is related to the number of the I/O points to be refreshed.
4. An interrupt event can generate an interrupt request only when the corresponding interrupt event flag is enabled (each type of interrupt event enable/disable control corresponds to the related SM elements. To enable some type of interrupt events requiressetting the corresponding SM element to ON), and the global interrupt enable flag is turned on.
5. When an interrupt request with no corresponding interrupt program in the user program is generated, the system responds to the interrupt request, but no operation.

9.3 Using timed interrupt

- Description of the timed interrupt

The timed interrupt is the interrupt event generated by the system periodically based on the set intervals.

The timed interrupt program is mainly applicable to the situation that requires timed and immediate processing by the system,such as the timed sampling of analog inputs, and timed analog output updating based on certain waveforms.

You can set the intervals (unit: ms) for the timed interrupts by setting the corresponding SD elements. The system generates the interrupt event only when the set time interval is reached (recommended mini.interval: > 4ms).

The ON/OFF state of certain SM elements can enable/disable the corresponding timed interrupts.

The IVC Series PLCs provides you with three timed interrupt resources as follows.

Table 9-1 Timed interrupt resource list

Timed interrupt	Interrupt event number	(SD)Intervals of timed interrupt (SD)	Enable control (SM)
0	26	SD66	SM66
1	27	SD67	SM67
2	28	SD68	SM68

📖 Note

1. Setting of enable control elements cannot affect the execution of the timed interrupts in the interrupt request queue.
2. The timing for a re-enabled interrupt starts from zero.

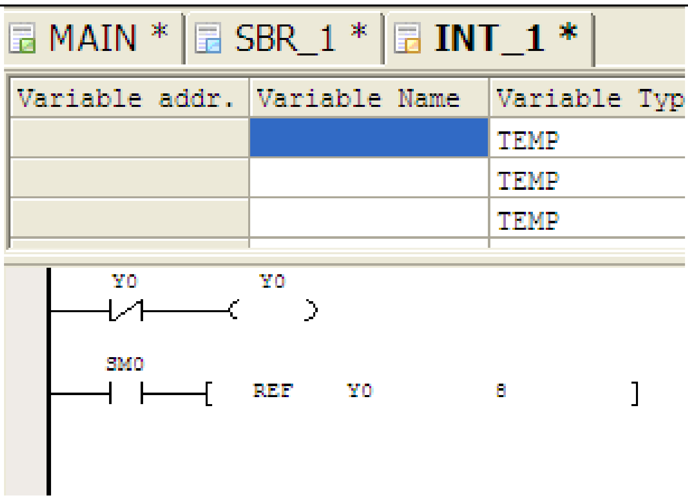
To change the interval of the timed interrupt when the program is running, it is recommended to follow the following

procedures:1. To disable the timed interrupt.2. To change the interval.3. To enable the timed interrupt.

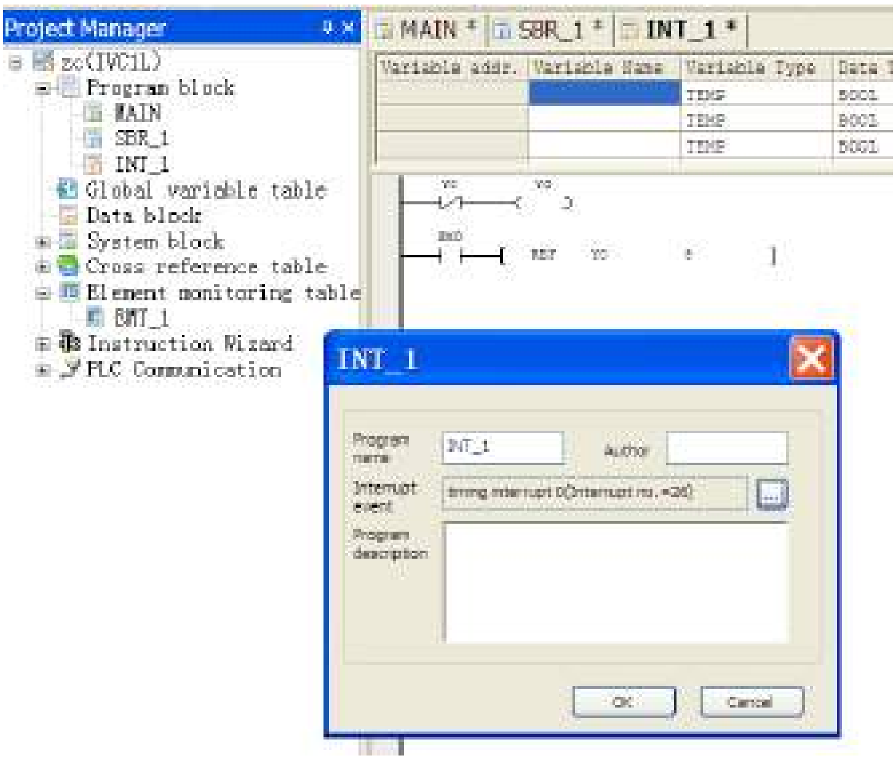
- Application instance of the timed interrupt

In the application instance, the system upsets the output of Y0 once a second based on the timed interrupt 0, which makes Y0 flashing periodically.

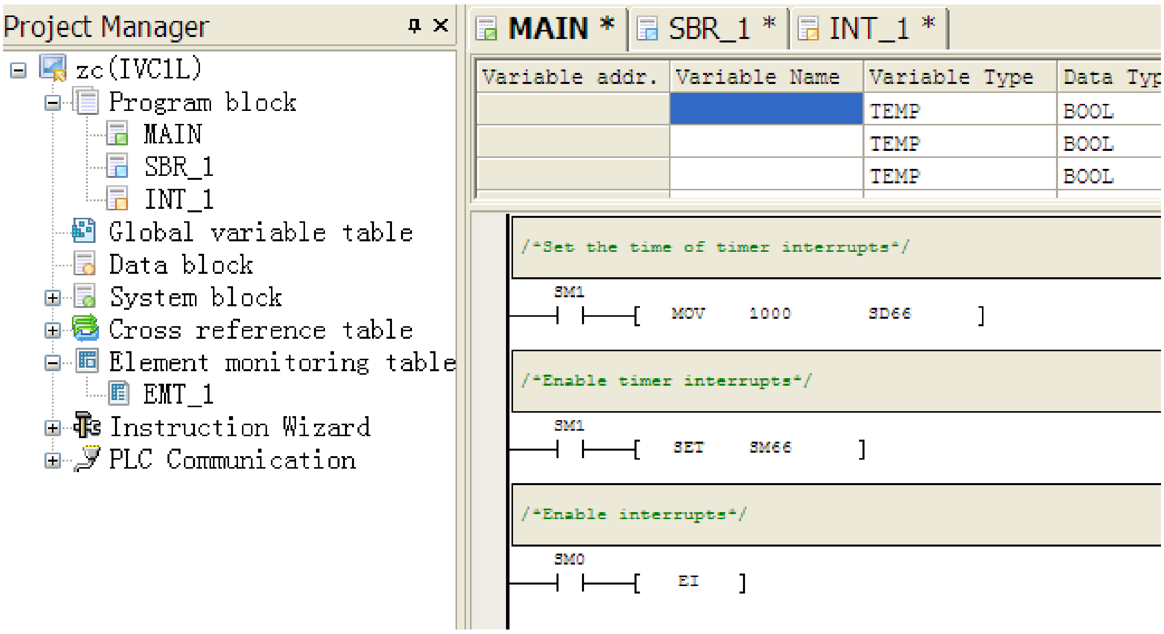
1. Writing an interrupt program for the interrupt event.



2. Specifying an interrupt event number for the interrupt program:



3. Setting the interval for the timed interrupt and enabling the timed interrupt in the main program (MAIN).



9.4 Using external interrupts

- Description of the external interrupts

The external interrupts are related to the actual input points of the PLCs, which are classified into input rising edge interrupt and input falling edge interrupt. In the user program, they add the actions related to external events to the external interrupt program.

The highest response frequency of the system to the external event is 1K. The external events over 1K may be lost.

The rising edge interrupt and falling edge interrupt cannot be used on the same port simultaneously. All the external interrupts are only valid when the global interrupt control EI and corresponding enabling SM are valid.

The detailed relationship is as follows:

Interrupt number	Enabling element		Interrupt number	Enabling element
0 or 10	SM40		4 or 14	SM44
1 or 11	SM41		5 or 15	SM45
2 or 12	SM42		6 or 16	SM46
3 or 13	SM43		7 or 17	SM47

The external interrupts are numbered as follows:

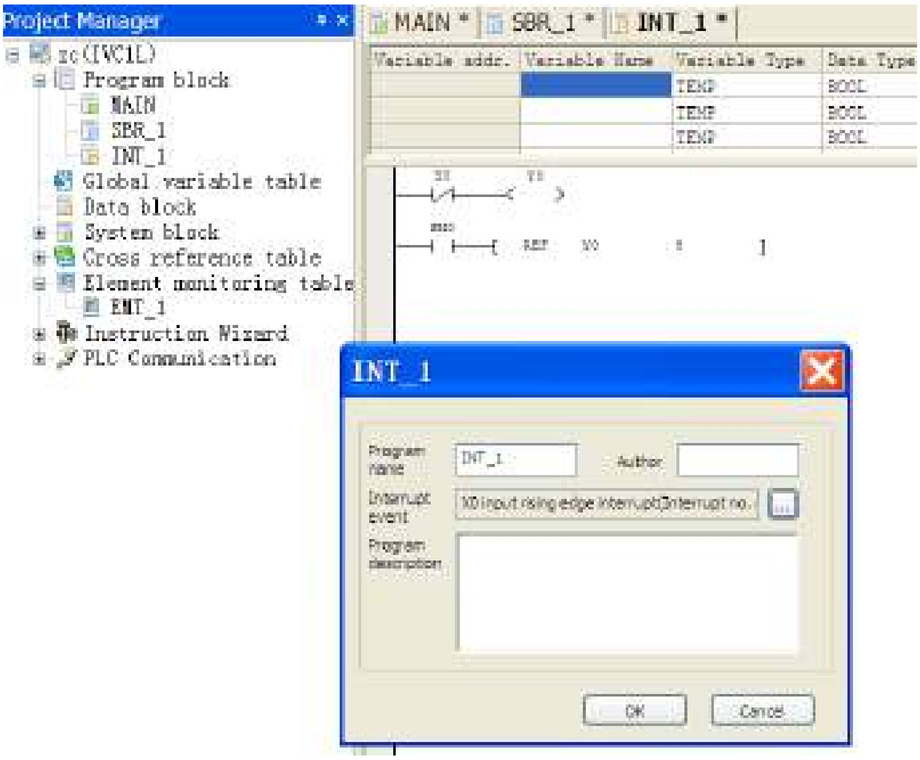
Interrupt number	Interrupt source		Interrupt number	Interrupt source
0	X0 input rising edge interrupt		9	Reserved
1	X1 input rising edge interrupt		10	X0 input falling edge interrupt
2	X2 input rising edge interrupt		11	X1 input falling edge interrupt
3	X3 input rising edge interrupt		12	X2 input falling edge interrupt
4	X4 input rising edge interrupt		13	X3 input falling edge interrupt
5	X5 input rising edge interrupt		14	X4 input falling edge interrupt
6	X6 input rising edge interrupt		15	X5 input falling edge interrupt
7	X7 input rising edge interrupt		16	X6 input falling edge interrupt
8	Reserved		17	X7 input falling edge interrupt

The single input pulse frequency of X0 - X7 is less than 200Hz.

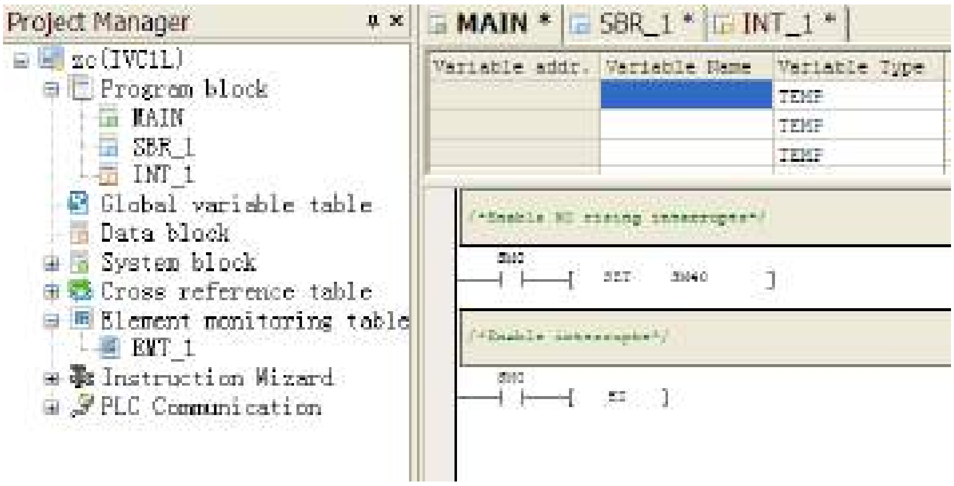
- Application instance of the external interrupt

In the application instance, the system upsets the output of Y0 based on the corresponding external interrupt 0 function and rising edge input event of X0.

1. Writing the interrupt program to upset the state of Y0 once upon every interrupt and output immediately. To use an interrupt,you should select its corresponding interrupt number. For details, you can see the specific operation in the following figure.



2. Writing the EI instruction in MAIN, and setting SM40, the interrupt enabling flag corresponding to X0 input rising edge interrupt, to be valid.



9.5 Using high-speed counter interrupt

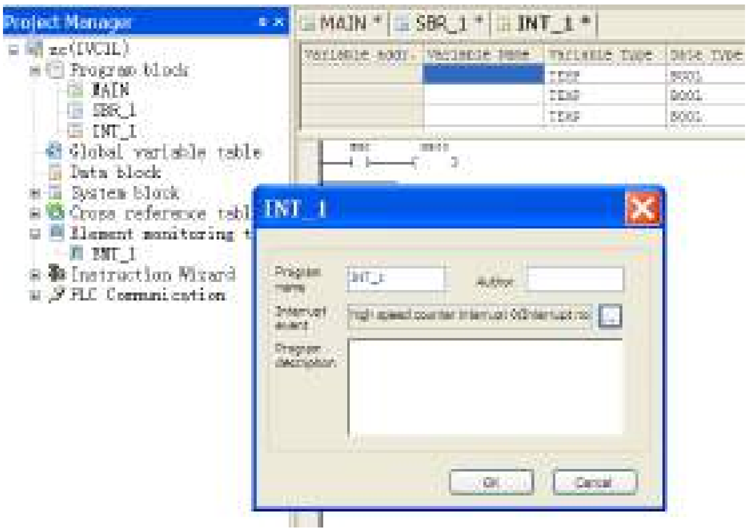
- Description of the high-speed counter interrupt

The high-speed counter interrupt is valid only when it is used together with the HCNT instruction or DHSCI instruction, and generates high-speed counter interrupt based on the count value of the high-speed counter. You can compile programs related to external pulse input in the high-speed interrupt program. All high-speed counter interrupts (20﹣25,42, and 43) arevalid only when the EI instruction and corresponding interrupt enable flag are valid.

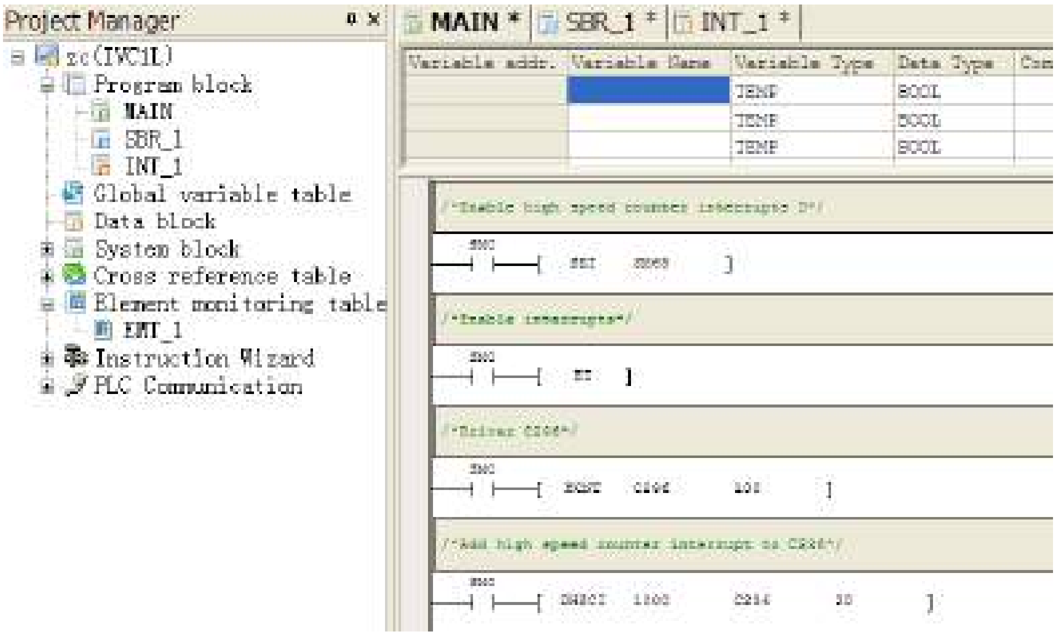
- Application instance of the high-speed counter interrupt

In the application instance, the system uses the high-speed counter interrupt instruction of X0 to respond to the interrupt program (number 20) when the value of the externalcounter C236 reaches the value specified through the DHSCI instruction.

1. Writing the interrupt subprogram. To use an interrupt subprogram, you need to select its corresponding interrupt number. For details, you can see the specific operation in the following figure.



2. Writing the EI instruction in MAIN, and setting SM65, the interrupt enabling flag corresponding to the high-speed counter interrupt, to be valid. Driving the high-speed counter C236 and high-speed counter interrupt instruction.



9.6 Using PTO output completion interrupt

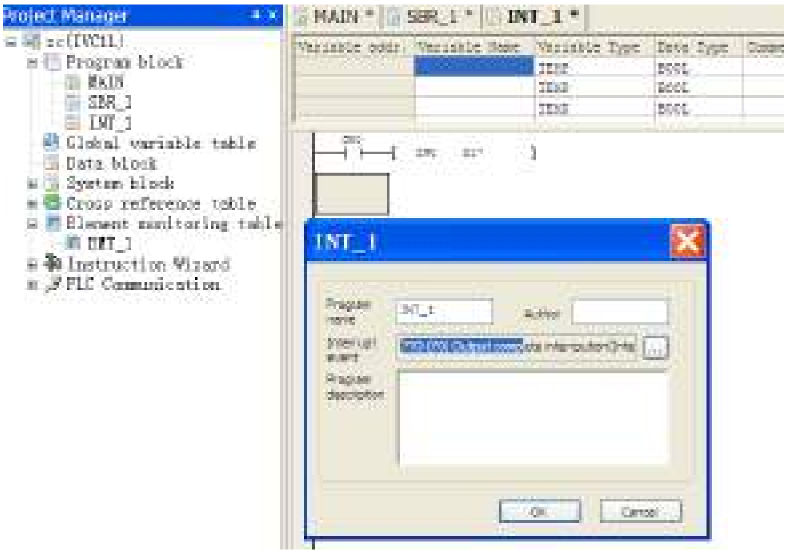
- Description of the PTO output completioninterrupt

The PTO output completion interrupt is triggered when the enable flag (SM63 or SM64) is set and the high-speed pulse output at Y0 or Y1 is finished. You can carry out the relevant processing in the interrupt subprogram. This function is applicable only to the IVC1 series PLCs.

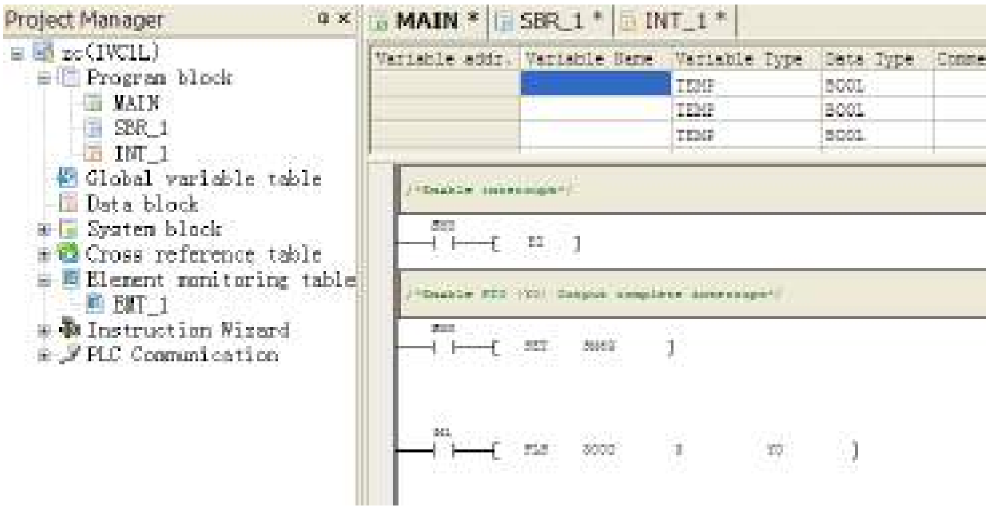
- Application instance of the PTO output completion interrupt

In the application instance, the system uses the high-speed pulse output of Y0 to respond to the interrupt program (number 18) when Y0 high-speed pulse output is finished.

1. Writing the interrupt program (INT_1): Writing a program for the interrupt code to realize the control. To use an interrupt subprogram, you need to select its corresponding interrupt number. For details, you can see the specific operation in the following figure.



2. Writing the functions in MAIN: Enabling the global interrupt of the system and setting the PTO output completion interrupt enable flag SM63 to be valid. Using the PLS instruction.



9.7 Using the power failure interrupt function

When the enable flag SM56 is set and the main module has been detected the power failure, the power failure interrupt is triggered and you can perform the relevant processing in the interrupt subprogram. This function is applicable only to the IVC1 series PLCs.

The power failure interrupt subprogram is executed when the system has no external power supply, so the execution time of the power failure interrupt subprogram cannot exceed 5ms. Otherwise, the elements that can be saved at power outage cannot be completely saved.

9.8 Using serial port-based interrupt

- Description of the serial port-based interrupt

When a serial port is in the free-port protocol mode, the system generates an interrupt event according to the sending or receiving events of the serial port.

For each serial port, the system providesfour interrupt resources for you. The serial port-based interrupt programis mainly applicable to the scenario that requires special processing on the receiving and sending of character/frame (XMT and RCV) at the serial port andtimely processing by the system. It can immediately respond to the processing requirements after a character or frame is transmitted or received, regardless ofthe scan time.

You can set the ON/OFF state of the corresponding SM elements to enable or disable the serial port-based interrupt. When the serial port-based interrupt is disabled, the ones that have been added to the interrupt queue continue to be executed. The XMT instruction cannot be called in the character transmission interrupt processing subprogram when the power flow is normally on. Otherwise, it may lead to the interrupt subprogram nesting and thus block the execution of the user program.

Frame receiving and sending interrupts refer to the interrupt events that are triggered after the XMT and RCV instructions are executed.

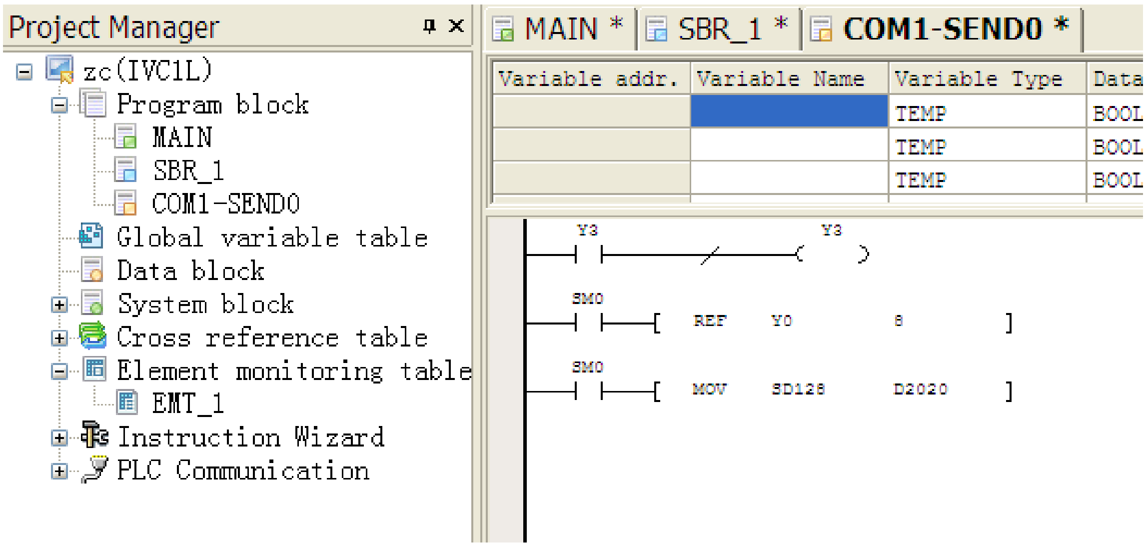
Serial-port interrupt resource list:

Interrupt number	Interrupt event	Interrupt enabling SM
30	Character transmission interrupt of PORT0	SM48
31	Character receiving interrupt of PORT0	SM49
32	Frame sending interrupt of PORT0	SM50
33	Frame receiving interrupt of PORT0	SM51
34	Character sending interrupt of PORT1	SM52
35	Character receiving interrupt of PORT1	SM53
36	Frame sending interrupt of PORT1	SM54
37	Frame receiving interrupt of PORT1	SM55
8	Frame sending interrupt of PORT2	SM59
9	Frame receiving interrupt of PORT2	SM60
38	Character sending interrupt of PORT2	SM57
39	Character receiving interrupt of PORT2	SM58

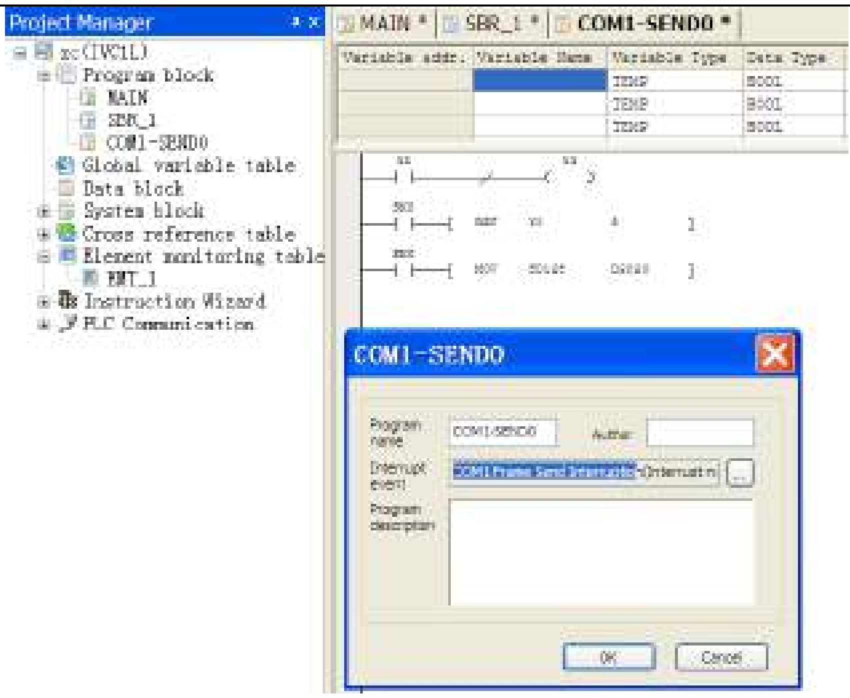
- Application instance of theserial port-based interrupt

In the example, with the serial port-based frame sending function, the system upsets the Y3 output once when aframe is sent out and generate flashing effect based on the frequency of the character sending frame.

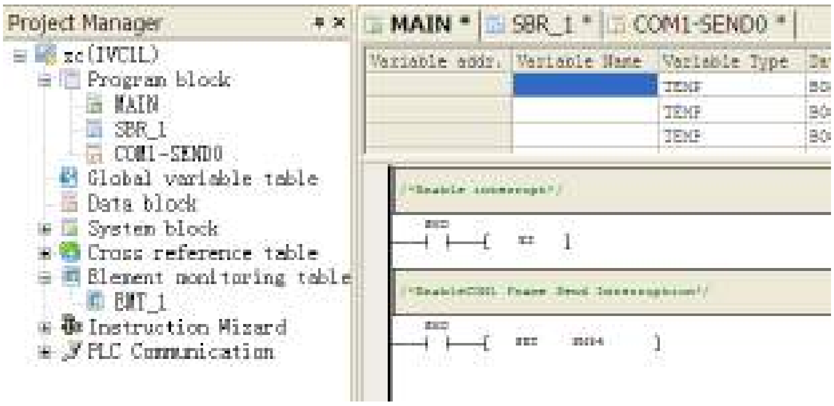
1. Writing the interrupt program and the processing code when the serial port sending frame is completed and the interrupt is triggered.



2. Specifying the corresponding interrupt event number for the interrupt program:



3. Writing the code of enabling serial port sending frame interrupt in MAIN.



For details about the application instance of the serial port interrupt, refer to Chapter 10"Communication function guide".

9.9 Example of short time pulse measurement

The following contents currently are only applicable to IVC1.

The high-speed ring counter provides a high-precision counting function in units of 0.1ms, which can be used in conjunction with input interrupts to measure the pulse widths for short periods of time.

The relevant elements are shown as follows:

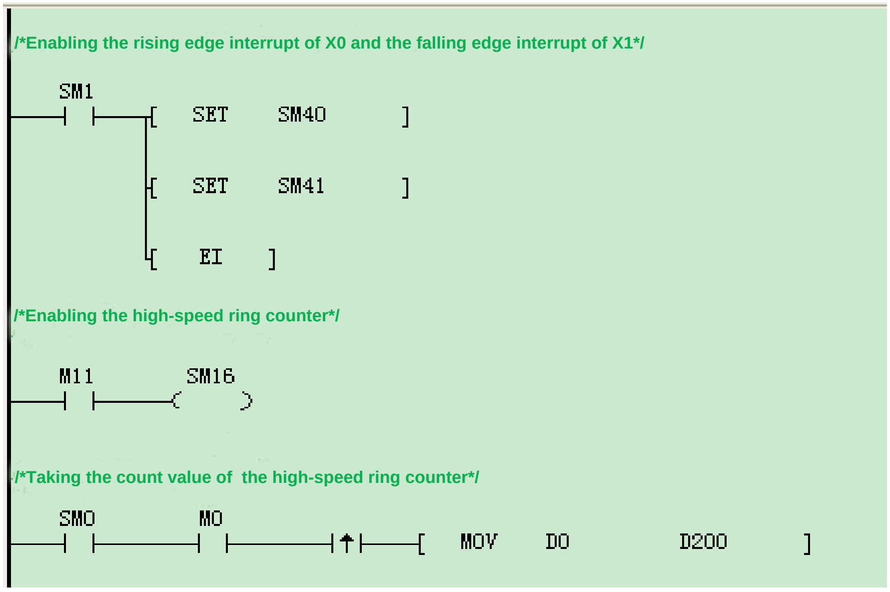
	Name	Function	Attribute	Range
SM16	High-speed ring counter enable flag bit	Unit: 0.1 ms, 16-bit Set: The high-speed ring counter starts counting Reset: The high-speed ring counter stops counting	R/W	
SD16	High-speed ring counter	0-20971 (Unit: 0.1 ms, 16-bit) increment ring counter. The 0.1 m clock is incremented from the next operation cycle after SM16 is set. When the count value exceeds 20971, it starts from zero again. The error is determined by the time at which a single instruction is executed.	R/W	0-20971

It is required to reset SD16 before using the high-speed ring counter.

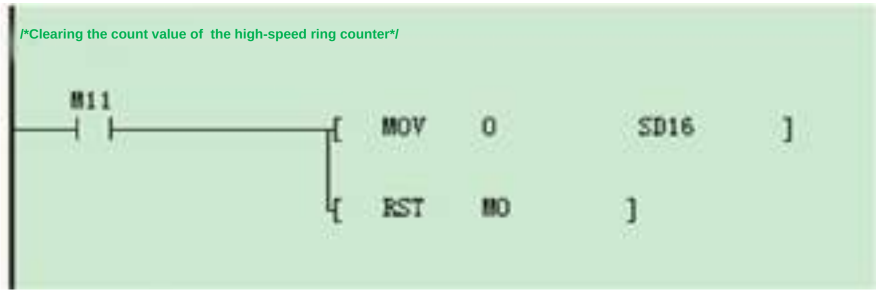
Below is an example of measuring the pulse width, connecting the signal to be measured to the X0 and X1 terminals, and setting the interrupt priorities of X0 and X1 to high.

In the following program, D200 indicates the width of the measured pulse (unit: 0.1ms)

Main program:



Rising edge interrupt of X0



Falling edge interrupt of X1.

Chapter 10 Communication function guide

This chapter detailedly introduces the communication functions of the IVC series micro-PLCs, including the communication resources and communication protocols, and uses application instances to illustrate.

10.1 Communication resources

The baud rates applicable to IVC series small PLC are listed in the following table:

Communication protocol	Applicable baud rate
Free-port protocol, Modbus protocol	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200
N:N protocol	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200

Communication protocols supported by the IVC series micro-PLCs are listed in the following table.

Main module	Communication port	Port type	Supported protocol
IVC2L	PORT0	RS232	Programming port protocol, free-port protocol, Modbus communication protocol (slave station), and N:N communication protocol (master and slave station)
	PORT1	RS232 or RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
IVC1	PORT0	RS232	Programming port protocol, free-port protocol, Modbus communication protocol (slave station), and N:N communication protocol (master and slave station)
	PORT1	RS232 or RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
IVC1L	PORT0	RS232	Programming port protocol, free-port protocol, Modbus communication protocol (slave station), and N:N communication protocol (master and slave station)
	PORT1	RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
	PORT2	RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
IVC1S	PORT0	RS232	Programming port protocol, free-port protocol, Modbus communication protocol (slave station), and N:N communication protocol (master and slave station)
	PORT1	RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
IVC3	PORT0	RS232	Programming port protocol, free-port protocol, Modbus communication protocol (slave station), and N:N communication protocol (master and slave station)
	PORT1	RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
	PORT2	RS485	Free-port protocol, Modbus communication protocol (master station and slave station), and N:N communication protocol (master and slave station)
	PORT3	USB	Programming port protocol
	PORT4	Ethernet	Programming port protocol, TCP_Modbus
	PORT5	CAN	CANopen

Besides, you can set the mode selection switch of the IVC series micro-PLCs to TM so as to forcibly transfer PORT0 into the programming port protocol.

10.2 Programming port communication protocol

The programming port protocol is an internal protocol dedicated to the communication between the PC software Auto Station and the main module.USB, network port, and serial port can be used to communicate with the main module. Only one of them can be selected for monitoring, and they cannot be used simultaneously.

10.3 Free-port communication protocol

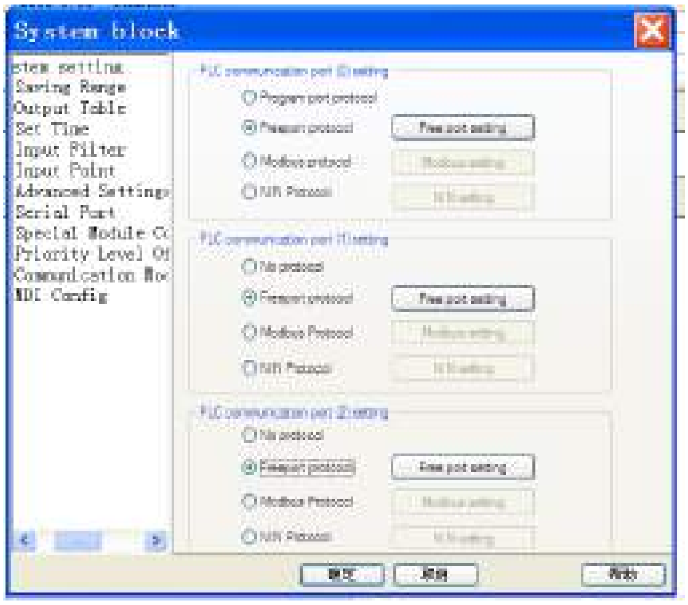
10.3.1 Introduction

The free-port protocol is a communication mode with user-defined data file format, which can realizes data sending and receiving through the instructions. It supports two data formats: ASCII and binary. The free-port communication is available only when the PLC is in RUN state.

The free-port communication instructions include XMT (free-port sending instruction) and RCV (free-port receiving instruction).

10.3.2 Parameters setting of free-port

Selecting**Communication Port** in the system block dialogue box, and selecting **Free port protocol** in he corresponding setting area to enable the Free port setting button, as shown in the following figure.



Clicking any one of the free port setting buttons to enter the Freeport parameter setting interface, as shown in the following figure.

Configurable items are listed in the following table.

Item	Setting content	Remark
Baud rate	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200. Default: 9600	—
Data bit	7 or 8 (default)	—
Parity check	None (default), odd, even	—
Stop bit	1 (default) or 2	—
Allow the start character detection	Enabled or disabled (default)	—
Start character detection	0–255 (corresponding to 00–FF)	Start receiving after the designated start character is detected, and saving the received characters (including the start character) to the designated buffer
Allow the end character detection	Enabled or disabled (default)	—
End character detection	0–255 (corresponding to 00–FF)	Stopping receiving after the preset end character is received, and saving the end character to the buffer
Allow the interframe timeout	Enabled or disabled (default)	—

Intercharacter timeout	0–65535 ms	Stop receiving if the interval between two received characters is longer than the preset intercharacter timeout
Interframe timeout enable	Enabled or disabled (default)	—
Interframe timeout	0–65535 ms	When the energy flow is turned on and the communication conditions are met, that is, the timing starts when the communication serial port starts to receive frames. If the receiving of one frame is finished when the time is up, the receiving is terminated.

10.3.3 Free-port instruction

- Note

The free-port instructions XMT and RCV can be used to send/receive data to/from the designated communication port.For details about the usage of the free-port instructions, refer to section 6.12.11 "XMT: Free-port sending instruction" and section 6.12.12 "RCV: Free-port receiving instruction".

You need to note that to use the free-port instruction on a certain port, you need to set the free-port protocol and communication parameter for the communication port through the system block of Auto Station. After setting, you need to download the system setting to the PLCs and restart it.

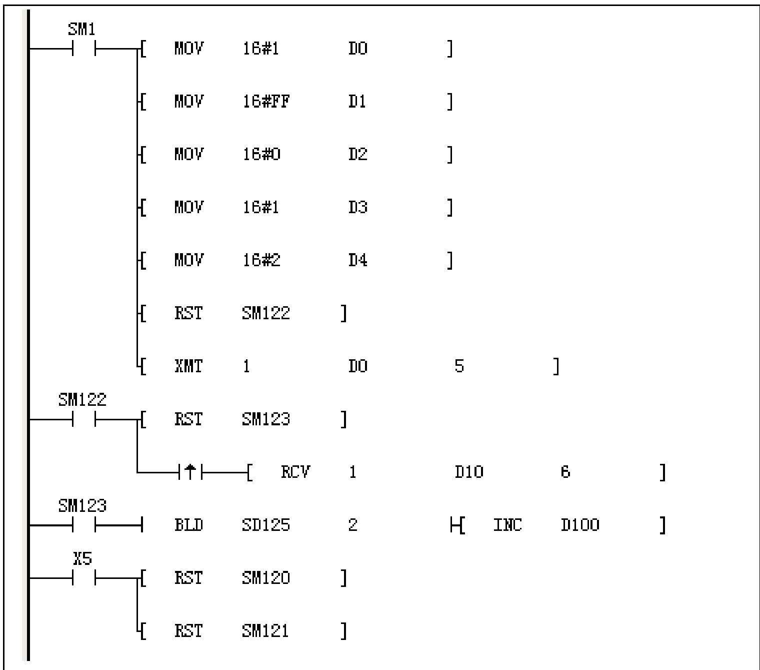
- Application instance

Example 1: Sending a 5-byte data, and then receiving a 6-byte data through PORT1.

Data to be sent:	01	FF	00	01	02		Data to received	01	FF	02	03	05	FE

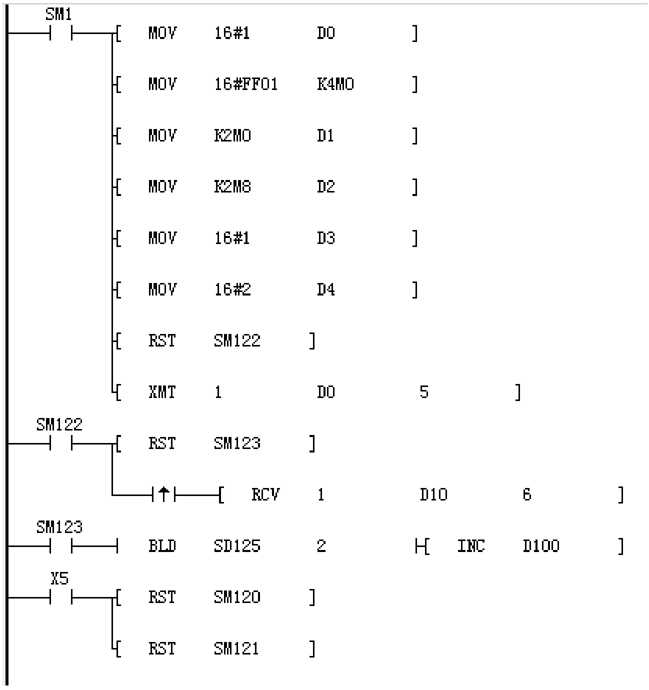
Storing the received data in D elements starting from D10. Each byte occupies one D element, as shown below:

01	FF	02	03	05	FE
D10	D11	D12	D13	D14	D15



1. First, you need to change the setting of communication port in the system block to free-port communication and set the related parameters, such as baud rate and parity.
2. Storing the to-be-sent data in the communication BFM buffer starting from D0, sending data through the XMT instruction, and resetting SM122 (sending completion flag bit) before the sending when SM1 is valid.
3. Setting SM122 after the data is sent, and beginning to receive data upon the rising edge. The max. length for the received characters is 6.
4. Setting SM123 after the data is received, and performing the corresponding operations based on the receiving completion information register (SD125).
5. Using X5 as the enable bit for interrupting the sending and receiving.

Example 2: Sending the data through PORT1, and then receiving the data.



Different from "**Example 1**", before sending the high and low bytes of a word element, the word element needs to be divided into two parts: high and low bytes.
For instance, if you want to send the content of D2, you can store its high bytes and low by tesseparately in D3 and D4, and then send D3 and D4. You can also store the data in a K4MX (such as K4M0 in this program)element, and then take K2M0 as the high byte and K2M8 as the low byte.

10.4 Modbus communication protocol

10.4.1 Introduction

For the serial communication of the IVC series micro-PLCs, Modbus communication protocol is available. Two communication modes: ASCII and RTU (IVC1 only supports the RTU mode) are supported. The PLCs can be set as the master or slave station.

10.4.2 Characteristics of links

1. Physical layer: RS-232 and RS-485

2. Link layer: asynchronous transmission mode

(1) Data bit: 7 bits (ASCII) or 8 bits (RTU)

(2) Transmission rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200

(3) Parity method: even , odd, or none

(4) Stop bit: 1 or 2 stop bits

3. Networking configuration: a maximum of 31 sets of equipments. Address range: 1﹣247. Broadcast is supported.

10.4.3 RTU transmission mode

1. Hex data.

2. The interval between two characters shall not be less than the time of 1.5 characters.

3. There is no frame head or tail, and the interval between two frames is at least the time of 3.5 characters.

4. Using the CRC16 check.

5. The max. length of the RTU frame is 256 bytes and the structure of a frame is shown as follows:

Structure of a frame	Address	Function code	Data	CRC
Number of bytes	1	1	0–252	2

6. Calculation of interval among characters:

If the communication baud rate is 19200, the interval of 1.5 characters is 1/19200×11×1.5×1000 = 0.86ms.

The interval of 3.5 characters is 1/19200×11×3.5×1000 = 2ms.

10.4.4 ASCII transmission mode

1. Using the ASCII data communication.

2. The frame takes ": (3A)"as the head, and CRLF (0D 0A) as the tail.

3. The allowed interval among characters is 1s.

4. Using the LRC check.

5. The frame of ASCII is longer than that of RTU. It is required two character codes for transferring one byte (HEX) in ASCII mode. The max. length for data field (2×252) of ASCII is twice of RTU data field (252). The max . length of ASCII frame is 513 characters and the structure of a frame is shown as follows:

Structure of a frame	Head	Address	Function code	Data	LRC	Tail
Number of bytes	1	2	2	0–2×252	2	2

10.4.5 Supported Modbus function codes

Supported Modbus function codesinclude 01, 02, 03, 05, 06, 08, 15, and 16.

10.4.6 Addressing mode of the PLC elements

1. Mapping between read-write element function codes and the elements:

Function code	Name of function code	Modicon data address	Type of operable element	Remark
01	Read coils	0[note 1]:xxxx	Y, X, M, SM, S, T, and C	Bit read
02	Read discrete input	1[note 2]:xxxx	X	Bit read
03	Read registers	4[note 3]:xxxx[note 4]	D, SD, Z, T, C, and R	Word read
05	Write single coil	0:xxxx	Y, M, SM, S, T, and C	Bit write
06	Write single register	4:xxxx	D, SD, Z, T, C, and R	Word write
15	Write multiple coils	0:xxxx	Y, M, SM, S, T, and C	Bit write
16	Write multiple registers	4:xxxx	D, SD, Z, T, C, and R	Word write

Notes:
1. 0 indicates "coil".
2. 1 indicates "discrete input".
3. 4 indicates "register".
4. xxxx indicates range "1–9999". Each type has an independent logic address range from 1 to 9999 (protocol address starts from 0).
5. 0, 1 and 4 do not have the physical meaning and are not involved in actual addressing.
6. Users shall not write X element with function codes 05 and 15, otherwise, the system does not feed back the error information if the
written operands and data are correct, but the system does not perform any operation on the write instruction.

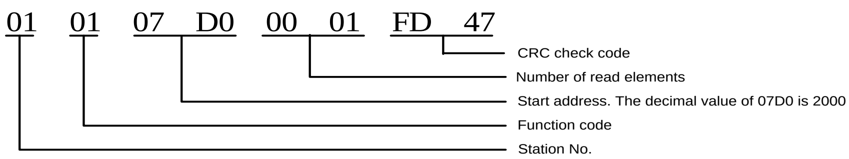
2. Mapping between the PLC elements and Modbus communication protocol address:

Element	Type	Physical element	Protocol address	Supported function code	Remark
Y	Bit	Y0–Y377 (octal code) 256 points in total	0000–0255	01, 05, and 15	Output state, element number: Y0–Y7 and Y10–Y17
X	Bit	X0–X377 (octal code) 256 points in total	1200–01455 0000–0255	01, 05, and 15 02	Input state, it supports two kinds of addresses, and the element number is same as the above
M	Bit	M0–M2047	2000–4047	01, 05, and 15	

Element	Type	Physical element	Protocol address	Supported function code	Remark
		M2048–M10240	12000-20191		
SM	Bit	SM0–SM255 SM256–SM511	4400–4655 30000-30255	01, 05, and 15	
S	Bit	S0–S1023 S1024–S4095	6000-7023 31000-34071	01, 05, and 15	
T	Bit	T0–T255 T256–T511	8000–8255 11000-11255	01, 05, and 15	State of T element
C	Bit	C0–C255 C256–C306	9200–9455 10000-10050	01, 05, and 15	State of C element
D	Word	D0–D7999	0000–7999	03, 06, and 16	
SD	Word	SD0–SD255 SD256–SD511	8000–8255 12000-12255	03, 06, and 16	
Z	Word	Z0–Z15	8500–8515	03, 06, and 16	
T	Word	T0–T255 T256–T511	9000–9255 11000-11255	03, 06, and 16	Current value of T element
C	Word	C0–C199	9500–9699	03, 06, and 16	Current value of C element (WORD)
C	Double word	C200–C255	9700–9811	03 and 16	Current value of C element (DWORD)
C	Double word	C256–C306	10000-10101	03 and 16	Current value of C element (DWORD)
R	Word	R0–R32767	13000-45767	03, 06, and 16	

Note:

The protocol address is the address used on data transmission and corresponds with the logic address of Modicon data. The protocol address starts from 0 while the logic address of Modicon data starts from 1, that is, protocol address + 1 = logic address of Modicon data. For example, if M0 protocol address is 2000, and its corresponding logic address of Modicon data is 0:2001. In practice, the read and write of M0 is completed through the protocol address, for example, frame for reading M0 element (sent by the master station).



3. Mapping between the IVC3 series PLC elements and Modbus communication protocol address:

Element	Type	Physical element	Protocol address	Supported function code	Remark
Y	Bit	Y0–Y777 (octal code) 512 points in total	0000–0511	01, 05, and 15	Output state, element number: Y0–Y7 and Y10–Y17
X	Bit	X0–X777 (octal code) 512 points in total	1200–01711 0000–0511	01, 05, and 15 02	Input state, it supports two kinds of addresses, and the element number is same as the above
M	Bit	M0–M2047 M2048–M10240	2000–4047 12000-20191	01, 05, and 15	
SM	Bit	SM0–SM255 SM256–SM1023	4400–4655 30000-30767	01, 05, and 15	
S	Bit	S0–S1023 S1024–S4095	6000-7023 31000-34071	01, 05, and 15	
T	Bit	T0–T255 T256–T511	8000–8255 11000-11255	01, 05, and 15	State of T element
C	Bit	C0–C255 C256–C511	9200–9455 10000-10511	01, 05, and 15	State of C element
D	Word	D0–D7999	0000–7999	03, 06, and 16	
SD	Word	SD0–SD255 SD256–SD1023	8000–8255 12000-12767	03, 06, and 16	
Z	Word	Z0–Z15	8500–8515	03, 06, and 16	
T	Word	T0–T255 T256–T511	9000–9255 11000-11255	03, 06, and 16	Current value of T element
C	Word	C0–C199	9500–9699	03, 06, and 16	Current value of C element (WORD)

Element	Type	Physical element	Protocol address	Supported function code	Remark
C	Double word	C200–C255	9700–9811	03 and 16	Current value of C element (DWORD)
C	Double word	C256–C306	10000-10101	03 and 16	Current value of C element (DWORD)
R	Word	R0–R32767	13000-45767	03, 06, and 16	

Note:
The protocol address is the address used on data transmission and corresponds with the logic address of Modicon data. The protocol address starts from 0 while the logic address of Modicon data starts from 1, that is, protocol address + 1 = logic address of Modicon data. For example, if M0 protocol address is 2000, and its corresponding logic address of Modicon data is 0:2001. In practice, the read and write of M0 is completed through the protocol address, for example, frame for reading M0 element (sent by the master station).

01 01 07 D0 00 01 FD 47

- 01 – Station No.
- 01 – Function code
- 07 D0 – Start address. The decimal value of 07D0 is 2000
- 00 01 – Number of read elements
- FD 47 – CRC check code

10.4.7 Modbus slave station

Modbus slave station responds to the master station according to the received message of local addressing, rather than sending out message actively. The slave station only supports Modbus function codes 01, 02, 03, 05, 06, 08, 15, and 16, and the other codes are "illegal function codes"(except broadcast frame).

10.4.8 Reading and writing elements

All the function codes supported by IVC2L, except 08 are used for reading and writing elements. In principle, in one frame, there are 2000 bit elements and 125 word elements for reading, 1968 bit elements and 120 word elements for writing at most. However, theactual protocol addresses for different types of elements are different and discontinuous (for example, Y377's protocol addressis 255 while X0's protocol address is 1200). Therefore, when reading or writing an element, the element read at one time can only be the same type, and the max. number of the read elements depends on the number of the elements thatare actually defined. For example, when reading Y element (Y0 – Y377, 256 points in total), the protocol address ranges from 0 to 255, the corresponding logic address of Modicon data is from 1 to 256, and the max. number of the Y elements that can be readis 256.

The examples are as follows:

1. Sending from the master station: 01 01 00 00 01 00 3D 9A

01 – address; 01- function code; 00 00 – starting address; 01 00 – number of read elements; 3D 9A – check

Response of the slave station: providing correct response

2. Sending from the master station: 01 01 00 00 01 01 FC 5A

The master station reads 01 01 (257) elements starting from 0000, which is beyond the defined number of Y elements.

Response of the slave station: 01 81 03 00 51

The response of the slave station is illegal data value, because 257>256, and 256 is the allowed max. number of Y elements.

3. Sending from the master station: 01 01 00 64 00 A0 7D AD

00 64 (decimal 100) is the start address for themaster station to read, and 00 A0 (decimal 160) is the number of elements.

Response of the slave station: 01 81 02 C1 91

The response of the slave station is illegal data address, because there are only 156 Y elements which are defined to the protocol address 100, but 160>156, so 160 is illegal.

4. Sending from the master station: 01 04 00 02 00 0A D1 CD

The master station sends the frame of function code 04

Response of the slave station: 01 84 01 82 C0

The response of the slave station is illegal function codes. Function code 04 is not supported by the IVC2L series.

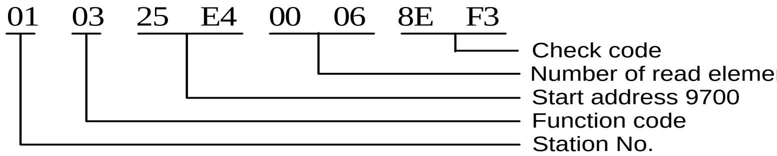
📖 Note

1. X element does not support the write operation (that is, the write of element X is invalid). For details about the writable properties of SM and SDelements, refer to Appendix A "Special auxiliary relay" and Appendix B "Special data register" (if the element is un-writable,the write operation is invalid).
2. The address of the slave station is 01, the last two bytes are CRC check codes and the second byte is function code.

10.4.9 Processing of double word element

The current count value of C element is word or double word element. The values from C200 to C255 are double word elements, which are read and written through the function codes (03 and 16) of read/write registers. The address of every two registers corresponds to one C double word element, and the registers can only be read or written in pair.

For example, reading the RTU frame of three C double word elements from C200 to C202.



In the returned data, two addresses 9700 and 9701 indicate the contents of C200. 9700 is MSB, and 9701 is LSB.

When reading a double word element, if the start address for reading is not an even number, then the system responds with errorcode of illegal address. If the number of the read registers is not an even number, the system responds with error code of illegal data.

For example:

Sending from the master station: 01 03 25 E5 00 04 5E F2

The start address for the reading of master station: 4 word elements of 25 E5 (decimal 9701)

Response of the slave station: 01 83 02 C0 F1

Response of the slave station: illegal data address

Sending from the master station: 01 03 25 E4 00 05 CE F2

The start address for the master station reading: 5 word elements of 25 E5

Response of the slave station: 01 83 03 01 31

The slave station returns the illegal data.

10.4.10 Processing of LONG INT data

One LONG INT data can be saved in two D elements. For example, if a LONG INT data is saved in D3 and D4, the INVT PLC thinks that MSB are stored in D3 and LSB are stored in D4. When the master station reads the LONG INT data through Modbus, the 32-bit data shall be regrouped based on the LONG INT storage principle of INVT PLC after reading the data.The storage principle of FLOAT is the same as the storage principle of LONG INT.

10.4.11 Diagnostic function code

The diagnostic function code is used for testing the communication between the master and slave station, and various internal error conditions of the slave station. The supported diagnostic subfunction codes are shown in the following table:

Function code	Subfunctio n code	Name of the subfunction code		Function code	Subfunctio n code	Name of the subfunction code

Function code	Subfunction code	Name of the subfunction code		Function code	Subfunction code	Name of the subfunction code
08	00	Return the query data		08	12	Return the bus communication error count
08	01	Restart the communication option		08	13	Return the bus exceptional error count
08	04	Forced listen only mode		08	14	Return the slave message count
08	10	Clear the counter		08	15	Return the slave station no response count
08	11	Return the bus message count		08	18	Return the bus character overrun count

Applicable to the IVC2L/IVC3 series

10.4.12 Error code

When the instruction sent from the master station is in the normal response state, the slave station returns data or statistic value in the data field. But in the abnormal response state, the server returns error codes in the data field. The error codes are shown in the following table.

Error code	Meaning of error code
0x01	Illegal function code
0x02	Illegal register address
0x03	Illegal data

In addition, if the slave station receives data under the following situations, no message is returned:

(1)When there are errors in the broadcast frame, such as data error, address error, etc.

(2)When the character is overrun, no message is returned. For example, the RTU frame is more than 256 bytes;

(3)When in the RTU transmission mode, the interval between characters times out, which is equivalent to receiving an error frame,and no message is returned;

(4) When the slave station is in the listen-only mode, no message is returned.

(5)The slave station received ASCII error frame, including frame tail error and character range error.

Note

Read station is configured with compulsory elements. What is read is the value run by the program, which may be inconsistent with the compulsory value.

10.4.13 Modbus parameter setting

- Setting the communication port in the system block

  There are two serial ports (PORT0 and PORT1) on the communication port interface. PORT0 only supports the Modbus slave station while PORT1 supports both the Modbus master and slave stations.

- Setting the Modbus communication protocol parameters

  There is a button of default value on the Modbus operand interface. The default value is the communication setting recommended by the Modbus communication protocol. For details about the parameter setting items, you can refer to the following table.

Item	Setting content
Station No.	0–247
Baud rate	115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200
Data bit	Setting to 7 or 8 bits:7 bits for ASCII mode, and 8 bits for RTU mode
Parity check	Setting to none, odd, and even
Stop bit	Setting to 1 or 2: 1 for odd or even check, and 2 for none.
Modbus master/slave	It can be set to master or slave station: PORT1 can be set to master/slave station, and PORT0 can only be set to slave station
Transmission mode	Selecting RTU or ASCII mode
Timeout time of the	The time for waiting the slave response by the master station exceeds the preset time

Item	Setting content
master mode	
Note: After the operand is set and downloaded in the system block, it is valid only after one operation.	

10.4.14 Modbus instruction

When a PLC is used as a Modbus master station, the Modbus data frame can be sent/received through the Modbus instruction provided by the system. For details about the use of the Modbus instruction, refer to section 6.12.1 "Modbus: Master station communication instruction".

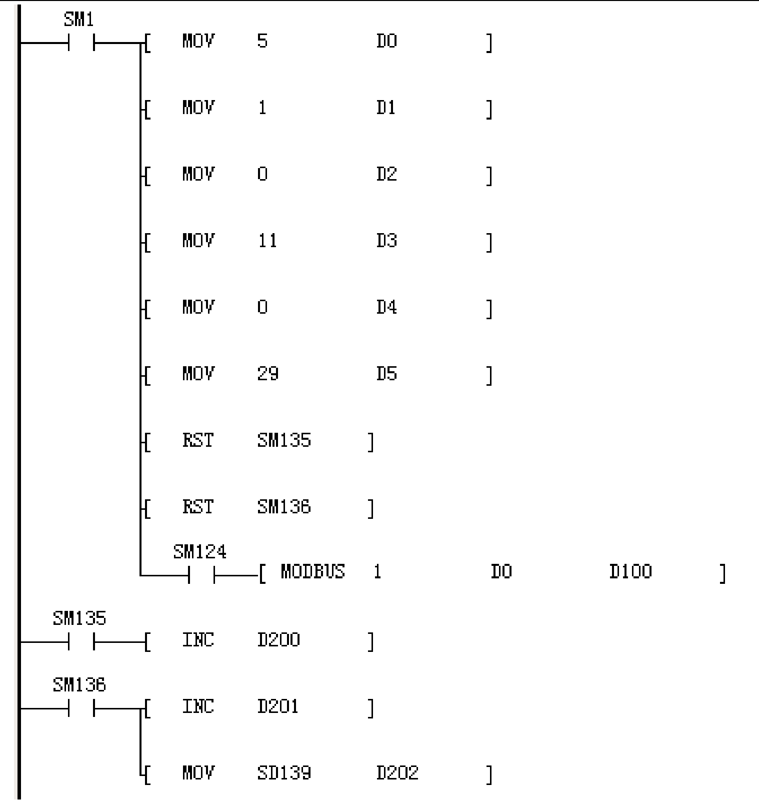
If a PLC is set to be a master station, there is a timeout item in the master mode when setting the Modbus parameter in the system block. To ensure the correctness of the received data, the timeout time shall be longer than a scan cycle of the Modbus slave station and with reasonable margin. For example, if an IVC2L series PLC is used as a slave station, and a scan cycle of the IVC2L series PLCs is 300ms,the master mode timeout of the master station shall be over 300ms. It is proper to set the timeout to be 350ms.

Application program

Example 1: When anIVC2L series PLC is a Modbus master and slave station, it reads the bit element state of 5# station. The protocol address of the slave station read by the master station is the bit values ranging from 11 to 39. Assuming that the read data are as follows, the storage location for the received data starts from D100, D100 saves the address, D101 saves the function code, D102 saves the number of registers, and the units starting from D103 save the read bit element values.

42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27
X	X	X	0	0	1	1	0	0	0	0	1	0	1	1	0
D106								D105							
26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11
1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1
D104								D103							

When the number of registers read is not a multiple of 8, the insufficient upper bits are filled with zero. In the example, the three upper bits (40, 41, and 42) of D106 are filled with zeros.



1. Designating 5 as the address of the slave station
to be accessed (it is saved to D0).
2. Designating 1 as the function code (it is saved to D1).
3. The address of the register to be read is 11 (it is saved to D2/D3 according to MSB and LSB).
4. The number of the registers to be read is 29 (it is saved to D4/D5 according to MSB and LSB).
5. The received data is saved to D100.
6. Adding 1 to D200 if the receiving is completed (SM135 is set).
7. Adding 1 to D201, and saving the error code to D202 if the communication fails (SM136 is set).
8. SM124 is the idle flag of the communication port.

📖 Note

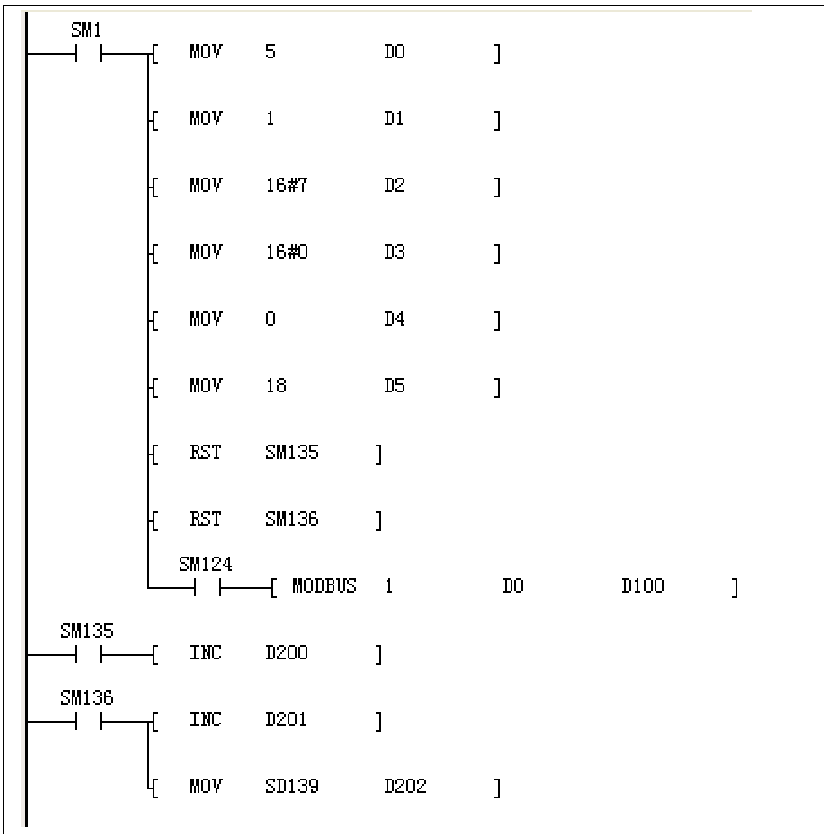
1. When the logic address is used for addressing the bit elements of the IVC2L series PLCs, the logic address 1 is the protocol address 0. In the above example, to read the values of 11–39 bit elements (protocol address) in the slave station, the logic address shall be started from 12.

2. The failure of this communication does not affect the next communication, that is, if there are two Modbus XMT instructions inone user program, the first communication fails and has error code, it does not affect the data sending of the second Modbus instruction. Thus, in the example, the error code in SD139 is placed in D202, so the code can be observed through D202.

3. For the message sending of the slave station, if the master station is in listen-only mode, there is no data to be returned and the system sets the error flag. Therefore, when you use the Modbus network of the IVC2L series, if the IVC2L is used as the master station, the user shall clearly know which PLC slave station is under listen-only mode, so as to ensure that the failure of the communication is not caused by the listen-only mode of the slave station.

Example 2: When an IVC2L series PLC is a Modbus master and slave station, it reads the states of the bit elements 2000–2017 of 5# station protocol address.

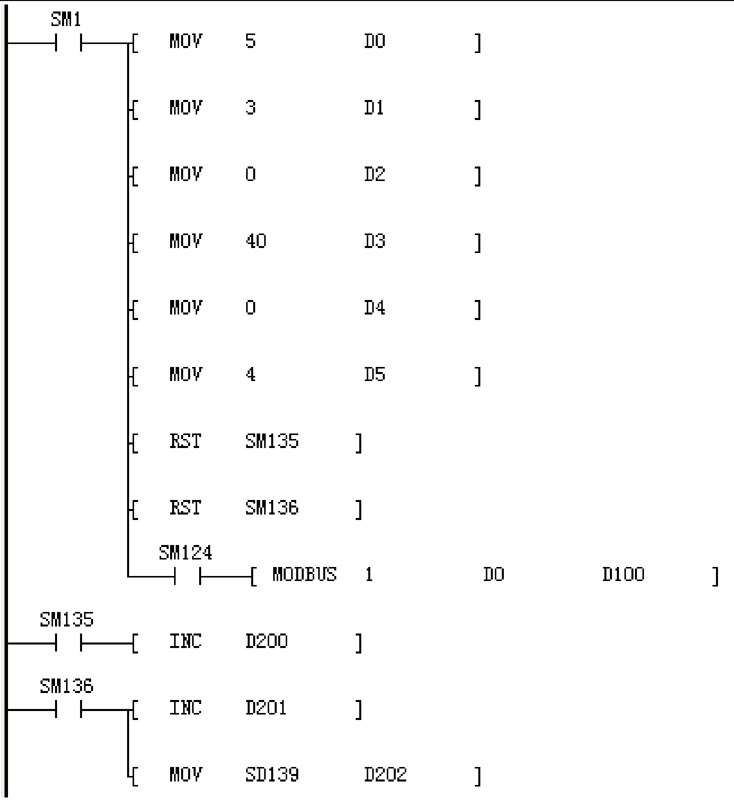
The read datas are as follows. The received frame starts from D100, D100 saves the address, D101 saves the function code, D102 saves the number of registers, and the units starting from D103 save the read bit element values.



1. Designating 5 as the address of the slave station to be accessed (it is saved to D0).
2. Designating 1 as the function code (it is saved to D1).
3. The start address of the register to be read is 07D0 (hex) (it is saved to D2/D3 according to MSB and LSB).
4. The number of the registers to be read is 29 (it is saved to D4/D5 according to MSB and LSB).
5. The received data is saved to D100.
6. Adding 1 to D200 if the receiving is completed (SM135 is set).
7. Adding 1 to D201, and saving the error code to D202 if the communication fails (SM136 is set).
8. SM124 is the idle flag of the communication port.

Example 3: When an IVC2L series PLC is a Modbus master and slave station, it reads the values of the word elements 40–43 of 5# station protocol address.

MSB of element 40	LSB of element 40	MSB of element 41	LSB of element 41	MSB of element 42	LSB of element 42	MSB of element 43	LSB of element 43
D103	D104	D105	D106	D107	D108	D109	D110



1. Designating 5 as the address of the slave station
to be accessed (it is saved to D0).
2. Designating 3 as the function code (it is saved to D1).
3. The start address of the register to be read is 40 (it is saved to D2/D3 according to MSB and LSB).
4. The number of the registers to be read is 4.
5. The received data is saved to D100.
6. Adding 1 to D200 if the receiving is completed (SM135 is set).
7. Adding 1 to D201 and saving the error code to D202 if the communication fails (SM136 is set).
8. SM124 is the idle flag of the communication port.

10.5 N:N communication protocol

10.5.1 N:N introduction

N:N is a micro-PLC network developed by INVT Auto-Control Technology Co., Ltd. The physical layer of N:N adopts RS485, so the PLCs can be directly connected through PORT1 or connected through PORT0 by a RS-232/RS-485 converter to N:N network. The PLCs that are connected to N:N can automatically exchange thevalues between certain D and M elements, which makes the access to the other PLC elements in the network as simple and convenient as accessing its own element. In N:N, the data access between the PLCs is completely equivalent (N:N communication network).

It is convenient to configure N:N. Most parameters of N:N only need to be configured for No.0 PLC. In addition,N:N supports the online modification of the network parameters, and is able to detect the newly added PLCs automatically in the network. If any PLC is disconnected from the network, the other PLCs continue to exchange the data. It isalso able to monitor the communication state of the whole network through the relevant SM elements of any PLC inN:N.

10.5.2 N:N network data transmission form

There are two kinds of messages in N:N tokens issued by the master station, and broadcast by each PLC to its own data.

The token is uniformly issued by the master station. The master station first holds the token. After the data is broadcasted, the token is cyclically and sequentially issued to each slave station. Only the slave station that receives the token can broadcast to other PLCs (including the master station).

Figure 10-1–Figure 10-5 show the main processes of network communication. In the figure, 1# stands as the master station. It needs to be noted that in general, 0# is amaster station by default while 1# is a standby master station (when the master station communication error or power failure occurs, it is switched to the master station).

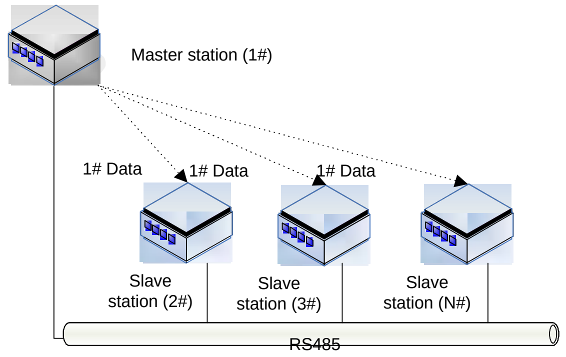


Figure 10-1 Master station broadcasts

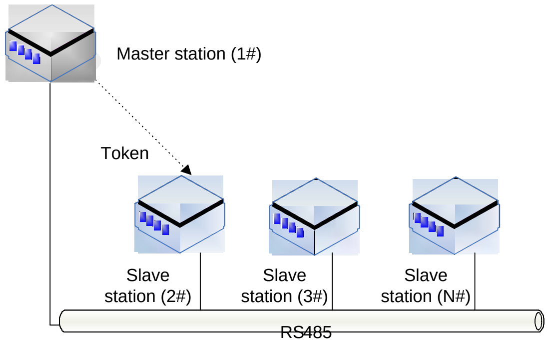


Figure 10-2 Master station sends the token to 2# slave station

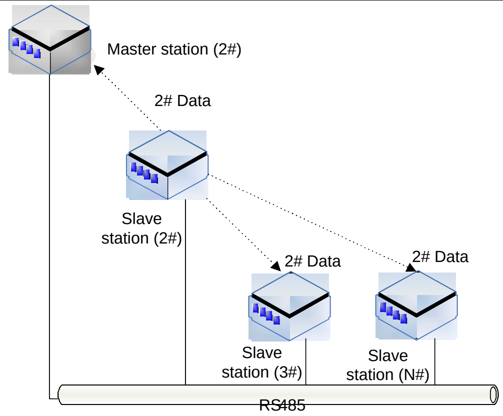


Figure 10-3 2＃slave station broadcasts

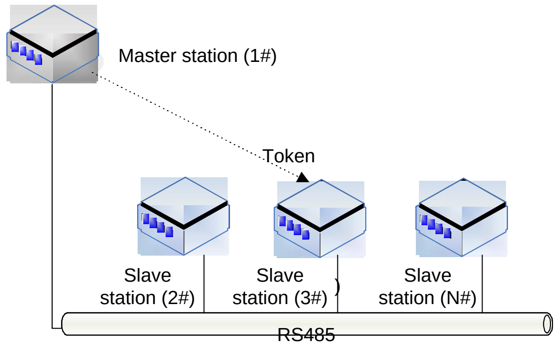


Figure 10-4 Master station sends the token to 3# slave station

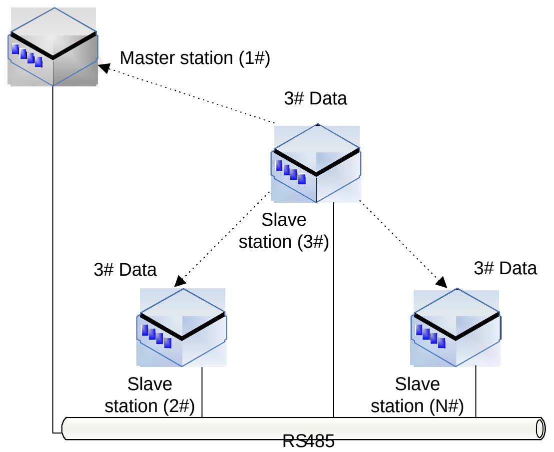


Figure 10-5 3＃slave station broadcasts

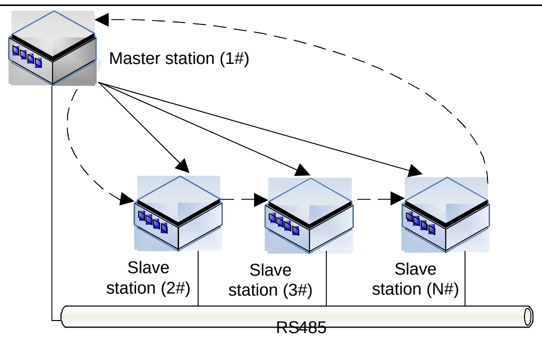


Figure 10-6 Token issue and flow orders

Figure 10-6 shows the flow order of tokens. The heavy line indicates the actual process of issuing the token, and the dashed line indicates the order of the station in which the token is held and broadcasted. You need to note that the token is not passed by one slave station(such as 2#PLC) to another slave station (such as 3#PLC), but the token is first issued by the master station to 2#PLC, and then issued by the master station to 3# PLC.

10.5.3 N:N network structure

N:N supports two kinds of networks: single-layer network and multiple-layer network, as shown in the following figures.

Figure10-7 N:N single-layer network

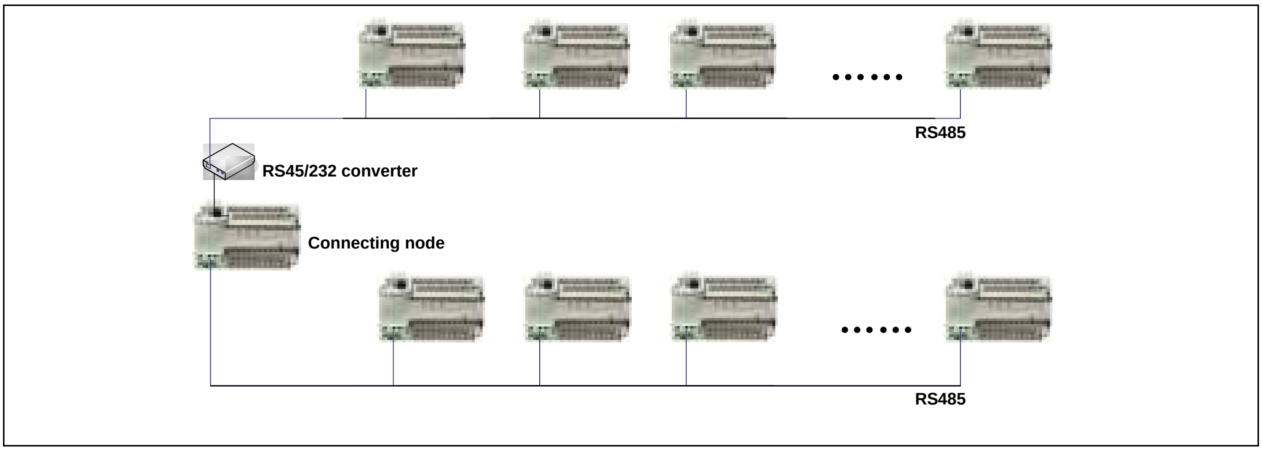
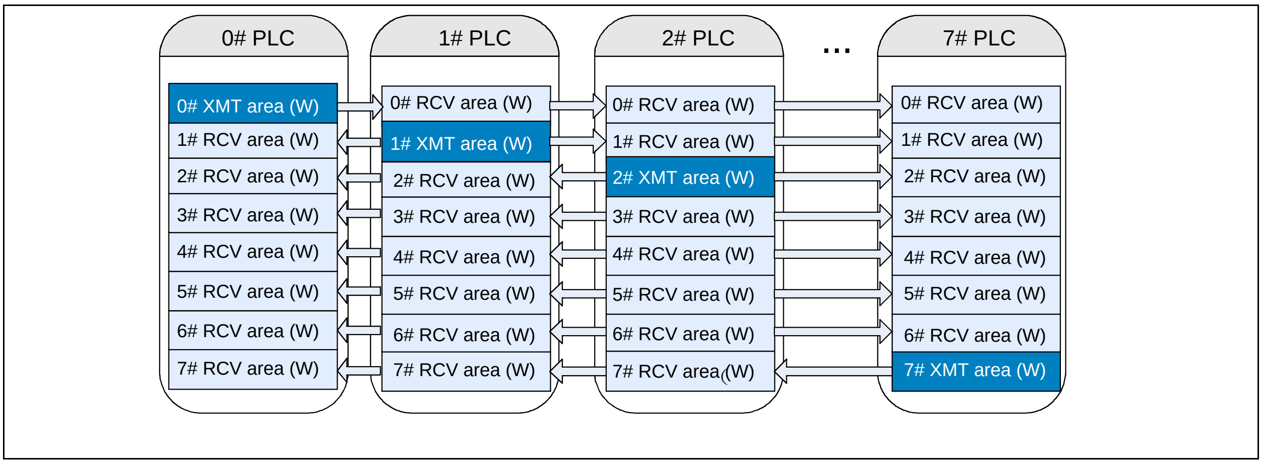


Figure10-8 N:N multiple-layer network

In the single-layer network, each PLC is connected to N:N through only one port. In the multi-layer network, the layer-to-layer PLC (intermediate node) shall be connected, and the two communication ports of the PLC are respectively connected to different layers. The single-layer network can support a maximum of 32 PLCs, and each layer of the multi-layer network can support a maximum of 16 PLCs.

10.5.4 N:N refresh mode

The PLCs connected to N:N can automatically exchange between parts of D elements and M elements in the network. The quantity and No. of these D and M elements are fixed, and these elements are called "Elements Sharing Area". If the PLC uses N:N, the value of the Elements Sharing Area keeps refreshing automatically, so as to keep the value consistency of the Elements Sharing Area for each PLC in the network.



As shown in the above figure, each PLC connected to N:N has a writable sending area in the Element Sharing Area. N:N automatically sends the information (values of designated D and M elements) of the writable sending area to other PLCs, receives the information from other PLCs, and saves it to the read-only sending area.

The number of elements in the Element Sharing Area is fixed (a total of 64 D components and 512 M components can be shared), and these elements are distributed to more than one PLC. Therefore, the fewer PLCs are connected to the network, the more elements can be distributed to each PLC. This relationship is defined by N:N refresh mode table.

- Distribution of D element on N:N single-layer network

<table>
<tr><th>Distribution of D elements in sending area</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="4">#4</td><td rowspan="8">#2</td><td rowspan="16">#1</td></tr>
<tr><td>D7734–D7735</td><td>#17</td></tr>
<tr><td>D7736–D7737</td><td>#18</td><td rowspan="2">#9</td></tr>
<tr><td>D7738–D7739</td><td>#19</td></tr>
<tr><td>D7740–D7741</td><td>#20</td><td rowspan="2">#10</td><td rowspan="4">#5</td></tr>
<tr><td>D7742–D7743</td><td>#21</td></tr>
<tr><td>D7744–D7745</td><td>#22</td><td rowspan="2">#11</td></tr>
<tr><td>D7746–D7747</td><td>#23</td></tr>
<tr><td>D7748–D7749</td><td>#24</td><td rowspan="2">#12</td><td rowspan="4">#6</td><td rowspan="8">#3</td></tr>
<tr><td>D7750–D7751</td><td>#25</td></tr>
<tr><td>D7752–D7753</td><td>#26</td><td rowspan="2">#13</td></tr>
<tr><td>D7754–D7755</td><td>#27</td></tr>
<tr><td>D7756–D7757</td><td>#28</td><td rowspan="2">#14</td><td rowspan="4">#7</td></tr>
<tr><td>D7758–D7759</td><td>#29</td></tr>
<tr><td>D7760–D7761</td><td>#30</td><td rowspan="2">#15</td></tr>
<tr><td>D7762–D7763</td><td>#31</td></tr>
</table>

For example:

(1) In mode 1, the D elements distributed to the sending zone assigned by 0# station are D7700–D7701. The 0# station PLC can write values to D7700 and D7701, and other stations (1#-–31#) can directly read the values of D7700 and D7701.

(2) In mode 2, the D elements distributed to the sending zone assigned by 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 the values of D7700–D7704.

- Distribution of M element on N:N single-layer network

<table>
<tr><th>Distribution of M elements in sending area</th><th>Mode 1</th><th>Mode 2</th><th>Mode 3</th><th>Mode 4</th><th>Mode 5</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="12">#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="4">#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>
</table>

Distribution of M elements in sending area	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
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				
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:

(1) In mode 1, the Melements distributed to the sending zone assigned by 0# station are M1400﹣M1415. The 0# station PLC can write values to M1400–M1415, and other stations (1#-﹣31#) can directly read the values of M1400﹣M1415.

(2) In mode 2, the Melements distributed to the sending zone assigned by 0# station are M1400–M1431. The 0# station PLC can write values to M1400–M1431, and other stations (1#-﹣31#) can directly read the values of M1400﹣M1431.

- Distribution of D element on N:N multiple-layer network (layer 0)

Distribution of D elements in sending area	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:

In mode 6, the D elements distributed to the sending zone assigned by 0# station are D7700﹣D7701. The 0# station PLC can write values to D7700–D7701, and other stations (1#-–15#) can read directly the value of D7700–D7701.

- Distribution of D elements on N:N multiple-layer network (layer 1)

Distribution of D elements in sending area	Mode 10	Mode 11	Mode 12	Mode 13
D7732–D7733	#0	#0	#0	#0

Distribution of D elements in sending area	Mode 10	Mode 11	Mode 12	Mode 13
D7734–D7735	#1			
D7736–D7737	#2	#1		
D7738–D7739	#3			
D7740–D7741	#4	#2	#1	
D7742–D7743	#5			
D7744–D7745	#6	#3		
D7746–D7747	#7			
D7748–D7749	#8	#4	#2	#1
D7750–D7751	#9			
D7752–D7753	#10	#5		
D7754–D7755	#11			
D7756–D7757	#12	#6	#3	
D7758–D7759	#13			
D7760–D7761	#14	#7		
D7762–D7763	#15			

For example:

In mode 10, the D elements distributed to the sending zone assigned by 0# station (layer 1) are D7732–D7733. The 0# station PLC can write values to D7732–D7733, and other stations (1#-–15#) can read directly the value of D7732–D7733.

- Distribution of M elements on N:N multiple-layer network (layer 0)

Distribution of M elements in sending area	Mode 6	Mode 7	Mode 8	Mode 9
M1400–M1415	#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	
M1608–M1623	#13			
M1624–M1639	#14	#7		
M1640–M1655	#15			

For example:

In mode 6, the M elements distributed to the sending zone assigned by 0# station (layer 0) are M1400–M1415. The 0# station PLC can write values to M1400–M1415, and other stations (1#-–15#) can read directly the value of M1400–M1415.

- Distribution of M elements on N:N multiple-layer network (layer 1)

Distribution of M elements in sending area	Mode 10	Mode 11	Mode 12	Mode 13
M1656–M1671	#0	#0	#0	#0
M1672–M1687	#1			
M1688–M1703	#2	#1		
M1704–M1719	#3			
M1720–M1735	#4	#2	#1	
M1736–M1751	#5			

Distribution of M elements in sending area	Mode 10	Mode 11	Mode 12	Mode 13
M1752–M1767	#6	#3		
M1768–M1783	#7			
M1784–M1799	#8	#4	#2	#1
M1800–M1815	#9			
M1816–M1831	#10	#5		
M1832–M1847	#11			
M1848–M1863	#12	#6	#3	
M1864–M1879	#13			
M1880–M1895	#14	#7		
M1896–M1911	#15			

For example:

In mode 10, the M elements distributed to the sending zone assigned by 0# station (layer 1) are M1656–M1671. The 0# station PLC can write values to M1656–M1671, and other stations (1#-–15#) can read directly the value of M1656–M1671.

Note

Once the PLC is configured with the N:N communication protocol,D7700–D7763 and M1400–M1911 are used as the public resources for network data exchange. You need to pay attention to these elements when using them in the program!

10.5.5 Enhanced refresh mode

To support more elements sharing, the IVC series micro-PLCs offer modes 14–18. These modes are only applicable to the single-layer structure, or situations where there are many shared components. M components and D components have been expanded on the original basis (M1400-M1911 and D7500–D7755).

The M component areas (512) are shown in the following table:

Distribution of M elements	Mode 14	Mode 15	Mode 16	Mode 17	Mode 18
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		
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				
M1784-M1799	#24	#12	#6	#3	
M1800-M1815	#25				
M1816-M1831	#26	#13			
M1832-M1847	#27				
M1848-M1863	#28	#14	#7		

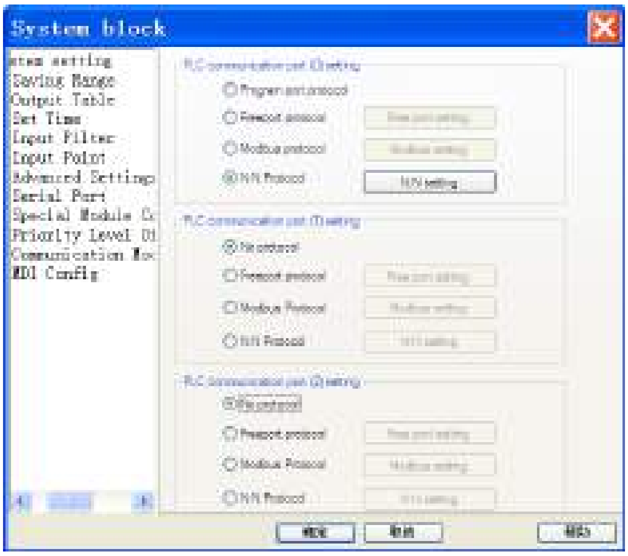
M1864-M1879	#29				
M1880-M1895	#30	#15			
M1896-M1911	#31				

The D element areas (256) are shown in the following table:

Distribution of D elements	Mode 14	Mode 15	Mode 16	Mode 17	Mode 18
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 setting

Selecting the **Communication Port** item in the System block, and selecting the **N:N protocol** in the **PLC communication port (0) setting** or **PLC communication port (1) setting** to enable the corresponding **N:N setting** button as shown below.



Clicking the **N:N setting** button to enter the **N:N Protocol** setting interface, as shown below.

As shown in the preceding figure, the N:N parameters are set through the system block. The **Station no.**shall be set starting from 0#, and set in order. Several PLCs cannot share the same station number. 0#station is used for starting and setting the whole network. The setting of **Max number of stations**, **Additional delay time**, **Retry times**, **Mode** can be realized through 0#station. For the stations with other station numbers, except that the baud rate and parity check shall be consistent with those of 0#station, they only need to set their own station number, as shown in the following figure.

The maximum number of inspection stations refers to the total number of PLCs used in the network. If you use a total of 6 PLCs, you need to set the maximum number of inspection stations to 6, and set the station number of these 6 PLCs to 0-5. If you want to add another two PLCs to the network without any interruption of the network, you can set the maximum number of inspection stations to 8 and the PLC stations to be added in the future to 6 and 7respectively. When 6 and 7 are connected to the network, they are automatically detected by N:N within one second and included into the data exchange with 0-5.

10.6 Several control strategies

10.6.1 Determination of the master station

0# station is a master station by default. Only 0#station can initialize and start the entire network. The settings about N:N, such as refresh mode, additional delay time, and retry times, needs to be configured only through the 0# station. When 0# stationmodifies the relevant configuration online and the standby master station takes over the network during the process of downloading the system blocks. When 0#station completes the downloading of the system block, the standby master station givesthe place of the master station to 0#station.

Master station strategy in the network: The station with the mini. station number acts as the master station.

10.6.2 Maximum number of inspection stations

When setting the max. number of the inspection stations, it is recommended to set the max. number of the inspection stations to the total number of PLCs included in the actual network, and set the station number starting from 0# in turn. When the max.

number of the inspection stations is set to N#, the network only manages the stations ranging from 0# to N-1#. In particular, when the max. number of the inspection stations set by the user is incorrect, that is, when the maximum number of inspection stations is smaller than the number of PLCs included in the actual 485 network, PLCs with station numbers that are greater than or equal to the max. number of the inspection stations cannot broadcast their data but can receive the broadcast data of the PLCs with station numbers that are less than the max. number of the inspection stations.

10.6.3 Multi-master and slave (M: N)

A network of multi-master and multi-slave structures can be builtby using N:N. The meaning of "master" and "slave" here is: "master" indicates a PLC that cannot only write its own M and D elements, but also read M and D elements of other stations; "slave" indicates a PLC that can only read M and D elements of other stations. Under the set maximum number of inspection stations (this number is also subject to the refresh mode), the PLC whose station number is less than the number of inspection stations can be used as the "master" while the PLC whose station number is greater than the number of inspection 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 the 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. There are no corresponding M and D elements for the slave station in these tables.

10.6.4 Examples of using N:N

There are 5 PLCs in total, and the station number ranges from 0# to 4#. You can select 3 for the refresh mode 3. If you want to store the sum of D100 in 0#PLC and D305 in 2#PLC in D500 of 4#PLC, you can program as follows:

Programming 0#: **MOV D100 D7700**

Programming 2#: **MOV D305 D7716**

Programming 4#: **ADD D7700 D7716 D500**

Description: This example show the N:N single-layer network. There are 5 PLC stations on the network and the refresh mode is set to 3. Each station can be distributed with 8 D elements and 64 M elements. The D elements distributed to 0# station ranges from D7700 to D7707, the ones distributed to 2# station ranges from D7716 to D7723, and the ones distributed to 4# station ranges from D7732 to D7739. Storing the D100 value of 0# station in a write common area D7700 distributed by the network, and the D305 value of 2# station in a write common area D7716 distributed by the network. Executing the sum operation of D7700 and D7716 in 4#PLC and storing the sum in the local element D500.

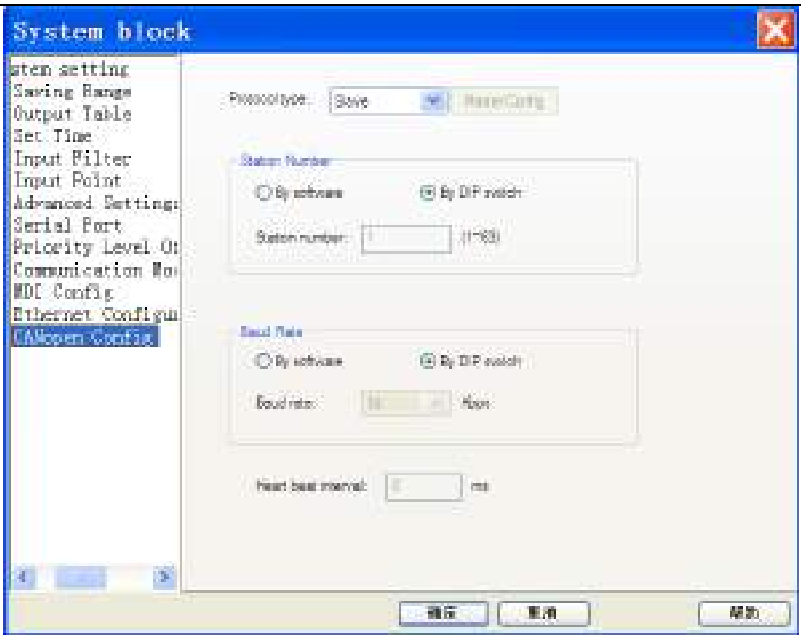
10.7 CAN communication

10.7.1 Introduction

CAN communication mainly supports CANopen standard protocol DS301 v4.02, CANopen master and slave station.

Protocol selection

Double-clicking the CANopen configuration tab in the system block to enter the configuration, and configure the protocol type, master and slave station according to the actual situation.



Function	Slave station	Master station
Supported protocol	DS301 V4.02	DS301 V4.02
Max. number of supported RxPDOs	4	64
Max. number of supported TxPDOs	4	64
Number of the slave nodes	/	32
Baud rate	1 Mbps 800 Kbps 500 Kbps 250 Kbps 125 Kbps 100 Kbps 50 Kbps 20 Kbps 10 Kbps	
Data exchange area	SD500-SD531	D elements Configurable

The station number and baud rate can be set in two ways. The software setting is directly input in the station number and baud rate fields; the dial setting is set on the PLC main module through the DIP switch. Bit1-bit6 of the DIP switch set the station number to 0, the CAN communication does not take effect; and bit7-bit8 of the DIP switch set the local baud rate.

10.7.2 CANopen indicator

LED display	CAN operation (green)	CAN error (red)
Off	None	No error
On	Working state	Bus off
Slow flashing (cycle: 0.8S)	Preoperation	
Single flash (period 1.2S)	Stop	Error
Double flash (cycle 1.6S)	None	Error control event (heartbeat)

10.7.3 Term explanation of CANopen

NMT: Network Management

It includes the network management services, application layer management, network state management, and node ID assignment management. The service mode is a master-slave communication mode: only one NMT master station, and one or more slave stations are in a CAN network. The master station is used to control the state of the slave station.

SDO: Server Data Object

Service data object, which can access the data in the slave device object dictionary through the indexes and sub-indexes. This is mainly used for the configuration process of the slave station. Each frame of SDO needs to be replied and confirmed.

PDO: Process data object

Process data object, mainly used to transmit the real-time data. For data transmission, the object is limited to 1 to 8 bytes. The transmission of PDO data is divided into synchronous and asynchronous modes. The PDO frame is the primary data interaction frame after the slave station is started.

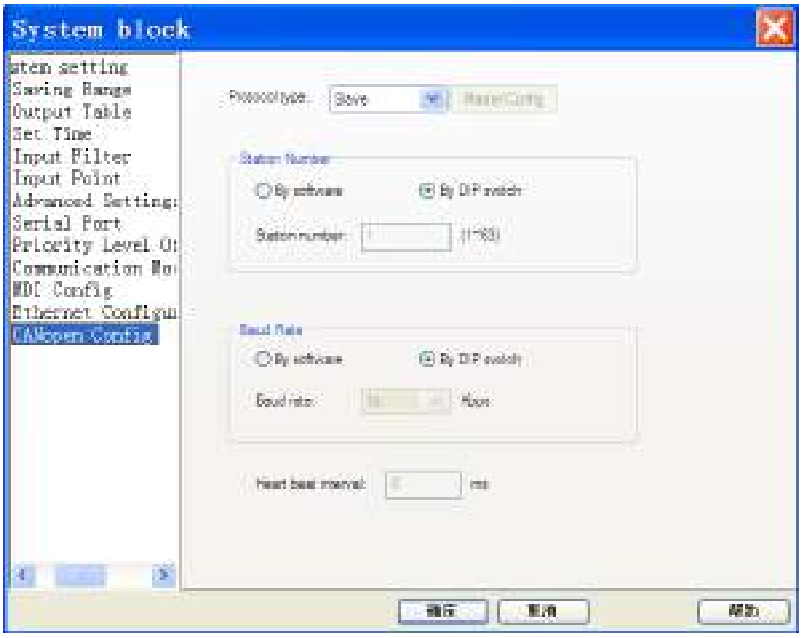
SYNC: Synchronous

The synchronization service, it adopts the master-slave communication mode, and the SYNC object is periodically sent by the master node. It executes the task simultaneously when receiving the SYNC object from the master station. 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 the communication object identifier of the CAN frame. The COB_ID is not equal to the slave station number, but it is associated with the slave station number by default.

10.7.4 Example of the slave configuration



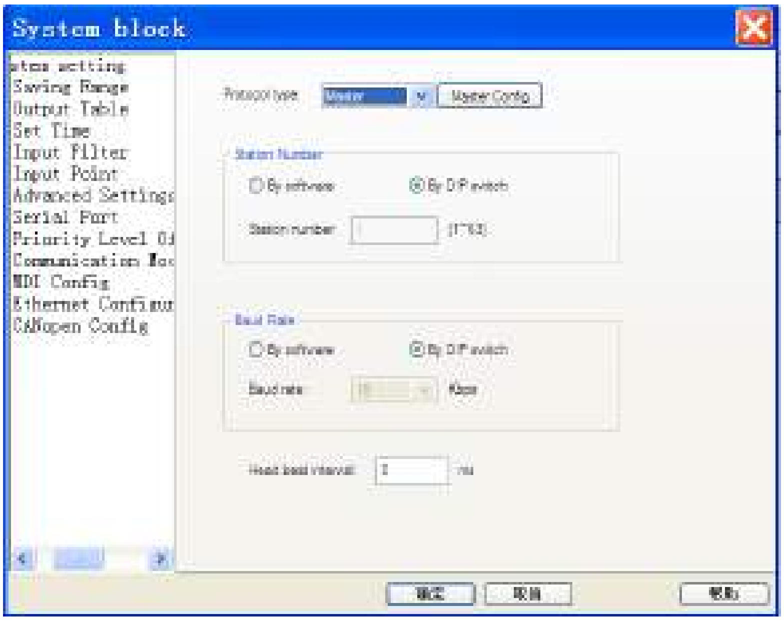
1. Double-clicking the CANopen configuration tab in the system block to enter the setup page, and selecting the slave from the protocol type option.

2. The station number is set by the software to 1#.

3. The baud rate is set by the software to 500Kbps.

4. Clicking the confirm button, and then the setting is completed and can be downloaded to the PLC together with the user program.

10.7.5 Example of the master configuration

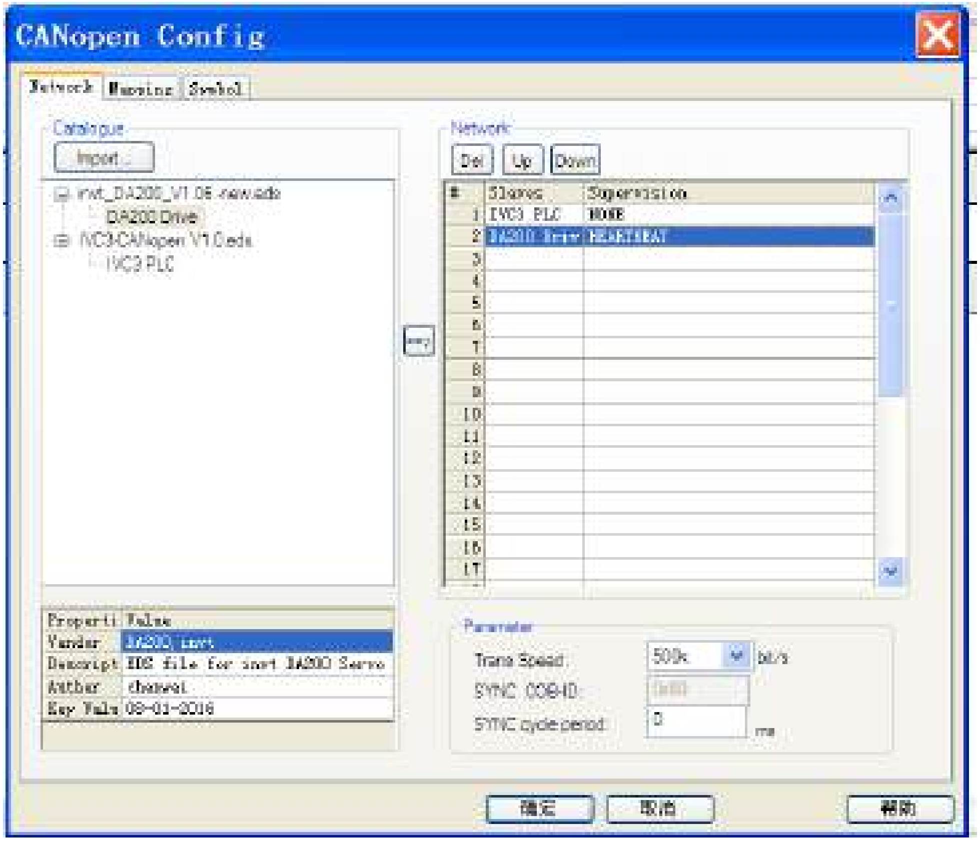


1. Double-clicking the CANopen configuration tab in the system block to enter the setup page, and selecting the slave from the protocol type option.

2. The station number is set by the software to 63#.

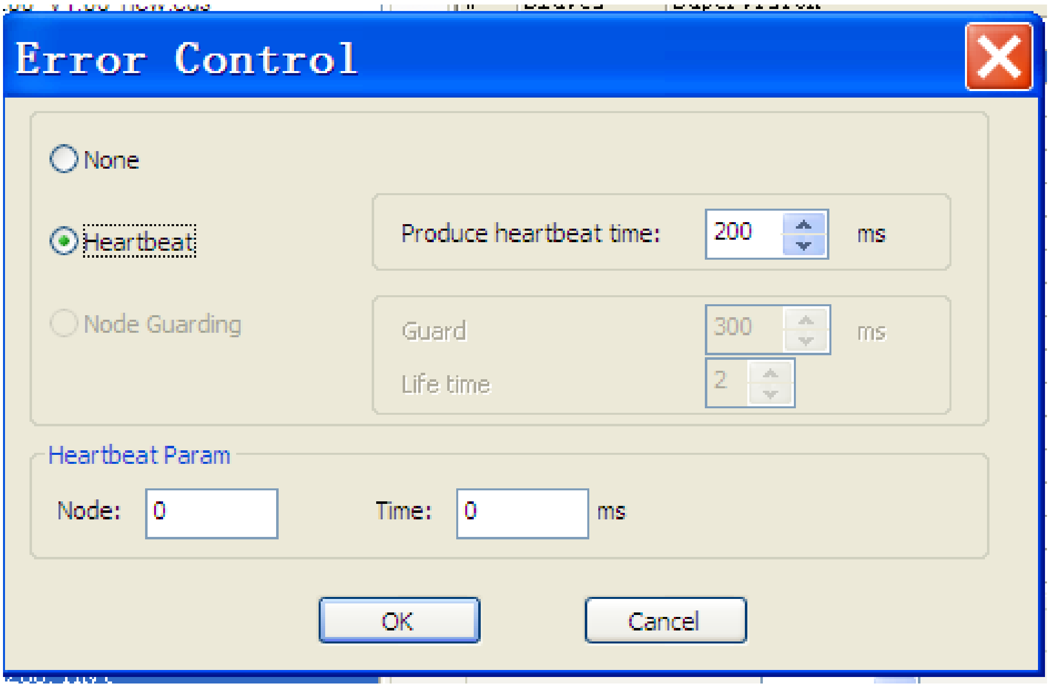
3. The baud rate is set by the software to 500Kbps.

4. Clicking the master configuration button to enter the master configuration page.



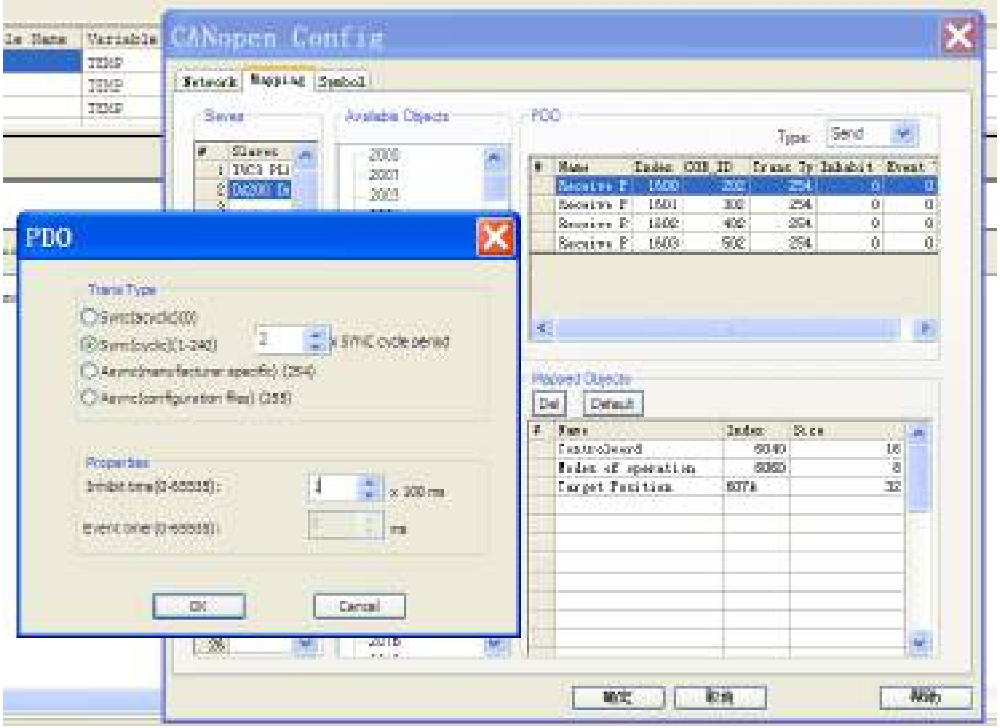
5. On the Network page, you need to choose to import the EDS file, and double-click the EDS file to join the network. Slave 1 is IVC3 while slave 2 is DA200. Transmission speed of the bus parameter is not set, synchronization message COB-ID is 0x80 by default which cannot be set. The synchronization message is set to 500MS in the corresponding period (the master station generates a synchronization message every 500MS).

6. Double-clicking2# stationDA2000, setting2# slave station to use the Heartbeat protocol. This setting requires the slave station to support the Heartbeat protocol. The master station configures that the slave station generates a heartbeat every 200MS, and at the same time the master station monitor the slave station. If the slave station does not run due to the fault, and the time is 1.5 times the heartbeat time of the slave station (1.5×200 MS), the master station reports an error(CAN ERR LED double flash)and 1# station does not monitor.

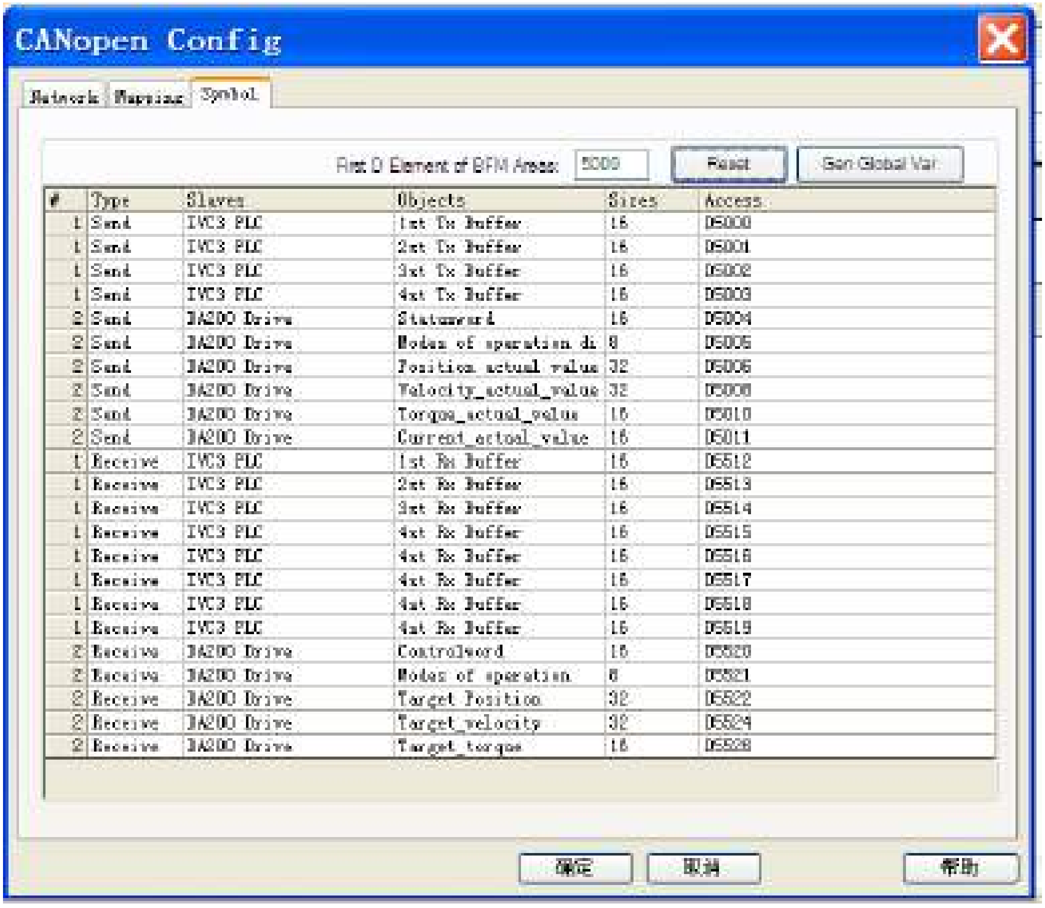


7. On the Map page setting, you can select DA200, and then select send button in Type of the PDO column. The PDO has two types: send and receive. Hereby, the send PDO refers to the slave station's sending data, that is, the slave station sends the data through this PDO to the receiver (this receiver is the master station). The receive PDO here refers to the slave station's receiving data, that is, the slave station receives the sender's data through this PDO (this sender is the master station).

8. Setting the transmission type and attributes of PDO. Synchronous non-cycle means that it sends the information only once after receiving the synchronization information while the synchronous cycle (1-240) means that it sends the information once when every n pieces of synchronization information are received. Asynchronization (manufacturer event) refers to sendinformation when the data changes while asynchronization (profile event) is determined according to the CANopen subprotocol.Suppression time refers to the mini. time interval between two transmissions of the same PDO, and the event timer can trigger some events periodically. In the following figure, we choose the synchronous cycle mode, and the cycle is 2 synchronization information.You can configure the PDO according to your actual needs.



9. On the Symbol page setting, the first D element of the BFM area is set to 5000, and then you can click the reset button.



10. Clicking the confirm button, and the setting is completed and can be downloaded to the PLC together with the user program.

11. Setting the station number and baud rate on the slave device, connecting the master-slave devices to the CANopen network, setting the terminal resistance, starting the network, and switching the master station PLC switch from STOP to RUN. If the network settings are correct, the CAN RUN LED is always on, and the CAN ERR is off. The master station receives the data sent by the slave station through D5000 – D5011, and sends data to the slave station through D5012-D5022.

10.8 Ethernet communication

10.8.1 Introduction of Ethernet

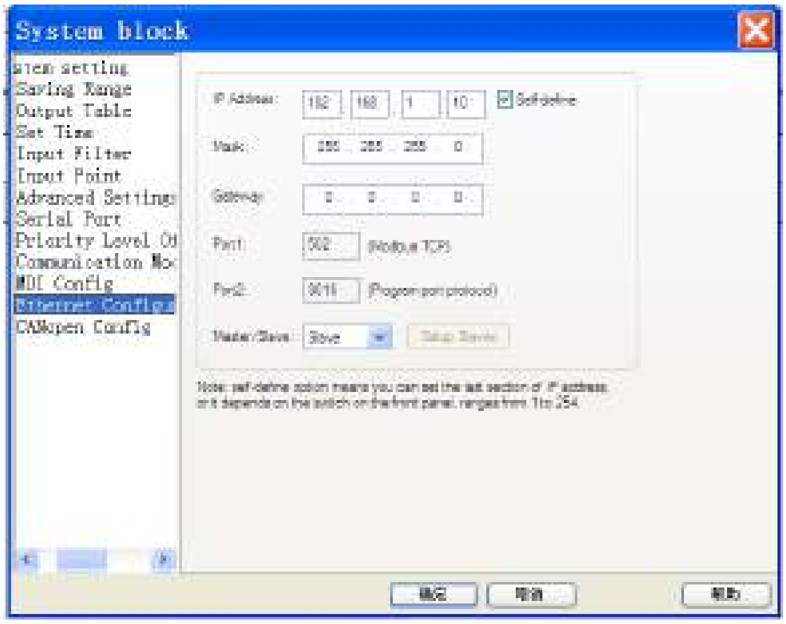
Main modules of the IVC3 series PLCs are configured with Ethernet interfaces, supporting 10M/100M adaptive rate and Modbus-TCP. The IVC3 series supports 16 connections ((the same connection is the same for IP and port number) for data exchange, and the same station can be used as the master and slave stations simultaneously.

10.8.2 Parameter settings of Ethernet

The default IP of the IVC3 communication is 192.168.1.10. This IP is defaulted after the PLC is factory formatted, and can directly communicate with the host computer and Modbus TCP client.

There are two ways to modify the IP. One is set by the host computer, and the other is set by the DIP switch.

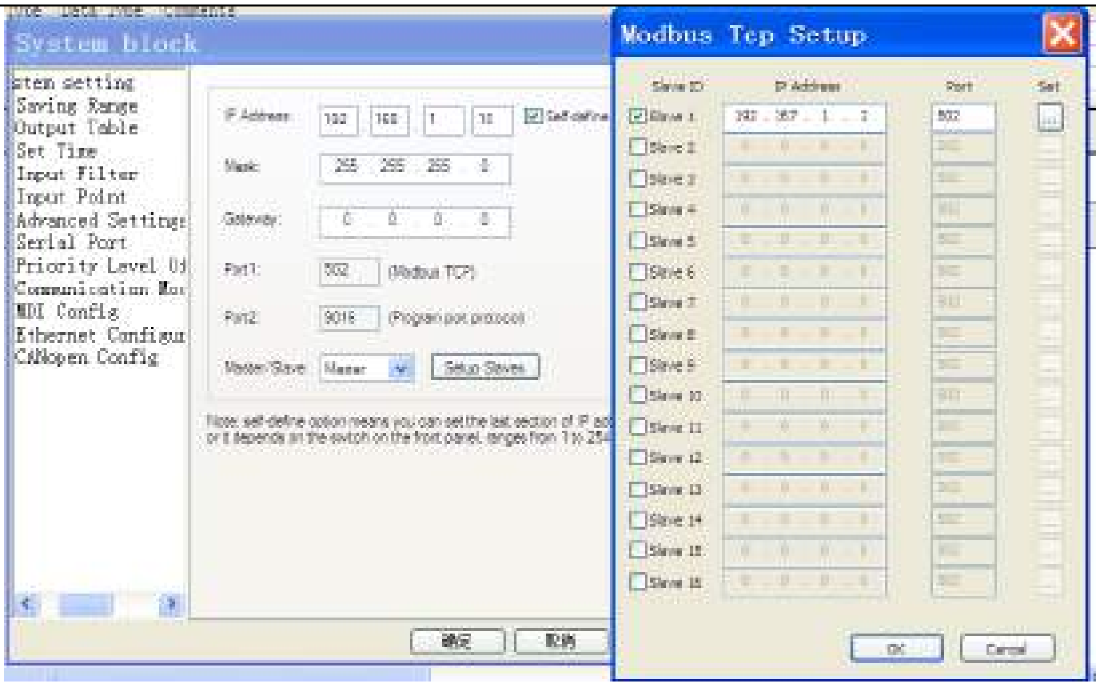
Double-clicking the Ethernet configuration tab in the system block to enter the Ethernet configuration page. The first three segments of the IP can only be configured by the software. As for the fourth segment, you need to select self-define button instead of DIP switch, and it is customized by the PC software configuration. Setting the mask and gateway while port 1 and port 2 are set by default and do not need to be modified. After that, you can click the confirm button, and the setting is download to the PLC.



10.8.3 Settings of Modbus TCP master station

On the Ethernet configuration page, you can select Master in "Master/Slave" item, and click Setup Slaves button to enter the settings.

Selecting the station number to be connected, entering the IP address, and the port number is 502 by default and generally not modified, and then click set.



Entering the slave instruction configuration table and clicking Add button to add a new input line.

No.: Self-added, and you have no need to manage it.

Function: There are read register, write register, read coil, and write coil.

Triggering conditions: Two modes: cycle and trigger, the cycle means to cyclically access the slave station, and the trigger means to access the slave station when the triggering element is ON.

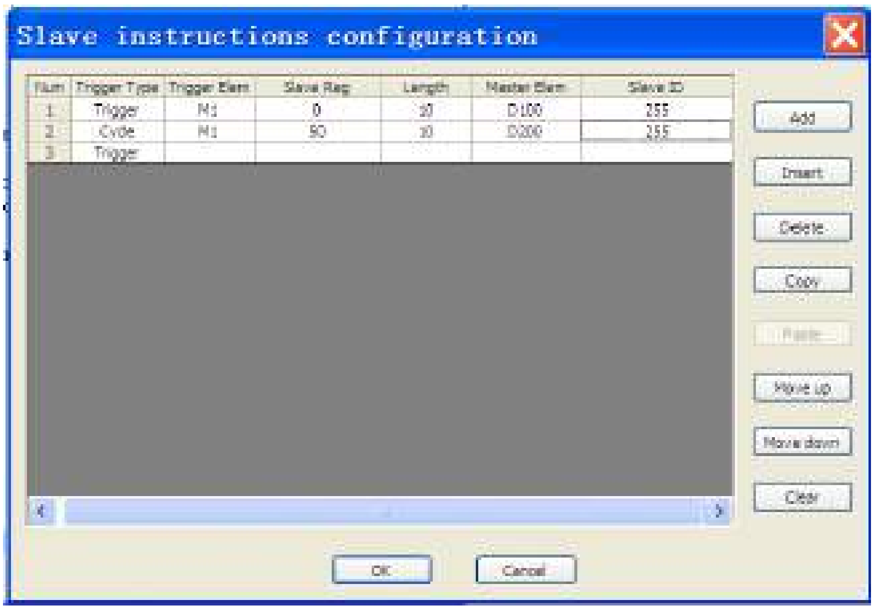
Triggering element: M and S elements are supported.

Register address of the slave station: It refers to the Modbus address (decimal) that needs to access the elements of the slave station. For details about the elements address of the IVC3 series PLC slave station, refer to section 10.4.6.

Data length: The length of the data to be accessed.

Element address of the master station: The start address of the master station buffer.

Station number: 255 by default.



No. 1: Reading 10 data (0-9) starting from the register address 0 of the slave station, and storing these data in the units ranging from D100 to D109 of the master station when M1 is ON.

No. 2: 10 bytes of data starting from the master station D200 (D element is two sections D200-D204) are written to 10 data lengths (50-59) starting from the register address 50 of the slave station in each cycle.

Note: There is a max. length limit in each configuration item of Modbus.

Read register	123
Write register	121
Read coil	1968
Write coil	1936

Once the master configuration of Ethernetis downloaded, the connection is automatically created and the state of the connection is managed without user intervention.

Chapter 11 Positioning function guide

11.1 Positioning control system

11.1.1 Absolute position system

The absolute position system obtains the absolute position data of the servo motor on the stroke by detecting the current position data of the servo motor encoder and total number of running turns. According to this principle, an absolute coordinate system can be established on the mechanical stroke. The following figure is a functional block diagram of an absolute position system.

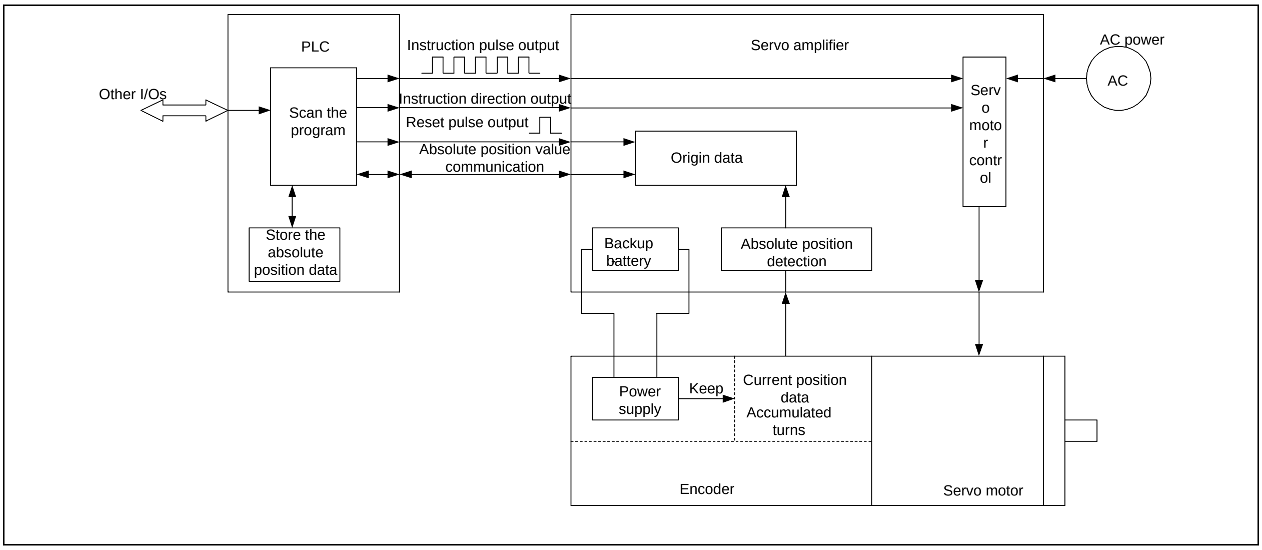


Figure 11-1 Absolute position system function diagram

In the above figure, unlike the conventional incremental encoder, the current position data and accumulated turns of the encoder of the absolute position system can be maintained. These data can be maintained by the power supply of a backup battery. Even in the event of a power failure, the servo drive can obtain the current absolute position data through being powered on again.

After the PLC is powered on, the absolute position data can be obtained from the servo drive through communication or other special methods, and the stroke coordinate position is determined. The PLC uses the positioning instruction to control the servo drive and the motor, thereby achieving the precise positioning on the stroke, and automatically refreshing the absolute position data automatically. In this way, a working system based on the absolute position coordinates can be formed.

The following diagram is a simplified mechanical example of an absolute position system built on the IVC series PLC positioning instructions.

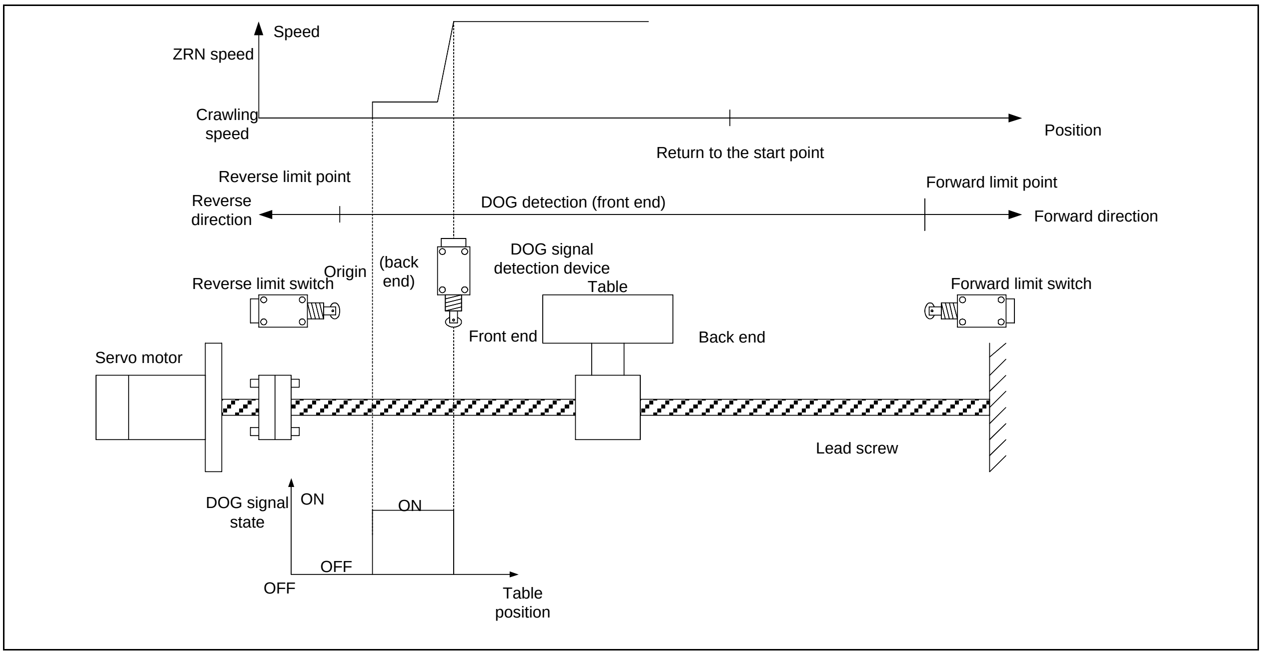


Figure 11-2 Mechanical example diagram of an absolute position system

The system drives the lead screw with a servo motor to drive the table. The position of the table in the stroke is detected by an absolute encoder. During the zero return process,when the DOG signal device detects the front end of the table (set), the servo motor decelerates to the crawling speed; when the DOG signal device detects that the rear end of the table (reset), it is an original position arriving signal, and the PLC stops the high-speed pulse output. The forward limit switch and the reverse limit switch must be set. Because the ZRN instruction is incapable of searching for DOG signal automatically, it is required to start the ZRN operation further than the front end of DOG detection device. You can perform the inching operation to manually adjust the position of the table through the design and programming.

11.1.2 Positioning control system

The positioning control system can be divided into open-loop control system, semi-closed loop control system and closed-loop control system according to different control modes.

The open-loop control system is a control system in which the regulating system does not accept the control of the feedback, and only controls the output, also known as the no feedback control system. The open-loop control system is mostly composed of a controller, stepping driver, and stepping motor. The controller sends a pulse instruction to the stepping driver, and then the stepping motor drives the table to move a certain distance. This kind of system is relatively simple, stable in operation, and easy to grasp, but it cannot detect errors, nor can it correct errors. Its control accuracy and the performance of suppressing interference are relatively poor, and it is sensitive to the changes in system parameters. Therefore, it is generally only used in applications where the external influence is not considered, or the inertia is small, or the accuracy is not high.

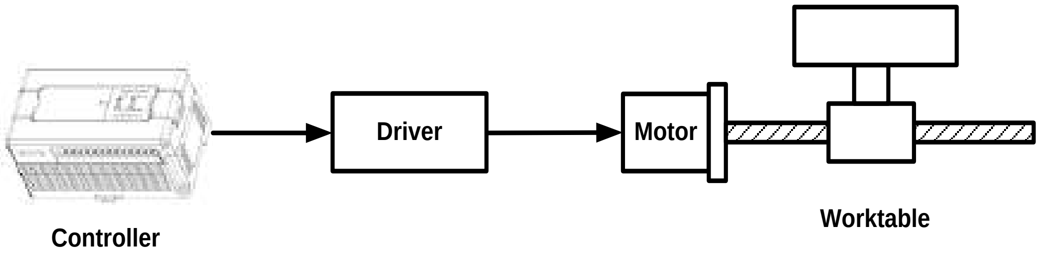


Figure 11-3 Open-loop control system diagram

The closed-loop control system is an automatic control system in which a closed loop consists of a signal forward path and feedback path, also known as a feedback control system. The closed-loop control system generally consists of a controller, servo drive, servo motor, detector, etc. The system automatically detects the actual displacement of the table and feeds back to the controller for the closed-loop control. This kind of system has the high positioning accuracy, but the system is complicated, and difficult to debug and maintain, and its price is relatively expensive. It is mainly used for the high-precision applications and large CNC machines.

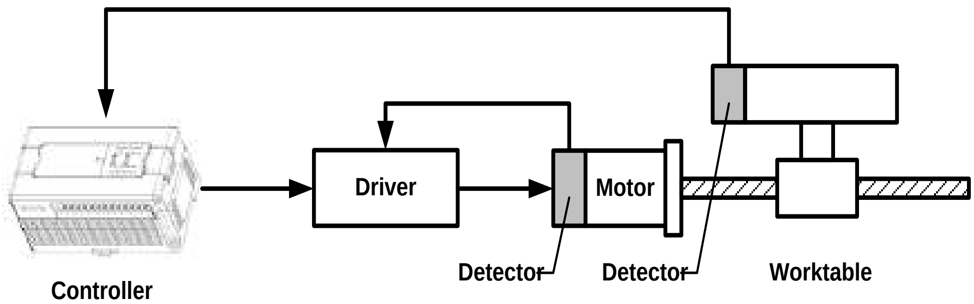


Figure 11-4 Closed-loop control systemdiagram

The semi-closed loop control system works similarly to the closed-loop control system except that the detector is not mounted on the table but mounted on the shaft of the servo motor. The accuracy, speed and dynamic characteristics of this system are superior to the open-loop control system. Its complexity and cost are lower than that of the closed-loop control system. They are mainly used for medium-precision applications and most of small and medium-sized CNC machines.

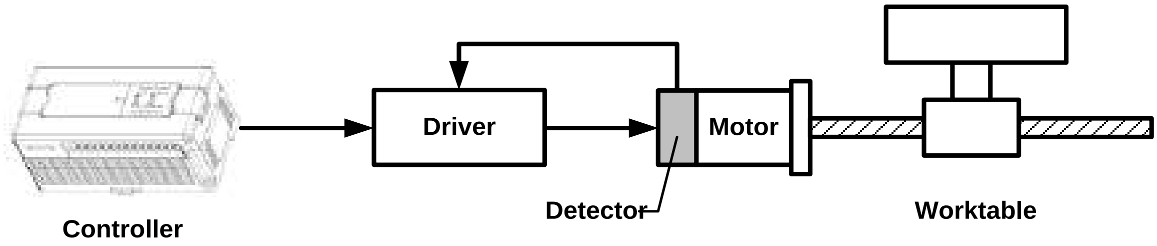


Figure 11-5Semi-closed loop control system diagram

11.1.3 Process of positioning control

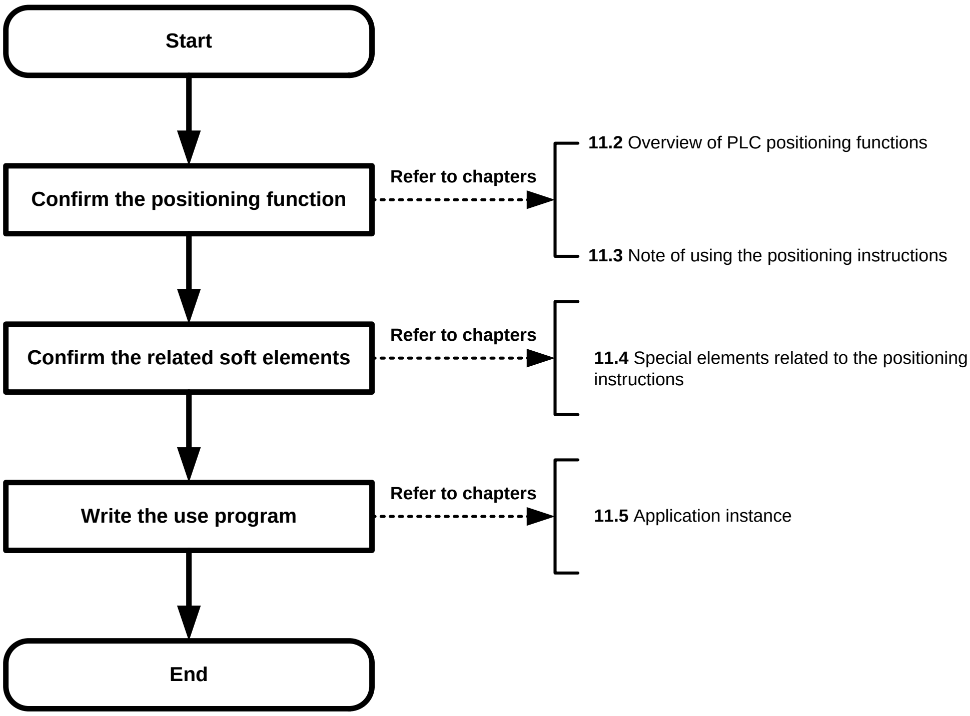


Figure 11-6 Process of positioning control

11.2 Overview of the IVC series PLC positioning functions

The positioning functions supported by IVC series PLCs include simple pulse output positioning, two-axis linear and arc trajectory interpolation, and inter-axis synchronous motion control, which can be widely used in positioning control systems to control the stepping and servo drives of various brands. The absolute position data can be obtained by the means provided by the corresponding servo drive.

Table 11-1 Overview of positioning functions of the IVC series PLC main module

Name	IVC3	IVC2L	IVC1L	IVC1	IVC1S
Controlled axes	8-axis	2-axis	3-axis	2-axis	2-axis
Max. output frequency	200 kHz	100 kHz	100 kHz	100 kHz	100 kHz
Pulse output mode	Open collector	Open collector	Open collector	Open collector	Open collector
Pulse output type	Pulse+direction, positive and negative pulses	Pulse+direction	Pulse+direction	Pulse+direction	Pulse+direction
Acceleration and deceleration treatment	Trapezoidal acceleration/deceleration, triangle acceleration and deceleration disabling	Trapezoidal acceleration/deceleration	Trapezoidal acceleration/deceleration	Trapezoidal acceleration/deceleration	Trapezoidal acceleration/deceleration
Interpolation function	2-axis linear interpolation and	—	—	—	—

	circular interpolation				
Synchronization function	Position synchronization, electronic gear	—	—	—	—
Absolute position detection	ABS instruction read		—	ABS instruction read	—
Positioning range	-2,147,483,648–+2,147,483,647 (pulse)				

*Note: For IVC3 series PLCs, only interpolation instruction supports the positive and negative pulse output modes.

When connecting to the servo, you need to set the input signal of the servo amplifier to the negative logic mode. The pulse output form is defined as follows:

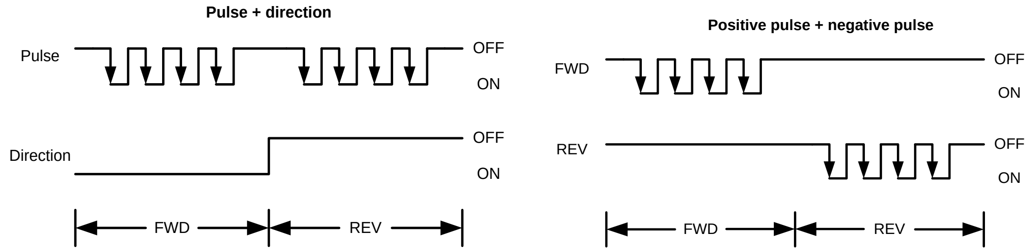
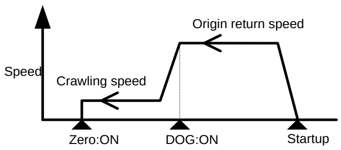
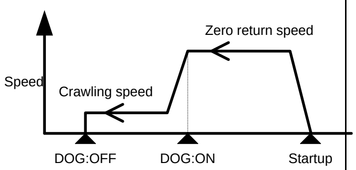
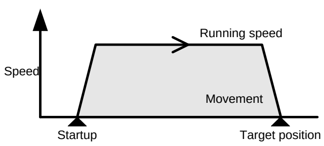
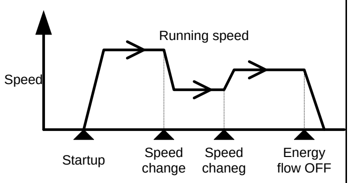
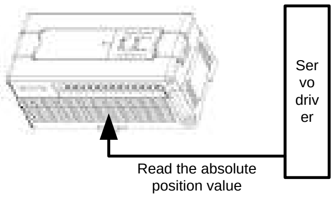
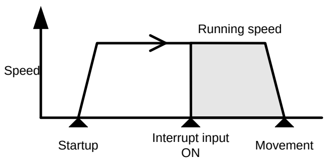
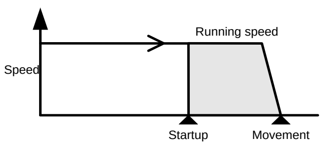
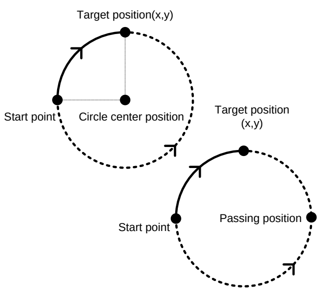
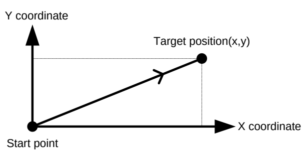


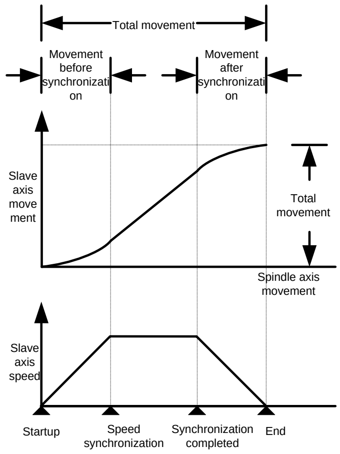
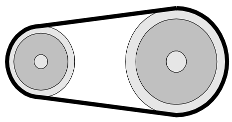
Figure 11-7 Definition of pulse output form

Note: The high-speed I/O instruction can also be used as a pulse output, but only for the output control of the pulse signal, and does not control the direction signal. When these high-speed I/O instructions are used, the corresponding position SD elements are subjected to the accumulation processing in the forward direction. When you want to drive the servo in forward motion, you needs to program and set the servo direction signal to ON, and set to OFF in reverse motion.

Table 11-2 Positioning function list of IVC series PLC main modules

Name	Action	Content	IVC3	IVC 2L	IVC 1	IVC 1L	IVC1S
DSZR		It acts according to the specified zero return speed and can automatically search for the DOG signal. Decelerating to the crawling speed once the DOG is detected (the sensor of DOG is ON). Stopping when a zero signal is inputted, the ZRN is completed.	•	•		•	
ZRN		It acts according to the specified zero return speed and decelerates to the crawling speed once the DOG is detected (the sensor of DOG is ON). Stopping when the sensor of DOG is OFF, and the ZRN is completed.	•	•	•	•	
DRVI		It acts according to the set running speed, stops at the target position, and uses relative coordinates for the position.	•	•	•	•	•

Name	Action	Content	IVC3	IVC 2L	IVC 1	IVC 1L	IVC1S
DRVA		It acts according to the set running speed, stops at the target position, and uses absolute coordinates for the position.	●	●	●	●	
PLSV		It acts according to the set running speed. If the running speed changes, it runs at the new speed; if the energy flow becomes invalid, the pulse output stops. When there is acceleration/deceleration, the acceleration/deceleration is executed when the speed is changed.	●	●	●	●	
ABS		Reading the current absolute position data from the servo drive.	●	●	●		
DVIT		It acts according to the set running speed. If the interrupt input is ON, it runs the specified number of pulses and then decelerates to stop.		●		●	
STOPDV		When a positioning operation is being executed, if this instruction is started, it runs the specified number of pulses and then decelerates to stop.	●				
CW		It moves to the target position along the circular trajectory in the clockwise direction at the specified linear speed.	●				
CCW		It moves to the target position along the circular trajectory in the anti-clockwise direction at the specified linear speed.	●				
LIN		It moves to the target position along a linear trajectory according to the specified vector speed.	●				

Name	Action	Content	IVC3	IVC 2L	IVC 1	IVC 1L	IVC1S
MOVELINK		The slave axis moves with the spindle axis and keeps pace with it within a specified position range, supporting the acceleration/deceleration control during the transition process before and after the synchronization.	•				
GEARBOX		It controls the slave axis to move with the spindle axis according to a certain electronic gear ratio.	•				

The positioning instruction and high-speed instruction output a controllable pulse at the high-speed port according to the settings, and the output of the pulse has nothing to do with the scan cycle of the user program. For details about the usages of these instructions, refer tosection 6.10 "High-speed I/O instruction". In the program, using positioning instructions or high-speed instructions for different output ports can obtain independent high-speed pulse output at the corresponding output port.

11.3 Note of using the positioning instructions

When the positioning instruction or high-speed instruction is valid (including the output is completed), other operations on the same port are invalid. Other instructions have the correct output only when the high-speed pulse output instruction is invalid.

When there are multiple positioning instructions or high-speed instructions on the same port, the first valid instruction occupies the output port, and the latter valid one does not occupy the output port.

- Transistor output
  An IVC series PLC with transistor output is a must.
- Requirements for positioning instructions in programming
  Positioning instructions can be used repeatedly in the program, but you need to pay attention to:

  1. You cannot drive and use other positioning or high-speed pulse output instructions on the same high-speed pulse output point at the same time. A high-speed pulse output point can only be driven by a positioning instruction (or high-speed instruction) at any one time.

  2. When the energy flow of a positioning instruction is disconnected, the instruction can be driven again only when the energy flow is connected after one or more PLC scan cycles.
- The main points for using the high-speedand positioning instructions at the same time
  Considering from the function implementation, it is recommended to use the positioning instruction to replace these high-speed pulse output instructions (PLSY, PLSR, and PLS), so as to complete the automatic update of the absolute position SD elements.

  The absolute position SD elements can be used to store and update the current absolute position after the positioning instruction is used. The automatic increase and decrease of the absolute position SD element value is determined based on the change value of the output pulse accumulation SD element and the running direction when the positioning instruction is called, so they are linked. Do not perform the write operation on the the output pulse accumulation SD element when using the positioning instruction. Otherwise, the data of the absolute position SD element may be disordered.

If it is necessary to use both positioning instructions and other high-speed pulse output instructions (PLSY, PLSR, and PLS) at the same time, you need to write the PLC program so that the data of absolute position SD elements in the absolute position register can be correctly updated.

- Restrictive conditions for the actual output frequency of the positioning instruction

When the positioning instruction is executed, the min. 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 max.speed, set by SD85, SD86, etc.; T indicates the acceleration/deceleration time, set by SD87, etc., and the unit is ms. The calculation result $F_{\min\_acc}$ is the min. output frequency limit value.

If the output frequency specified in the positioning instruction is F, there are three cases of the actual output frequency as follows.

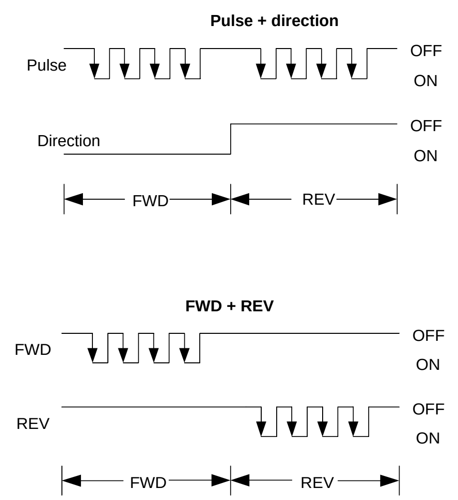
- When F is less than the base frequency or greater than the max. frequency $F_{\max}$, there is actually no output.
- When F is less than $F_{\min\_acc}$, and the actual output is $F_{\min\_acc}$.
- When F is greater than or equal to $F_{\min\_acc}$, and less than or equal to $F_{\max}$, the output is F.

11.4 Special elements related to the positioning instructions

11.4.1 Related soft elements of the IVC3 series positioning instructions

The definition and assignment of the output axes of the IVC3 series PLCs are shown in the table below.

Table 11-3 Definition of the IVC3-1616MAT output axis

Output axis	Supported mode	Definition of the output points		Output mode definition
0	Pulse + direction	Pulse	Y0	
		Direction	Any output points except Y0	
1	Pulse + direction	Pulse	Y1	
		Direction	Any output points except Y1	
2	Pulse + direction	Pulse	Y2	
		Direction	Any output points except Y2	
3	Pulse + direction	Pulse	Y3	
		Direction	Any output points except Y3	
4	Pulse + direction	Pulse	Y4	
		Direction	Any output points except Y4	
5	Pulse + direction	Pulse	Y5	
		Direction	Output points except Y5 are not limited	
6	Pulse + direction	Pulse	Y6	
		Direction	Output points except Y6 are not limited	
7	Pulse + direction	Pulse	Y7	
		Direction	Output points except Y7 are not limited	

Note

When using any one of the output axes to connect the servo, you need to consider the pairing of the output points. All the output single-axis positioning can only use the "pulse + direction" mode. Only the interpolation instruction can use the "FWD + REV" mode. Only one group of interpolation instructions can be executed at one time. One group of interpolation instructions occupies 4 ports: Y0, Y1, Y2, Y3, among which Y0 and Y1 are X-axes, and Y 2 and Y3 are Y-axes.When there are double pulses, Y0 is a positive pulse and Y1 is a negative pulse while Y2 is a positive pulse and Y3 is a negative pulse. In the "pulse + direction" mode, you need to note that when the direction signal is selected, the output point cannot be used for other purposes at the same time. For example, you cannot do not define the same output point as that of the pulse or direction signal of other output axes.

- Output channel control and monitoring of the output axis 0 (the axis corresponding to Y0)

Address	Name	Function	R/W
SM80	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y0 and reset to open the output function	R/W
SM82	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y0. It is ON when busy, and OFF when ready.	R
SM63	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y0 and reset to disable the pulse output completion interrupt.	R/W
SM280	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y0. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM281	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, the Y element Y (N) corresponding to the value N in SD206 is used to indicate the clear signal. When it is reset, Y10 is defined as the clear signal by default.	R/W
SM282	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM283	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y0. 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.	R/W
SM284	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y0. 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
SM285	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM286	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM288	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y0 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM280 is set, the default clear signal Y10 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 0 (the axis corresponding to Y0)

Address	Name	Function	R/W
SD50	The sum (MSB) of accumulated pulses of the output axis 0	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD200–SD201 is calculated and updated based on SD50–SD51 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD200.	R/W
SD51	The sum (LSB) of accumulated pulses of the output axis 0		R/W
SD200	The absolute position current value (MSB) of the output axis 0		R/W
SD201	The absolute position current value (LSB) of the output axis 0		R/W
SD202	Max. speed (MSB) of the output axis 0	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD203	Max. speed (LSB) of the output axis 0		R/W

Address	Name	Function	R/W
SD204	Base speed of the output axis 0	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD205	Acceleration/deceleration time of the output axis 0	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD206	Reset-to-zero signal soft element specified for output axis 0	When SM281 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD207	The crawling speed of the output axis 0	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD208	The zero return speed (MSB) of the output axis 0	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD209	The zero return speed (LSB) of the output axis 0		R/W
SD56	Segment number of the envelope output of output axis 0	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD202–SD205 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 1 (the axis corresponding to Y1)

Address	Name	Function	R/W
SM81	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y1 and reset to open the output function	R/W
SM83	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y1. It is ON when busy, and OFF when ready.	R
SM64	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y1 and reset to disable the pulse output completion interrupt.	R/W
SM290	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y1. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM291	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the Y element Y (N) corresponding to the value N in SD230 is used to indicate the clear signal. When it is reset, Y11 is defined as the clear signal by default.	R/W
SM292	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM293	FWD limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y1. 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.	R/W
SM294	REV limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y1. 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
SM295	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM296	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM297	Logic reversal of the interrupt signal	It is applicable to DVIT, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the interrupt signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the interrupt signal is ON).	R/W
SM298	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y1 when the DSZR/DVIT instruction is executed. It is ON when busy, and OFF when	R/W

		ready.	
SM299	The element specified by the Y1 interrupt signal is valid	It is applicable to DVIT, and acts on the axis corresponding to Y1. When it is set, the value X (N) in SD240 is used to indicate the interrupt input signal. When it is reset, X1 is defined as the interrupt input signal by default.	R/W
SM260	The element specified by the module interrupt signal is valid	It is applicable to DVIT, and acts on the axes corresponding to Y0 and Y1. When it is set, SM289, SM299, and SD240 can be used to designate the interrupt input signal. When it is reset, the designation of the interrupt input signal is disabled.	R/W
Note: When the clear valid signal SM290 is set, the default clear signal Y11 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register ofthe output axis 1 (the axis corresponding to Y1)

Address	Name	Function	R/W
SD52	The sum (MSB) of accumulated pulses of the output axis 1	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD210–SD211 is calculated and updated based on SD52–SD53 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD210.	R/W
SD53	The sum (LSB) of accumulated pulses of the output axis 1		R/W
SD210	The absolute position current value (MSB) of the output axis 1		R/W
SD211	The absolute position current value (LSB) of the output axis 1		R/W
SD212	Max. speed (MSB) of the output axis 1	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD213	Max. speed (LSB) of the output axis 1		R/W
SD214	Base speed of the output axis 1	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD215	Acceleration/deceleration time of the output axis 1	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD217	The crawling speed of the output axis 1	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD218	The zero return speed (MSB) of the output axis 1	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD219	The zero return speed (LSB) of the output axis 1		R/W
SD230	Reset-to-zero signal soft element specified for output axis 1	When SM291 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD240	Soft element specified by the interrupt signal of the output axis	When SM299 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the interrupt signal.	R/W
Note 1: SD212–SD215 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise the base speed is automatically assigned to 1/10 of the max. speed. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 2 (the axis corresponding to Y2)

Address	Name	Function	R/W
SM262	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y2 and reset to open the output function	R/W
SM272	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y2. It is ON when busy, and OFF when ready.	R
SM72	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y2 and reset to disable the pulse output completion interrupt.	R/W
SM320	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y2. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM321	The element specified by the clear	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is	R/W

	signal is valid	set, the Y element Y (N) corresponding to the value N in SD326 is used to indicate the clear signal. When it is reset, Y12 is defined as the clear signal by default.	
SM322	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM323	FWD limit	It is applicable to DSZR, and acts on the axis 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.	R/W
SM324	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y2. 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
SM325	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM326	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM328	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y2 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM280 is set, the default clear signal Y12 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 2 (the axis corresponding to Y2)

Address	Name	Function	R/W
SD160	The sum (MSB) of accumulated pulses of the output axis 2	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD320–SD321 is calculated and updated based on SD160–SD161 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD320.	R/W
SD161	The sum (LSB) of accumulated pulses of the output axis 2		R/W
SD320	The absolute position current value (MSB) of the output axis 2		R/W
SD321	The absolute position current value (LSB) of the output axis 2		R/W
SD322	Max. speed (MSB) of the output axis 2	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD323	Max. speed (LSB) of the output axis 2		R/W
SD324	Base speed of the output axis 2	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD325	Acceleration/deceleration time of the output axis 2	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD326	Reset-to-zero signal soft element specified for output axis 2	When SM321 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD327	The crawling speed of the output axis 2	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD328	The zero return speed (MSB) of the output axis 2	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD329	The zero return speed (LSB) of the output axis 2		R/W
SD252	Segment number of the envelope output of output axis 2	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD322–SD325 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 3 (the axis corresponding to Y3)

Address	Name	Function	R/W
SM263	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y3 and reset to open the output function	R/W
SM273	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y3. It is ON when busy, and OFF when ready.	R
SM73	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y3 and reset to disable the pulse output completion interrupt.	R/W
SM330	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y3. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM331	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, the Y element Y (N) corresponding to the value N in SD336 is used to indicate the clear signal. When it is reset, Y13 is defined as the clear signal by default	R/W
SM332	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM333	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y3. 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.	R/W
SM334	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y3. 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
SM335	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM336	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM338	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y3 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM330 is set, the default clear signal Y13 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 3 (the axis corresponding to Y3)

Address	Name	Function	R/W
SD162	The sum (MSB) of accumulated pulses of the output axis 3	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD330–SD331 is calculated and updated based on SD162–SD163 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD330.	R/W
SD163	The sum (LSB) of accumulated pulses of the output axis 3		R/W
SD330	The absolute position current value (MSB) of the output axis 3		R/W
SD331	The absolute position current value (LSB) of the output axis 3		R/W
SD332	Max. speed (MSB) of the output axis 3	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD333	Max. speed (LSB) of the output axis 3		R/W
SD334	Base speed of the output axis 3	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD335	Acceleration time of the output axis 3	Acceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD263	Deceleration time of the output axis 3	Deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W

SD336	Reset-to-zero signal soft element specified for output axis 3	When SM331 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD337	The crawling speed of the output axis 3	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD338	The zero return speed (MSB) of the output axis 3	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD339	The zero return speed (LSB) of the output axis 3		R/W
SD253	Segment number of the envelope output of output axis 3	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD332–SD335 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 4 (the axis corresponding to Y4)

Address	Name	Function	R/W
SM264	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y4 and reset to open the output function	R/W
SM274	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y4. It is ON when busy, and OFF when ready.	R
SM74	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y4 and reset to disable the pulse output completion interrupt.	R/W
SM340	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y4. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM341	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y4. When it is set, the Y element Y (N) corresponding to the value N in SD346 is used to indicate the clear signal. When it is reset, Y14 is defined as the clear signal by default.	R/W
SM342	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y4. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM343	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y4. 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.	R/W
SM344	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y4. 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
SM345	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y4. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM346	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y4. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM348	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y4 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM340 is set, the default clear signal Y14 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 4 (the axis corresponding to Y4)

Address	Name	Function	R/W
SD164	The sum (MSB) of accumulated pulses of the output axis 4	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD340–SD341 is calculated and updated based on SD164–SD165 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute	R/W
SD165	The sum (LSB) of accumulated pulses of the output axis 4		R/W

SD340	The absolute position current value (MSB) of the output axis 4	position data (32-bit long integer) needs to be put into SD340.	R/W
SD341	The absolute position current value (LSB) of the output axis 4		R/W
SD342	Max. speed (MSB) of the output axis 4	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD343	Max. speed (LSB) of the output axis 0		R/W
SD344	Base speed of the output axis 0	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD345	Acceleration/deceleration time of the output axis 0	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD346	Reset-to-zero signal soft element specified for output axis 4	When SM341 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD347	The crawling speed of the output axis 4	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD348	The zero return speed (MSB) of the output axis 4	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD349	The zero return speed (LSB) of the output axis 4		R/W
SD254	Segment number of the envelope output of output axis 4	It is applicable to PLS, and used to detect the current segment number of the envelope output	R

Note 1: SD342–SD345 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction.

Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed.

- Output channel control and monitoring of the output axis 5 (the axis corresponding to Y5)

Address	Name	Function	R/W
SM265	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y5 and reset to open the output function	R/W
SM275	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y5. It is ON when busy, and OFF when ready.	R
SM75	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y5 and reset to disable the pulse output completion interrupt.	R/W
SM350	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y5. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM351	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y5. When it is set, the Y element Y (N) corresponding to the value N in SD356 is used to indicate the clear signal. When it is reset, Y15 is defined as the clear signal by default.	R/W
SM352	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y5. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM353	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y5. 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.	R/W
SM354	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y5. 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
SM355	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y5. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM356	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y5. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W

Address	Name	Function	R/W
SM358	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y5 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM350 is set, the default clear signal Y15 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 5 (the axis corresponding to Y5)

Address	Name	Function	R/W
SD166	The sum (MSB) of accumulated pulses of the output axis 5	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD350–SD351 is calculated and updated based on SD166–SD167 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD350.	R/W
SD167	The sum (LSB) of accumulated pulses of the output axis 5		R/W
SD350	The absolute position current value (MSB) of the output axis 5		R/W
SD351	The absolute position current value (LSB) of the output axis 5		R/W
SD352	Max. speed (MSB) of the output axis 5	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD353	Max. speed (LSB) of the output axis 5		R/W
SD354	Base speed of the output axis 5	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD355	Acceleration/deceleration time of the output axis 5	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD356	Reset-to-zero signal soft element specified for output axis 5	When SM351 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD357	The crawling speed of the output axis 5	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD358	The zero return speed (MSB) of the output axis 5	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD359	The zero return speed (LSB) of the output axis 5		R/W
SD255	Segment number of the envelope output of output axis 5	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD352–SD355 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 6 (the axis corresponding to Y6)

Address	Name	Function	R/W
SM266	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y6 and reset to open the output function	R/W
SM276	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y6. It is ON when busy, and OFF when ready.	R
SM76	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y6 and reset to disable the pulse output completion interrupt.	R/W
SM360	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y6. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM361	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y6. When it is set, the Y element Y (N) corresponding to the value N in SD366 is used to indicate the clear signal. When it is reset, Y16 is defined as the clear signal by default.	R/W
SM362	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y6. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN	R/W

Address	Name	Function	R/W
		direction is the reverse direction.	
SM363	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y6. 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.	R/W
SM364	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y6. 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
SM365	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y6. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM366	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y6. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM368	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y6 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM360 is set, the default clear signal Y16 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 6 (the axis corresponding to Y6)

Address	Name	Function	R/W
SD168	The sum (MSB) of accumulated pulses of the output axis 6	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD360–SD361 is calculated and updated based on SD168–SD169 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD360.	R/W
SD169	The sum (LSB) of accumulated pulses of the output axis 6		R/W
SD360	The absolute position current value (MSB) of the output axis 6		R/W
SD361	The absolute position current value (LSB) of the output axis 6		R/W
SD362	Max. speed (MSB) of the output axis 6	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD363	Max. speed (LSB) of the output axis 6		R/W
SD364	Base speed of the output axis 6	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD365	Acceleration/deceleration time of the output axis 6	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD366	Reset-to-zero signal soft element specified for output axis 6	When SM361 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD367	The crawling speed of the output axis 6	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD368	The zero return speed (MSB) of the output axis 6	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD369	The zero return speed (LSB) of the output axis 6		R/W
SD256	Segment number of the envelope output of output axis 6	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD362–SD365 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 7 (the axis corresponding to Y7)

Address	Name	Function	R/W

SM267	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y7 and reset to open the output function	R/W
SM277	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y7. It is ON when busy, and OFF when ready.	R
SM77	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y7 and reset to disable the pulse output completion interrupt.	R/W
SM370	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y7. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM371	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y7. When it is set, the Y element Y (N) corresponding to the value N in SD376 is used to indicate the clear signal. When it is reset, Y17 is defined as the clear signal by default.	R/W
SM372	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y7. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM373	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y7. 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.	R/W
SM374	REV limit	It is applicable to DSZR, and acts 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
SM375	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y7. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM376	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y7. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM378	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y7 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM370 is set, the default clear signal Y17 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 7 (the axis corresponding to Y7)

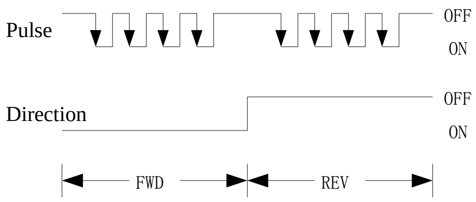
Address	Name	Function	R/W
SD170	The sum (MSB) of accumulated pulses of the output axis 7	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD370–SD371 is calculated and updated based on SD170–SD171 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD370.	R/W
SD171	The sum (LSB) of accumulated pulses of the output axis 7		R/W
SD370	The absolute position current value (MSB) of the output axis 7		R/W
SD371	The absolute position current value (LSB) of the output axis 7		R/W
SD372	Max. speed (MSB) of the output axis 7	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD373	Max. speed (LSB) of the output axis 7		R/W
SD374	Base speed of the output axis 7	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD375	Acceleration/deceleration time of the output axis 7	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD376	Reset-to-zero signal soft element specified for output axis 7	When SM371 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD377	The crawling speed of the output axis 7	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD378	The zero return speed (MSB) of the output axis 7	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W

Address	Name	Function	R/W
SD379	The zero return speed (LSB) of the output axis 7		R/W
SD257	Segment number of the envelope output of output axis 7	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD372–SD375 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

11.4.2 Related soft elements of the IVC2L series positioning instructions

The definition and assignment of the output axes of the IVC2L series PLCs are shown in the table below.

Table 11-4 Definition of the IVC2L series output axis

Output axis	Supported mode	Definition of the output points		Output mode definition
0	Pulse + direction	Pulse	Y0	**Pulse + direction** 
		Direction	Any output points except Y0	
1	Pulse + direction	Pulse	Y1	
		Direction	Any output points except Y1	

- Output channel control and monitoring of the output axis 0 (the axis corresponding to Y0)

Address	Name	Function	R/W
SM80	Y0 pulse output stop control	It can be set to stop the high-speed pulse output function of Y0 and reset to open the output function	R/W
SM82	Y0 pulse output monitoring	It is used to monitor the state of the high-speed output channel Y0. It is ON when busy, and OFF when ready.	R
SM63	Y0 pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y0 and reset to disable the pulse output completion interrupt.	R/W
SM280	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y0. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM281	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the Y element Y (N) corresponding to the value N in SD220 is used to indicate the clear signal. When it is reset, Y10 is defined as the clear signal by default.	R/W
SM282	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM283	FWD limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y0. 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.	R/W
SM284	REV limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y0. 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
SM285	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM286	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W

SM287	Logic reversal of the interrupt signal	It is applicable to DVIT, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the interrupt signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the interrupt signal is ON).	R/W
SM288	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y0 when the DSZR/DVIT instruction is executed. It is ON when busy, and OFF when ready.	R/W
SM289	The element specified by the Y0 interrupt signal is valid	It is applicable to DVIT, and acts on the axis corresponding to Y0. When it is set, the value X (N) in SD240 is used to indicate the interrupt input signal. When it is reset, X0 is defined as the interrupt input signal by default.	R/W
SM260	The element specified by the module interrupt signal is valid	It is applicable to DVIT, and acts on the axes corresponding to Y0 and Y1. When it is set, SM289, SM299, and SD240 can be used to designate the interrupt input signal. When it is reset, the designation of the interrupt input signal is disabled.	R/W
Note: When the clear valid signal SM280 is set, the default clear signal Y10 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 0 (the axis corresponding to Y0)

Address	Name	Function	R/W
SD50	The sum (MSB) of accumulated pulses of the output axis 0	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD200–SD201 is calculated and updated based on SD50–SD51 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD200.	R/W
SD51	The sum (LSB) of accumulated pulses of the output axis 0		R/W
SD200	The absolute position current value (MSB) of the output axis 0		R/W
SD201	The absolute position current value (LSB) of the output axis 0		R/W
SD202	Max. speed (MSB) of the output axis 0	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD203	Max. speed (LSB) of the output axis 0		R/W
SD204	Base speed of the output axis 0	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD205	Acceleration/deceleration time of the output axis 0	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD207	The crawling soeed of the output axis 0	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD208	The zero return speed (MSB) of the output axis 0	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD209	The zero return speed (LSB) of the output axis 0		R/W
SD220	Reset-to-zero signal soft element specified for output axis 0	When SM281 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD240	Soft element specified by the interrupt signal of the output axis	When SM289 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the interrupt signal of the axis 0.	R/W
Note 1: SD202–SD205 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise the base speed is automatically assigned to 1/10 of the max. speed. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 1 (the axis corresponding to Y1)

Address	Name	Function	R/W
SM81	Y1 pulse output stop control	It can be set to stop the high-speed pulse output function of Y1 and reset to open the output function	R/W
SM83	Y1 pulse output monitoring	It is used to monitor the state of the high-speed output channel Y1. It is ON when busy, and OFF when ready.	R

SM64	Y1 pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y1 and reset to disable the pulse output completion interrupt.	R/W
SM290	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y1. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM291	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the Y element Y (N) corresponding to the value N in SD230 is used to indicate the clear signal. When it is reset, Y11 is defined as the clear signal by default.	R/W
SM292	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM293	FWD limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y1. 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.	R/W
SM294	REV limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y1. 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
SM295	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM296	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM297	Logic reversal of the interrupt signal	It is applicable to DVIT, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the interrupt signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the interrupt signal is ON).	R/W
SM298	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y1 when the DSZR/DVIT instruction is executed. It is ON when busy, and OFF when ready.	R/W
SM299	The element specified by the Y1 interrupt signal is valid	It is applicable to DVIT, and acts on the axis corresponding to Y1. When it is set, the value X (N) in SD240 is used to indicate the interrupt input signal. When it is reset, X1 is defined as the interrupt input signal by default.	R/W
SM260	The element specified by the module interrupt signal is valid	It is applicable to DVIT, and acts on the axes corresponding to Y0 and Y1. When it is set, SM289, SM299, and SD240 can be used to designate the interrupt input signal. When it is reset, the designation of the interrupt input signal is disabled.	R/W
Note: When the clear valid signal SM290 is set, the default clear signal Y11 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 1 (the axis corresponding to Y1)

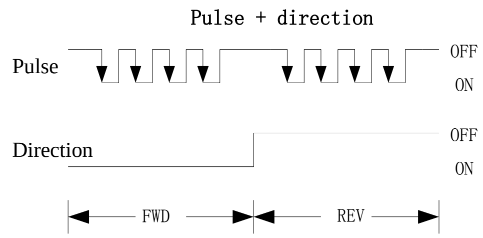
Address	Name	Function	R/W
SD52	The sum (MSB) of accumulated pulses of the output axis 1	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD210–SD211 is calculated and updated based on SD52–SD53 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD210.	R/W
SD53	The sum (LSB) of accumulated pulses of the output axis 1		R/W
SD210	The absolute position current value (MSB) of the output axis 1		R/W
SD211	The absolute position current value (LSB) of the output axis 1		R/W
SD212	Max. speed (MSB) of the output axis 1	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD213	Max. speed (LSB) of the output axis 1		R/W
SD214	Base speed of the output axis 1	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W

Address	Name	Function	R/W
SD215	Acceleration/deceleration time of the output axis 1	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD217	The crawling speed of the output axis 1	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD218	The zero return speed (MSB) of the output axis 1	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD219	The zero return speed (LSB) of the output axis 1		R/W
SD230	Reset-to-zero signal soft element specified for output axis 1	When SM291 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD240	Soft element specified by the interrupt signal of the output axis	When SM299 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the interrupt signal of the axis 1.	R/W
Note 1: SD212–SD215 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise the base speed is automatically assigned to 1/10 of the max. speed. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

11.4.3 Related soft elements of the IVC1L series positioning instructions

The definition and assignment of the output axes of the IVC1L series PLCs are shown in the table below.

Table 11-5 Definition of the IVC1L series output axis

Output axis	Supported mode	Definition of the output points		Output mode definition
0	Pulse + direction	Pulse	Y0	
		Direction	Any output points except Y0	
1	Pulse + direction	Pulse	Y1	
		Direction	Any output points except Y1	
2	Pulse + direction	Pulse	Y2	
		Direction	Any output points except Y2	
3	Pulse + direction	Pulse	Y3	
		Direction	Any output points except Y3	

- Output channel control and monitoring of the output axis 0 (the axis corresponding to Y0)

Address	Name	Function	R/W
SM80	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y0 and reset to open the output function	R/W
SM82	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y0. It is ON when busy, and OFF when ready.	R
SM63	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y0 and reset to disable the pulse output completion interrupt.	R/W
SM280	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y0. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM281	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, the Y element Y (N) corresponding to the value N in SD206 is used to indicate the clear signal. When it is reset, Y10 is defined as the clear signal by default.	R/W
SM282	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W

Address	Name	Function	R/W
SM283	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y0. 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.	R/W
SM284	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y0. 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
SM285	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM286	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y0. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM288	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y0 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM280 is set, the default clear signal Y10 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register of the output axis 0 (the axis corresponding to Y0)

Address	Name	Function	R/W
SD50	The sum (MSB) of accumulated pulses of the output axis 0	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD200–SD201 is calculated and updated based on SD50–SD51 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD200.	R/W
SD51	The sum (LSB) of accumulated pulses of the output axis 0		R/W
SD200	The absolute position current value (MSB) of the output axis 0		R/W
SD201	The absolute position current value (LSB) of the output axis 0		R/W
SD202	Max. speed (MSB) of the output axis 0	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD203	Max. speed (LSB) of the output axis 0		R/W
SD204	Base speed of the output axis 0	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD205	Acceleration time of the output axis 0	Acceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD260	Deceleration time of the output axis 0	Deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD206	Reset-to-zero signal soft element specified for output axis 0	When SM281 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD207	The crawling speed of the output axis 0	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD208	The zero return speed (MSB) of the output axis 0	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD209	The zero return speed (LSB) of the output axis 0		R/W
SD56	Segment number of the envelope output of output axis 0	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD202–SD205 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 1 (the axis corresponding to Y1)

Address	Name	Function	R/W
SM81	Y1 pulse output stop control	It can be set to stop the high-speed pulse output function of Y1 and reset to open the output function	R/W
SM83	Y1 pulse output monitoring	It is used to monitor the state of the high-speed output channel Y1. It is ON when busy, and OFF when ready.	R
SM64	Y1 pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y1 and reset to disable the pulse output completion interrupt.	R/W
SM290	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y1. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM291	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the Y element Y (N) corresponding to the value N in SD230 is used to indicate the clear signal. When it is reset, Y11 is defined as the clear signal by default.	R/W
SM292	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM293	FWD limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y1. 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.	R/W
SM294	REV limit	It is applicable to DSZR/DVIT, and acts on the axis corresponding to Y1. 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
SM295	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM296	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM297	Logic reversal of the interrupt signal	It is applicable to DVIT, and acts on the axis corresponding to Y1. When it is set, it is treated by the negative logic (When the input is OFF, the interrupt signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the interrupt signal is ON).	R/W
SM298	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y1 when the DSZR/DVIT instruction is executed. It is ON when busy, and OFF when ready.	R/W
SM299	The element specified by the Y1 interrupt signal is valid	It is applicable to DVIT, and acts on the axis corresponding to Y1. When it is set, the value X (N) in SD240 is used to indicate the interrupt input signal. When it is reset, X1 is defined as the interrupt input signal by default.	R/W
SM260	The element specified by the module interrupt signal is valid	It is applicable to DVIT, and acts on the axes corresponding to Y0 and Y1. When it is set, SM289, SM299, and SD240 can be used to designate the interrupt input signal. When it is reset, the designation of the interrupt input signal is disabled.	R/W
Note: When the clear valid signal SM290 is set, the default clear signal Y11 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register ofthe output axis 1 (the axis corresponding to Y1)

Address	Name	Function	R/W
SD52	The sum (MSB) of accumulated pulses of the output axis 1	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD210–SD211 is calculated and updated based on SD52–SD53 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD210.	R/W
SD53	The sum (LSB) of accumulated pulses of the output axis 1		R/W
SD210	The absolute position current value (MSB) of the output axis 1		R/W
SD211	The absolute position current value (LSB) of the output axis 1		R/W

Address	Name	Function	R/W
SD212	Max. speed (MSB) of the output axis 1	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD213	Max. speed (LSB) of the output axis 1		R/W
SD214	Base speed of the output axis 1	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD215	Acceleration time of the output axis 1	Acceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD261	Deceleration time of the output axis 1	Deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD217	The crawling speed of the output axis 1	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD218	The zero return speed (MSB) of the output axis 1	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD219	The zero return speed (LSB) of the output axis 1		R/W
SD230	Reset-to-zero signal soft element specified for output axis 1	When SM291 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD240	Soft element specified by the interrupt signal of the output axis	When SM299 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the interrupt signal of the axis 1.	R/W
Note 1: SD212–SD215 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise the base speed is automatically assigned to 1/10 of the max. speed.No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed.			

- Output channel control and monitoring of the output axis 2 (the axis corresponding to Y2)

Address	Name	Function	R/W
SM262	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y2 and reset to open the output function	R/W
SM272	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y2. It is ON when busy, and OFF when ready.	R
SM72	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y2 and reset to disable the pulse output completion interrupt.	R/W
SM320	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y2. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM321	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, the Y element Y (N) corresponding to the value N in SD326 is used to indicate the clear signal. When it is reset, Y12 is defined as the clear signal by default.	R/W
SM322	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM323	FWD limit	It is applicable to DSZR, and acts on the axis 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.	R/W
SM324	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y2. 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
SM325	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM326	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y2. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is	R/W

		ON, the zero signal is ON).	
SM328	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y2 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W
Note: When the clear valid signal SM320 is set, the default clear signal Y12 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.			

- Special data register ofthe output axis 2 (the axis corresponding to Y2)

Address	Name	Function	R/W
SD160	The sum (MSB) of accumulated pulses of the output axis 2	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD320–SD321 is calculated and updated based on SD160–SD161 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD320.	R/W
SD161	The sum (LSB) of accumulated pulses of the output axis 2		R/W
SD320	The absolute position current value (MSB) of the output axis 2		R/W
SD321	The absolute position current value (LSB) of the output axis 2		R/W
SD322	Max. speed (MSB) of the output axis 2	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD323	Max. speed (LSB) of the output axis 2		R/W
SD324	Base speed of the output axis 2	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD325	Acceleration time of the output axis 2	Acceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD262	Deceleration time of the output axis 2	Deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD326	Reset-to-zero signal soft element specified for output axis 2	When SM321 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD327	The crawling speed of the output axis 2	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD328	The zero return speed (MSB) of the output axis 2	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD329	The zero return speed (LSB) of the output axis 2		R/W
SD252	Segment number of the envelope output of output axis 2	It is applicable to PLS, and used to detect the current segment number of the envelope output	R
Note 1: SD322–SD325 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

- Output channel control and monitoring of the output axis 3 (the axis corresponding to Y3)

Address	Name	Function	R/W
SM263	Pulse output stop control	It can be set to stop the high-speed pulse output function of Y3 and reset to open the output function	R/W
SM273	Pulse output monitoring	It is used to monitor the state of the high-speed output channel Y3. It is ON when busy, and OFF when ready.	R
SM73	Pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y3 and reset to disable the pulse output completion interrupt.	R/W
SM330	Enabling the clear function	It is applicable to DSZR/ZRN, and acts on the axis corresponding to Y3. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM331	The element specified by the clear signal is valid	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, the Y element Y (N) corresponding to the value N in SD336 is used to indicate the clear signal. When it is reset, Y13 is defined as the clear signal by default	R/W

Address	Name	Function	R/W
SM332	The ZRN direction	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, the ZRN direction is the forward direction. When it is reset, the ZRN direction is the reverse direction.	R/W
SM333	FWD limit	It is applicable to DSZR, and acts on the axis corresponding to Y3. 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.	R/W
SM334	REV limit	It is applicable to DSZR, and acts on the axis corresponding to Y3. 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
SM335	Logic reversal of the DOG signal	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, it is treated by the negative logic (When the input is OFF, the DOG signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the DOG signal is ON).	R/W
SM336	Logic reversal of the zero signal	It is applicable to DSZR, and acts on the axis corresponding to Y3. When it is set, it is treated by the negative logic (When the input is OFF, the zero signal is ON). When it is reset, it is treated by the positive logic (When the input is ON, the zero signal is ON).	R/W
SM338	The positioning instruction is driving	It is used to monitor the state of the high-speed output channel Y3 when the DSZR instruction is executed. It is ON when busy, and OFF when ready.	R/W

Note: When the clear valid signal SM330 is set, the default clear signal Y13 corresponding to the output axis outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If the default clear signal of this axis is used for other purposes, the clear valid signal should be reset to disable this function.

- Special data register of the output axis 3 (the axis corresponding to Y3)

Address	Name	Function	R/W
SD162	The sum (MSB) of accumulated pulses of the output axis 3	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD330–SD331 is calculated and updated based on SD162–SD163 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD330.	R/W
SD163	The sum (LSB) of accumulated pulses of the output axis 3		R/W
SD330	The absolute position current value (MSB) of the output axis 3		R/W
SD331	The absolute position current value (LSB) of the output axis 3		R/W
SD332	Max. speed (MSB) of the output axis 3	The max. speed at which the output axis executes the positioning instruction. Range: 10–200000. Unit: pulse.	R/W
SD333	Max. speed (LSB) of the output axis 3		R/W
SD334	Base speed of the output axis 3	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction	R/W
SD335	Acceleration time of the output axis 3	Acceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD263	Deceleration time of the output axis 3	Deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD336	Reset-to-zero signal soft element specified for output axis 3	When SM331 is set, the Y element Y (N) corresponding to the value N in the element is used to indicate the clear signal.	R/W
SD337	The crawling speed of the output axis 3	It is applicable to DSZR, and used as the crawling speed at which the instruction is executed	R/W
SD338	The zero return speed (MSB) of the output axis 3	It is applicable to DSZR, and used as the zero return speed at which the instruction is executed	R/W
SD339	The zero return speed (LSB) of the output axis 3		R/W
SD253	Segment number of the envelope output of output axis 3	It is applicable to PLS, and used to detect the current segment number of the envelope output	R

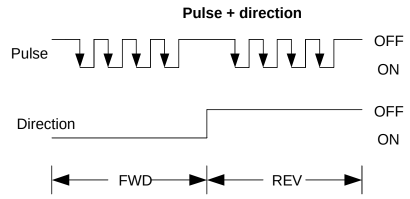
Note 1: SD332–SD335 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction.

Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed.

11.4.4 Related soft elements of the IVC1 series positioning instructions

The definition and assignment of the output axes of the IVC1 series PLCs are shown in the table below.

Table 11-6 Definition of the IVC1 series output axis

Output axis	Supported mode	Definition of the output points		Output mode definition
0	Pulse + direction	Pulse	Y0	
		Direction	Any output points except Y0	
1	Pulse + direction	Pulse	Y1	
		Direction	Any output points except Y1	

- Output channel control and monitoring

Address	Name	Function	R/W
SM80	Y0 pulse output stop control	It can be set to stop the high-speed pulse output function of Y0 and reset to open the output function	R/W
SM81	Y1 pulse output stop control	It can be set to stop the high-speed pulse output function of Y1 and reset to open the output function	R/W
SM82	Y0 pulse output monitoring	It is used to monitor the state of the high-speed output channel Y0. It is ON when busy, and OFF when ready.	R
SM83	Y1 pulse output monitoring	It is used to monitor the state of the high-speed output channel Y1. It is ON when busy, and OFF when ready.	R
SM63	Y0 pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y0 and reset to disable the pulse output completion interrupt.	R/W
SM64	Y1 pulse output completion interrupt enable control	It can be set to enable the pulse output completion interrupt of Y1 and reset to disable the pulse output completion interrupt.	R/W
SM85	Enabling the clear function	It is applicable to ZRN, and acts on the axes corresponding to Y0 and Y1. When it is set, the CLR signal output function of the ZRN instruction is valid. When it is reset, the CLR signal output is not provided.	R/W
SM86	Enabling Y0 pulse output when driven by aninterrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprogram. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow.	R/W
SM87	Enabling Y1 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprogram. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow.	R/W
SM89	PLSV gradual frequency conversion	When it is ON, the frequency changes gradually.	R/W
Note: When SM85 is set, Y2 or Y3 outputs a CLR pulse output with the pulse width of 20 ms+1 scan period when the origin returns to the origin position. If Y2 or Y3 is used for other purposes, SM85 should be reset to disable this function.			

- Special data register of the output channel

Address	Name	Function	R/W
SD50	The sum (MSB) of accumulated pulses of the output axis 0	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD80–SD81 is calculated and updated based on SD50–SD51 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute position data (32-bit long integer) needs to be put into SD80.	R/W
SD51	The sum (LSB) of accumulated pulses of the output axis 0		R/W
SD80	The absolute position current value (MSB) of the output axis 0		R/W
SD81	The absolute position current value (LSB) of the output axis 0		R/W
SD52	The sum (MSB) of accumulated pulses of the output axis 1	They are used to calculate and preserve the absolute position. Each time a positioning instruction is executed, SD82–SD83 is calculated and updated based on SD52–SD53 and the direction signal. When powered on and the absolute position data is read from the servo drive, the obtained absolute	R/W
SD53	The sum (LSB) of accumulated pulses of the output axis 1		R/W

SD82	The absolute position current value (MSB) of the output axis 1	position data (32-bit long integer) needs to be put into SD82.	R/W
SD83	The absolute position current value (LSB) of the output axis 1		R/W
SD84	Base speed of the output axis 0 and 1	The base speed (less than 1/10 of the max. speed) at which the output axis executes the positioning instruction.	R/W
SD85	Max. speed (MSB) of the output axis 0 and 1	The max. speed at which the output axis executes the positioning instruction. Range: 10–100000. Unit: pulse.	R/W
SD86	Max. speed (LSB) of the output axis 0 and 1		R/W
SD87	Acceleration/deceleration time of the output axis 0 and 1	Acceleration/deceleration time when the output axis executes the positioning instruction. Range: 50–5000. Unit: ms.	R/W
SD56	Segment number of the envelope output of output axis 0	It is applicable to PLS, and used to detect the current segment number of the envelope output.	R
SD57	Segment number of the envelope output of output axis 1	It is applicable to PLS, and used to detect the current segment number of the envelope output.	R
Note 1: SD84–SD87 can be modified by you as needed. You need to assign the value before driving the positioning instruction. Changing the above parameters during the execution of the positioning instruction is likely to affect the correct execution of the instruction. Note 2: The base speed must be less than 1/10 of the max. speed, otherwise no pulse is outputted. No pulse is outputted when the speed in the positioning instruction is lower than the base speed or greater than the max.speed			

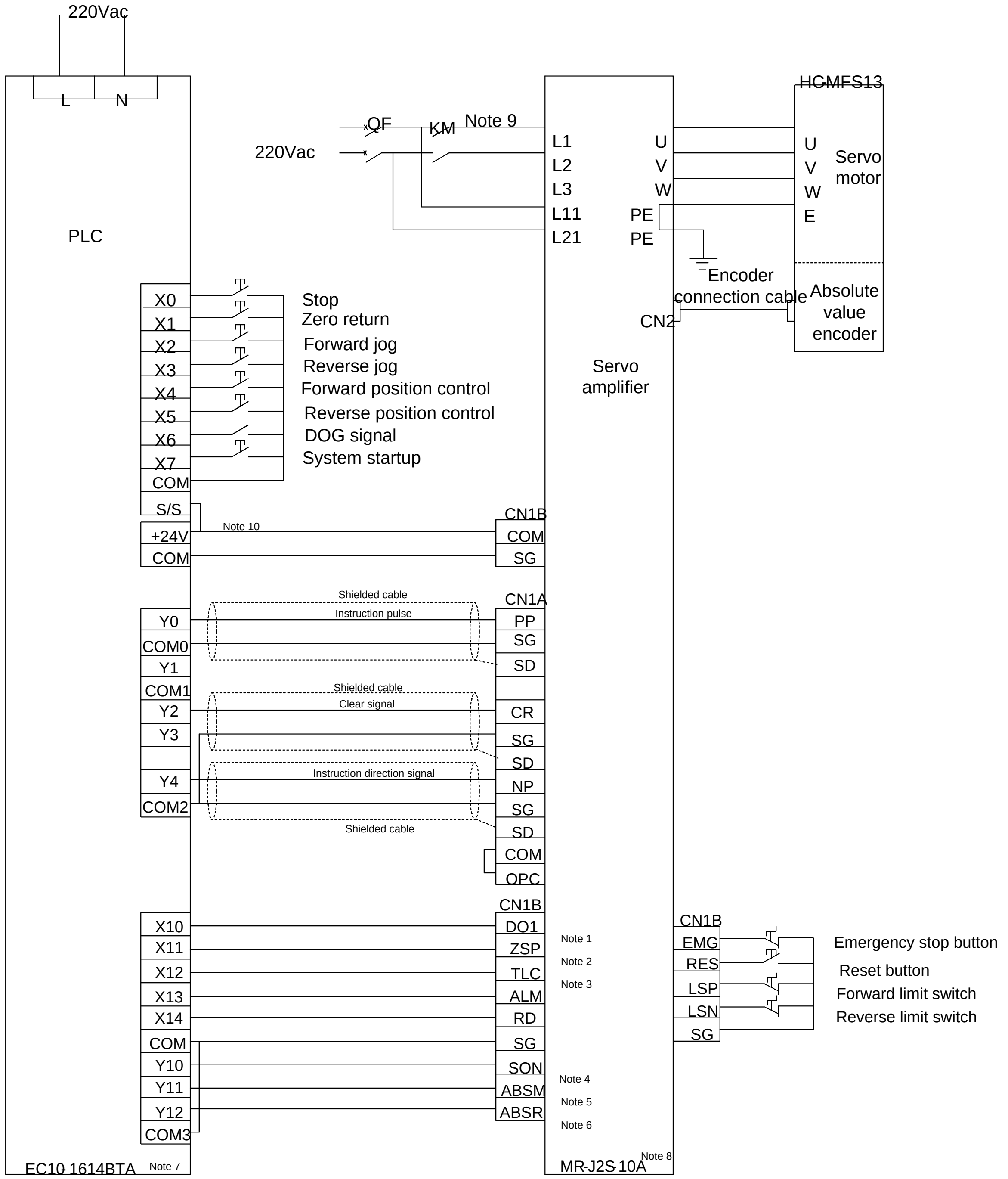
11.5 Application instance

11.5.1 Example of a pulse output program

- Mechanical sketch

  You can refer to the diagram of the example in Section 6.17.3. This example implements the absolute coordinate system of a single-axis screw.

- Device wiring diagram



Note 1: Used as the data transmission line of ABS bit 1 when ABSM is set to ON, and as a "Positioning completed" signal when ABSM is set to OFF.
Note 2: Used as the data transmission line of ABS bit 2 when ABSM is set to ON, and as a zero speed signal when ABSM is set to OFF.
Note 3: Used as "Ready for transfer data" signal when ABSM is set to ON, and as a "Torque limit" signal when ABSM is set to OFF.
Note 4: Used as the servo enable signal, and set to on before the ABS instruction is executed.
Note 5: Used as the ASB transmission mode signal.
Note 6: Used as the ASB transmission request signal.
Note 7: A transistor output type of PLC main module is required.
Note 8:Please install the device based on the installation guide for corresponding servo amplifier; the shape of the multiple interface plugs is exactly the same, please be careful not to insert the wrong.
Note 9:In the case of an alarm or emergency, the program needs to control the KM and disconnect the power supply.
Note 10:In this example, the PLC adopts a leaking input, and the +24V and S/S terminals are short connected.

Figure 11-8 Device wiring diagram

- Program example

The program implements the following functions:

1. When entering the running state, the PLC reads the absolute position data of the servo drive through the ABS instruction or communication mode (It is required that the servo drive be powered on before or at the same time as the PLC is running).

2. Entering the running state, SM85 is set, and the clear output function is set. Y2 outputs the clear pulse every time the origin returns to the origin position.

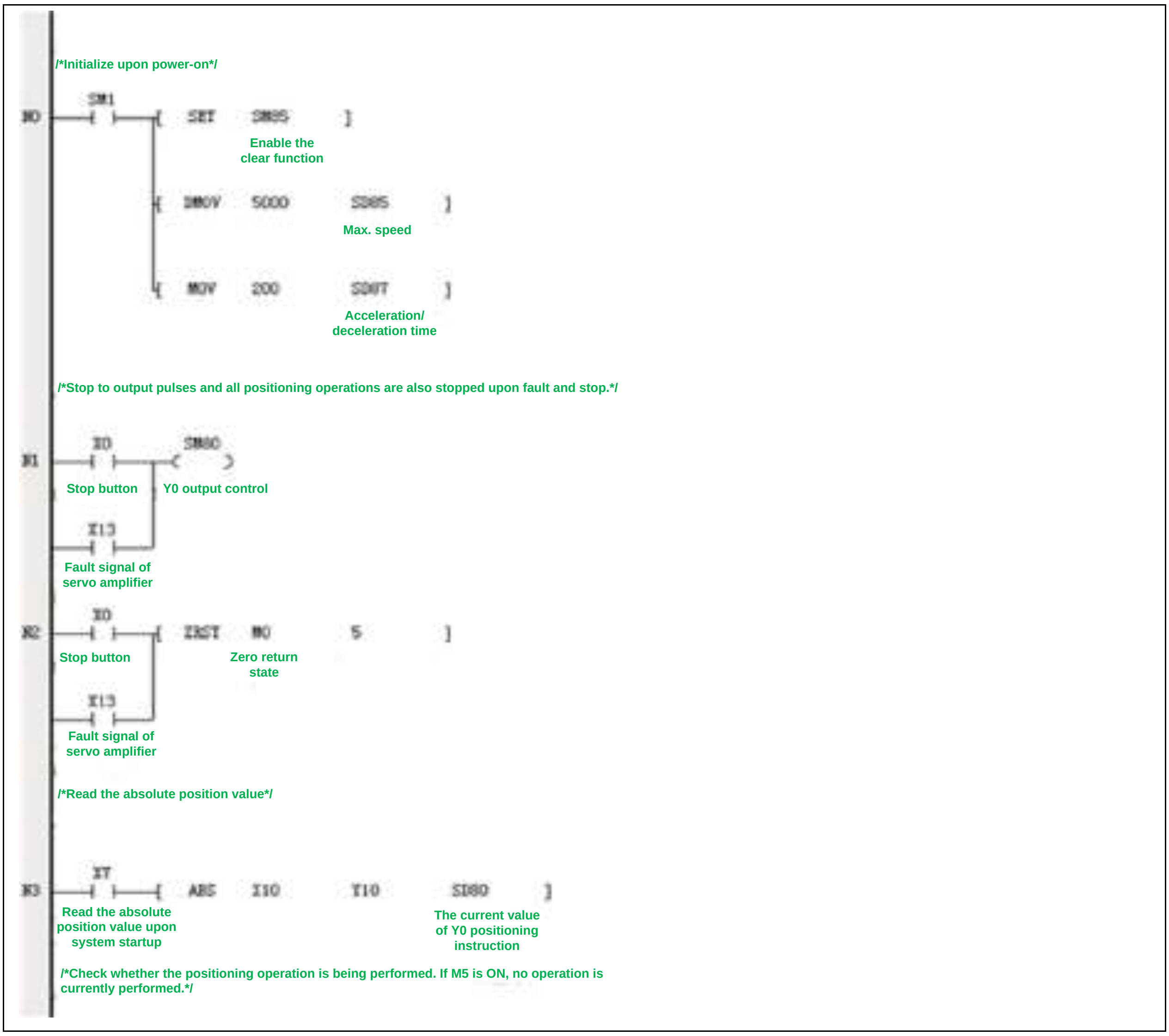
3. Operating the JOG+ button to jog in the forward direction.

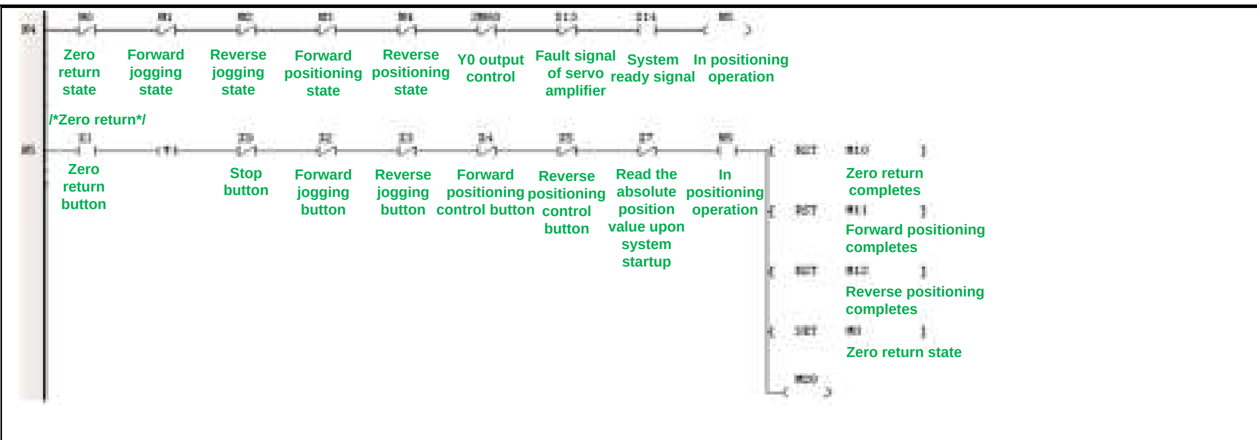
4. Operating the JOG-button to jog in the reverse direction.

5. Operating the ZRN button to perform the ZRN operation in the backward direction when the table is at a position farther than the front end of the DOG signal.

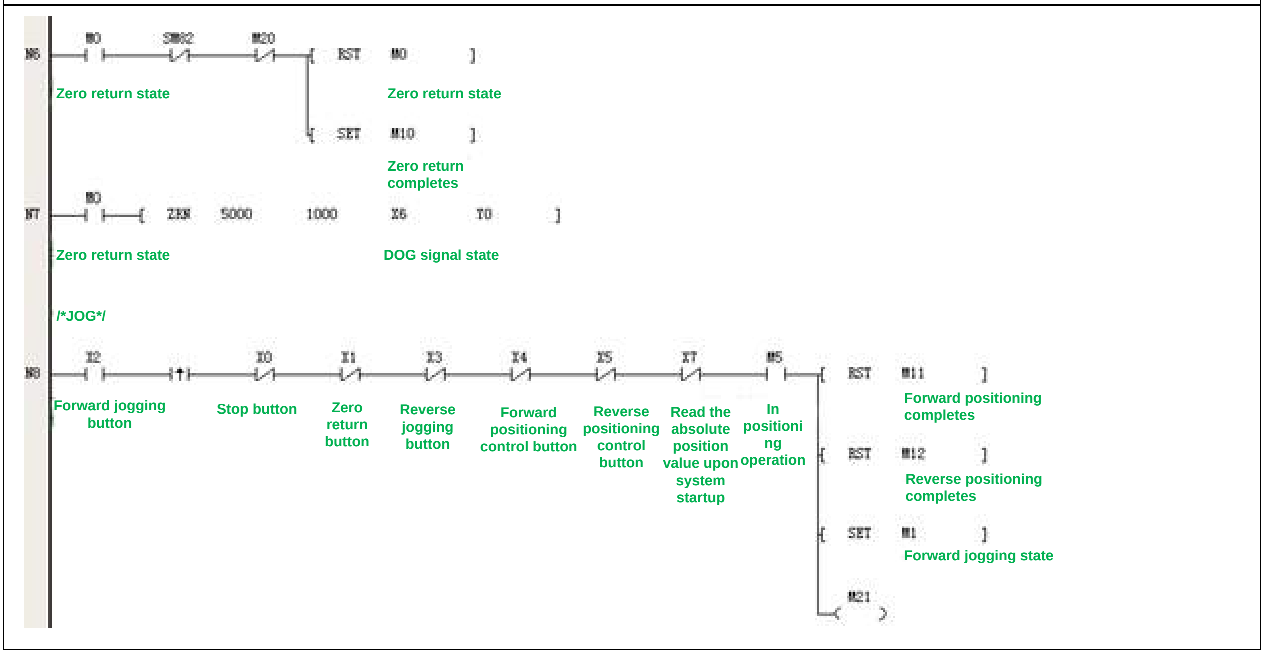
6. Pressing the stop button when the tableis running, then the table stops running.

7. Operating the forward and reverse position control buttons to implementthe positioning operation on the table.

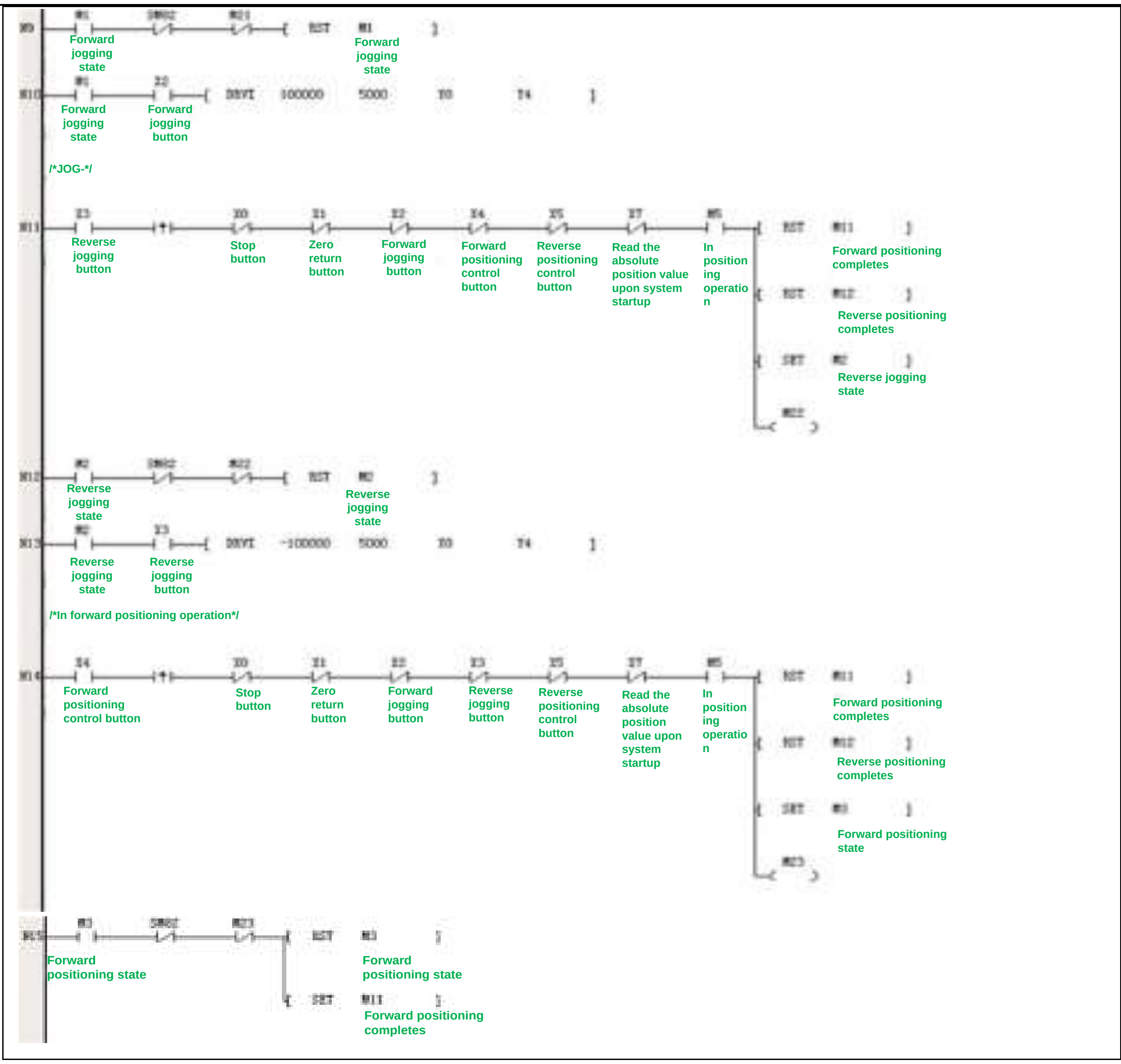


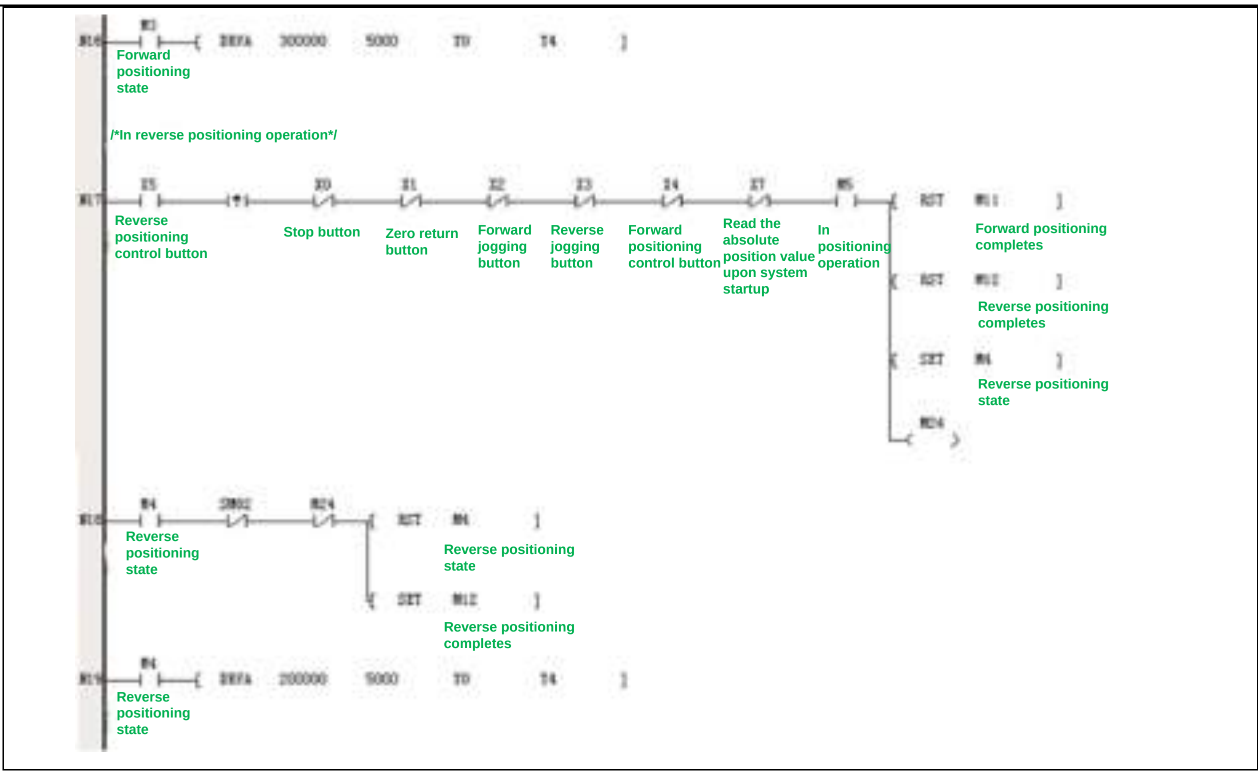


Zero return state
Forward jogging state
Reverse jogging state
Forward positioning state
Reverse positioning state
Y0 output control
Fault signal of servo amplifier
System ready signal
In positioning operation
/*Zero return*/
Zero return button
Stop button
Forward jogging button
Reverse jogging button
Forward positioning control button
Reverse positioning control button
Read the absolute position value upon system startup
In positioning operation
Zero return completes
Forward positioning completes
Reverse positioning completes
Zero return state



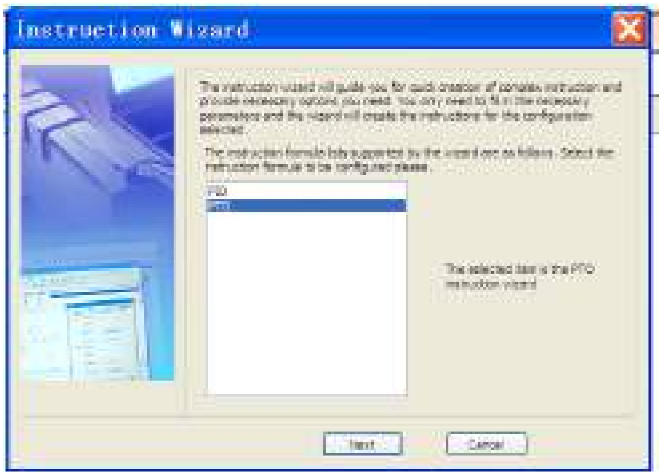
N6
M0
SM82
M20
RST
M0
Zero return state
Zero return state
SET
M10
Zero return completes
N7
M0
ZRN
5000
1000
X6
Y0
Zero return state
DOG signal state
/*JOG*/
N8
X2
X0
X1
X3
X4
X5
X7
M5
RST
M11
Forward positioning completes
Forward jogging button
Stop button
Zero return button
Reverse jogging button
Forward positioning control button
Reverse positioning control button
Read the absolute position value upon system startup
In positioning operation
RST
M12
Reverse positioning completes
SET
M1
Forward jogging state
M21


Forward jogging state
Forward jogging state
Forward jogging state
Forward jogging button
/*JOG-*/
Reverse jogging button
Stop button
Zero return button
Forward jogging button
Forward positioning control button
Reverse positioning control button
Read the absolute position value upon system startup
In positioning operation
Forward positioning completes
Reverse positioning completes
Reverse jogging state
Reverse jogging state
Reverse jogging state
Reverse jogging state
Reverse jogging button
/*In forward positioning operation*/
Forward positioning control button
Stop button
Zero return button
Forward jogging button
Reverse jogging button
Reverse positioning control button
Read the absolute position value upon system startup
In positioning operation
Forward positioning completes
Reverse positioning completes
Forward positioning state
Forward positioning state
Forward positioning state
Forward positioning completes



- Configuration method of the PLS envelope instruction

The PLS instruction is generated by using the PTO instruction wizard. In Auto Station, you can select **Tool** ->**Instruction Wizard**... to configure the PLS instruction. Selecting PTO as shown below.



Clicking **Next** to enter the configuration interface where high-speed pulse output points, max. and min. frequency of the high-speed pulse output, and ACC/DEC time can be set, shown in the following figure.

During acceleration/deceleration time, the acceleration speed of all the segments of the envelope curve is constant. For example, if the settings are as shown above, then the acceleration time taken to accelerate from 20000 Hz to 50000 Hz for the motor is:

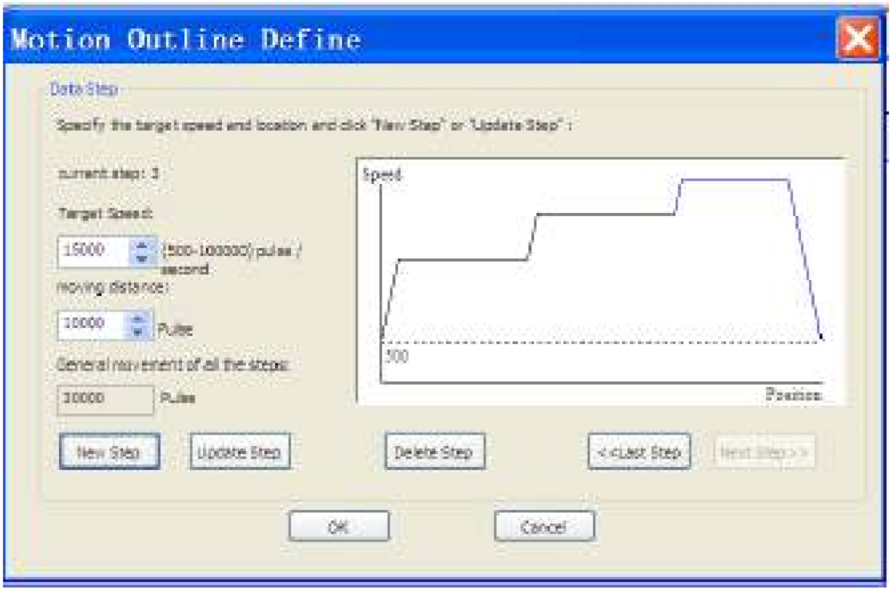
$$1000\times(50000-20000)\div(100000-5000)=316\ (ms)=0.316\ (s)$$

During the acceleration time, the total number of output pulses can be calculated through using the trapezoidal area calculation formula:

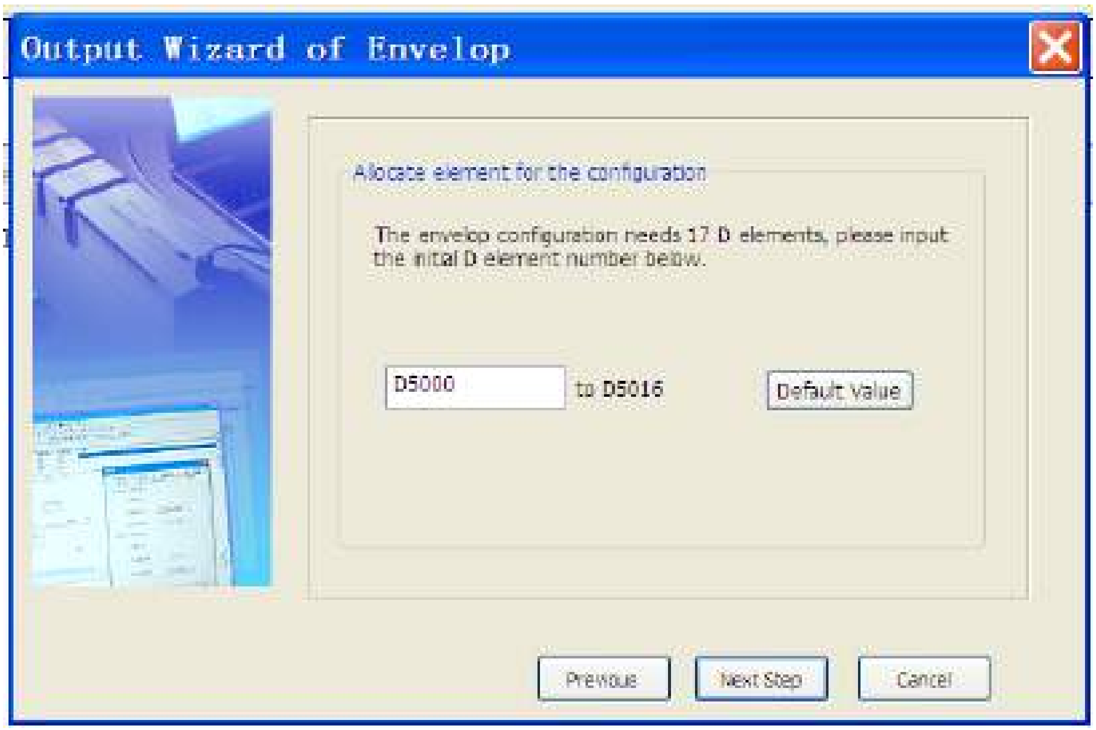
$$(20000+50000)\times0.316\div2=11060\ (\text{number of pulses})$$

Therefore, if you have a requirement for the time or number of pulses during acceleration/deceleration, you need to perform the related calculations before setting the max. speed, min. speed and acceleration/deceleration time.

After clicking **Next Step** in the above figure, enter **Motion Outline Define** in the figure below. You can enter the target speed and moving distance of the first step first, and click **New Step**. Then you can enter the target speed and moving distance of the second step, and click **New Step**, and so on, and finally click **OK**.

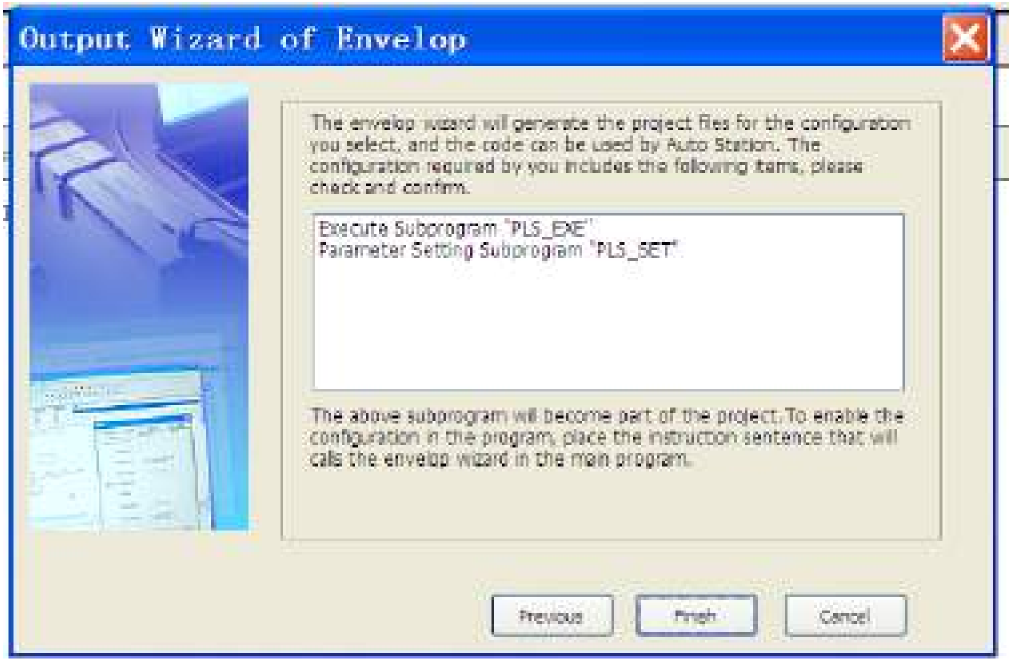


The above configuration is saved in the D elements, and you can choose which D elements to store these settings, as shown below.



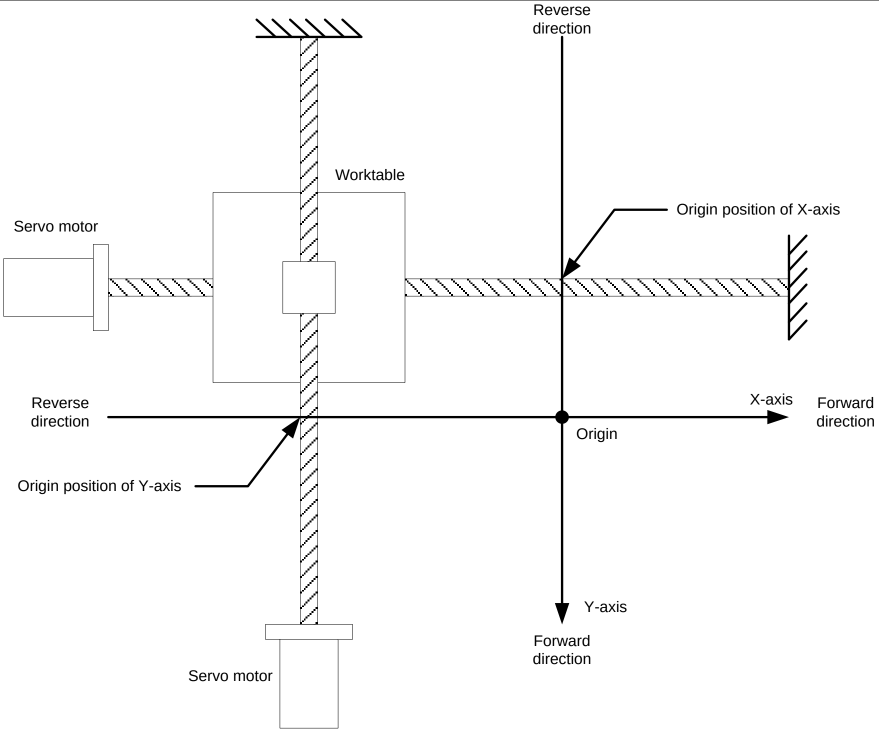
The wizard will create two subprograms, one is the parameter setting subprogram, and the otheris the PLS execution subprogram, as shown below. You need to make sure that the parameter setting subprogram is correctly called and executed (the relevant D elementsare assigned) when programming in the main program, and then call the execution subprogram.

At this point, all configuration has been completed, as shown in the figure below. Clicking **Finish** to complete the PTO configuration.

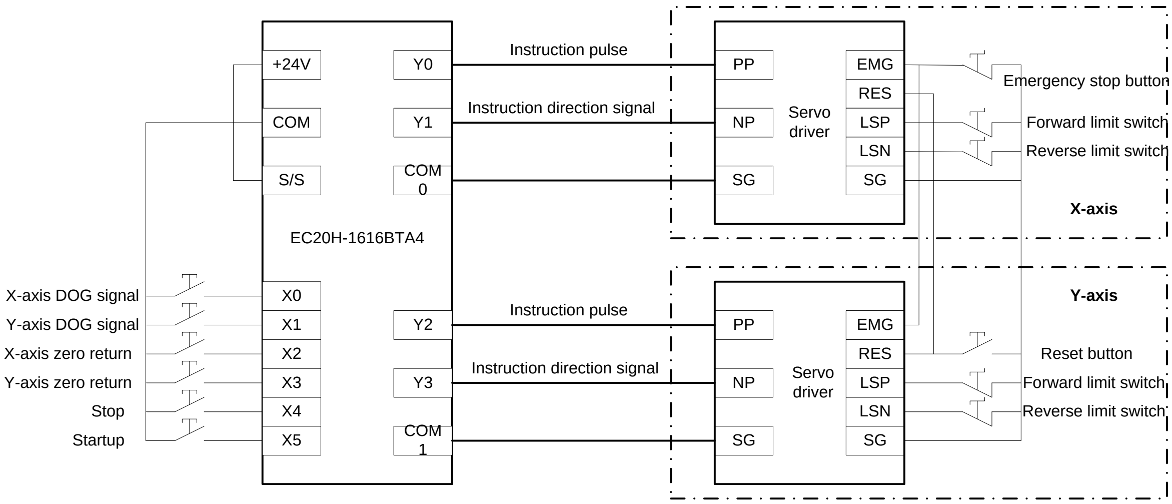


11.5.2 Example of trajectory interpolation

- Mechanical sketch



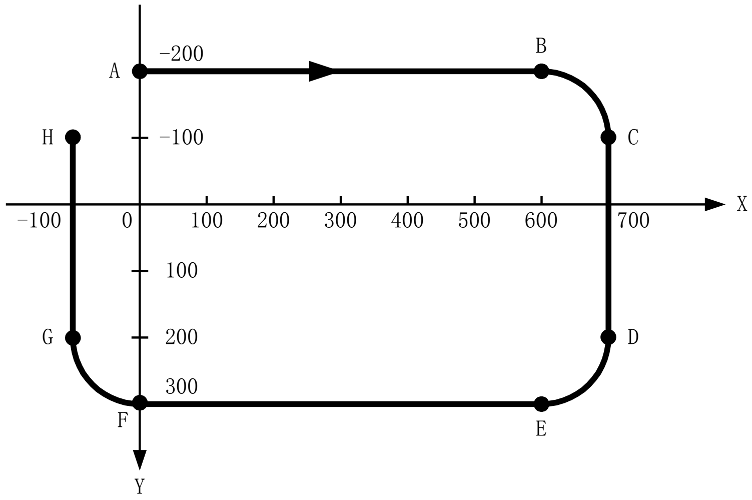
- Device wiring diagram

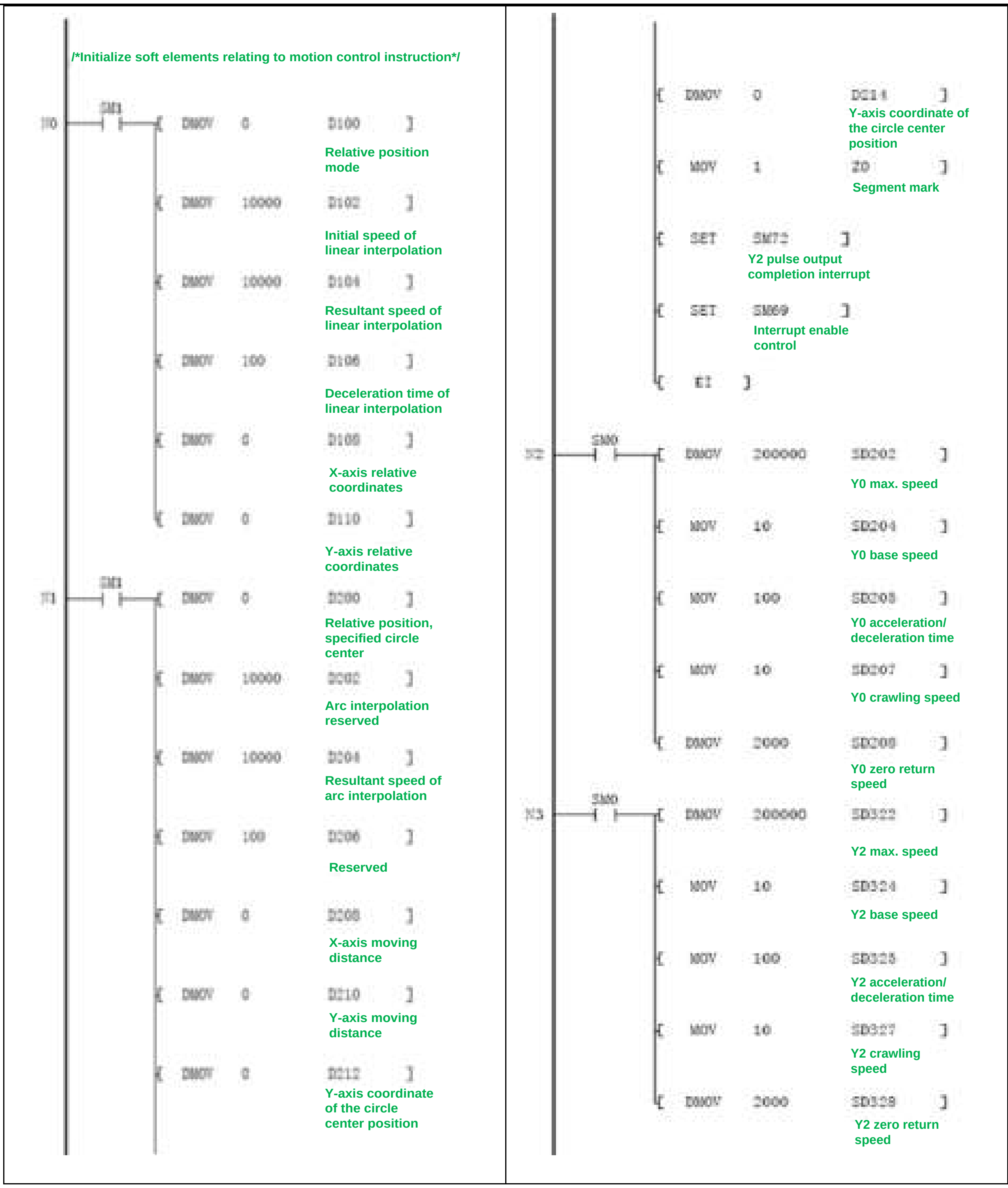


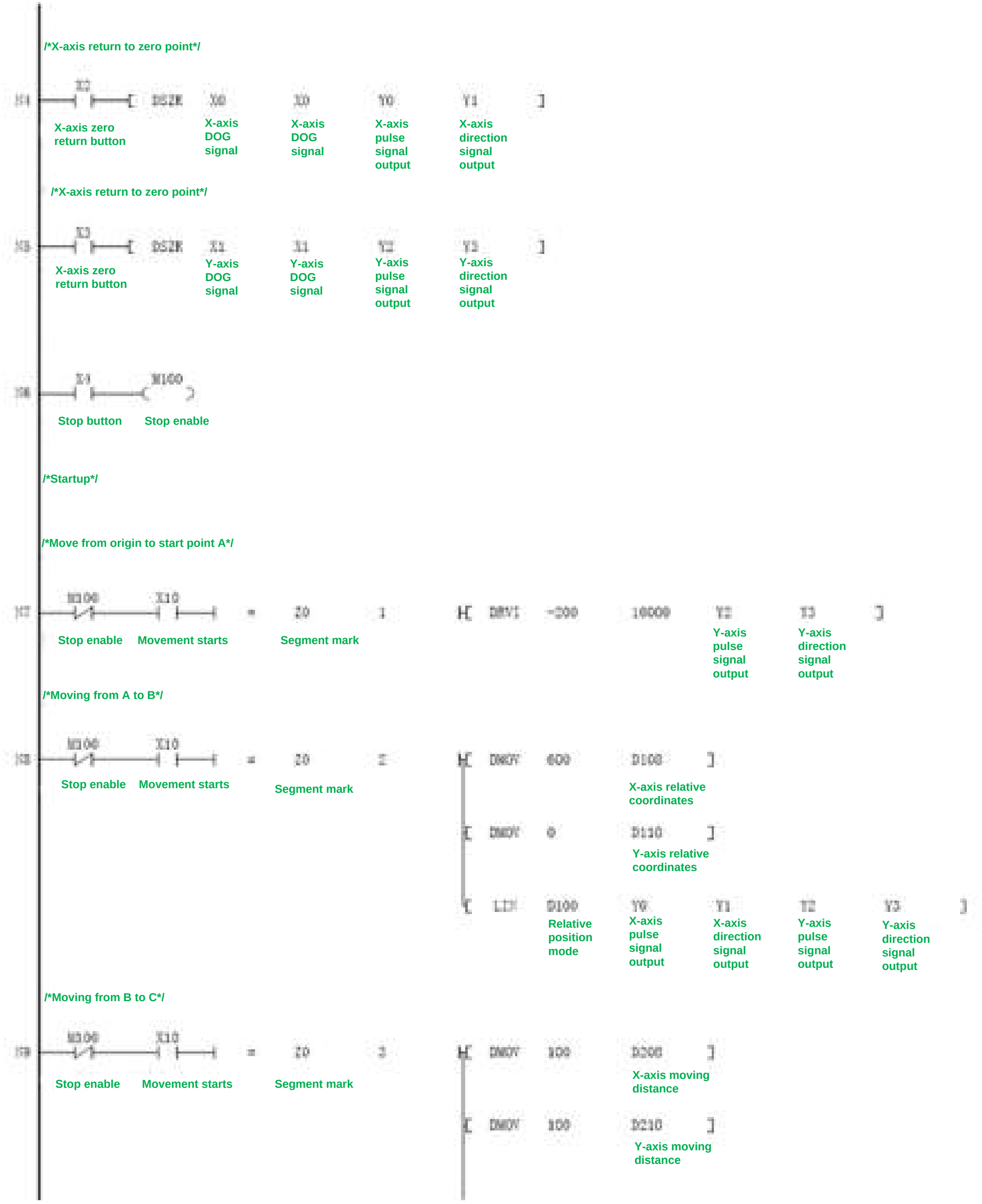
- Program example

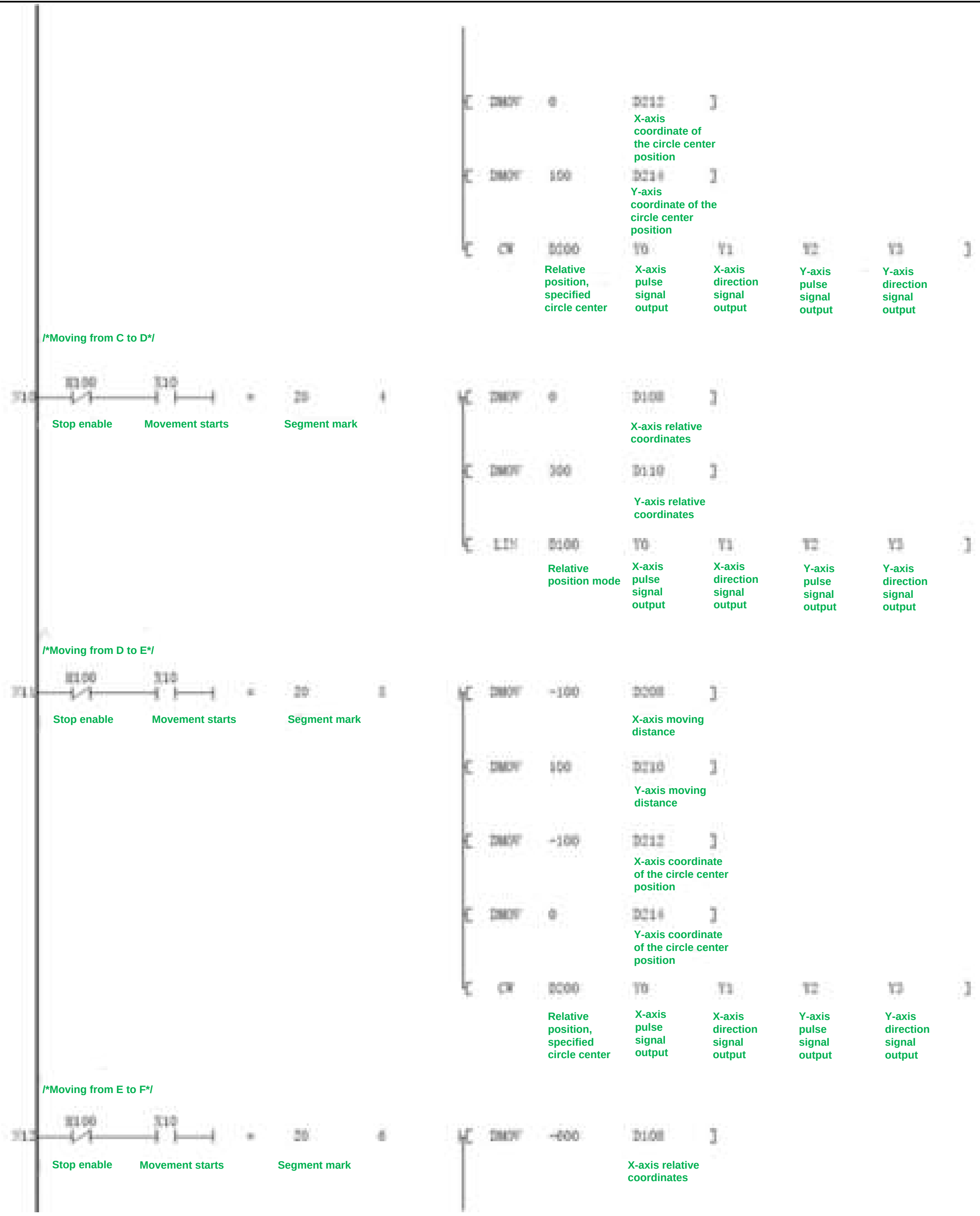
The program implements the following functions:

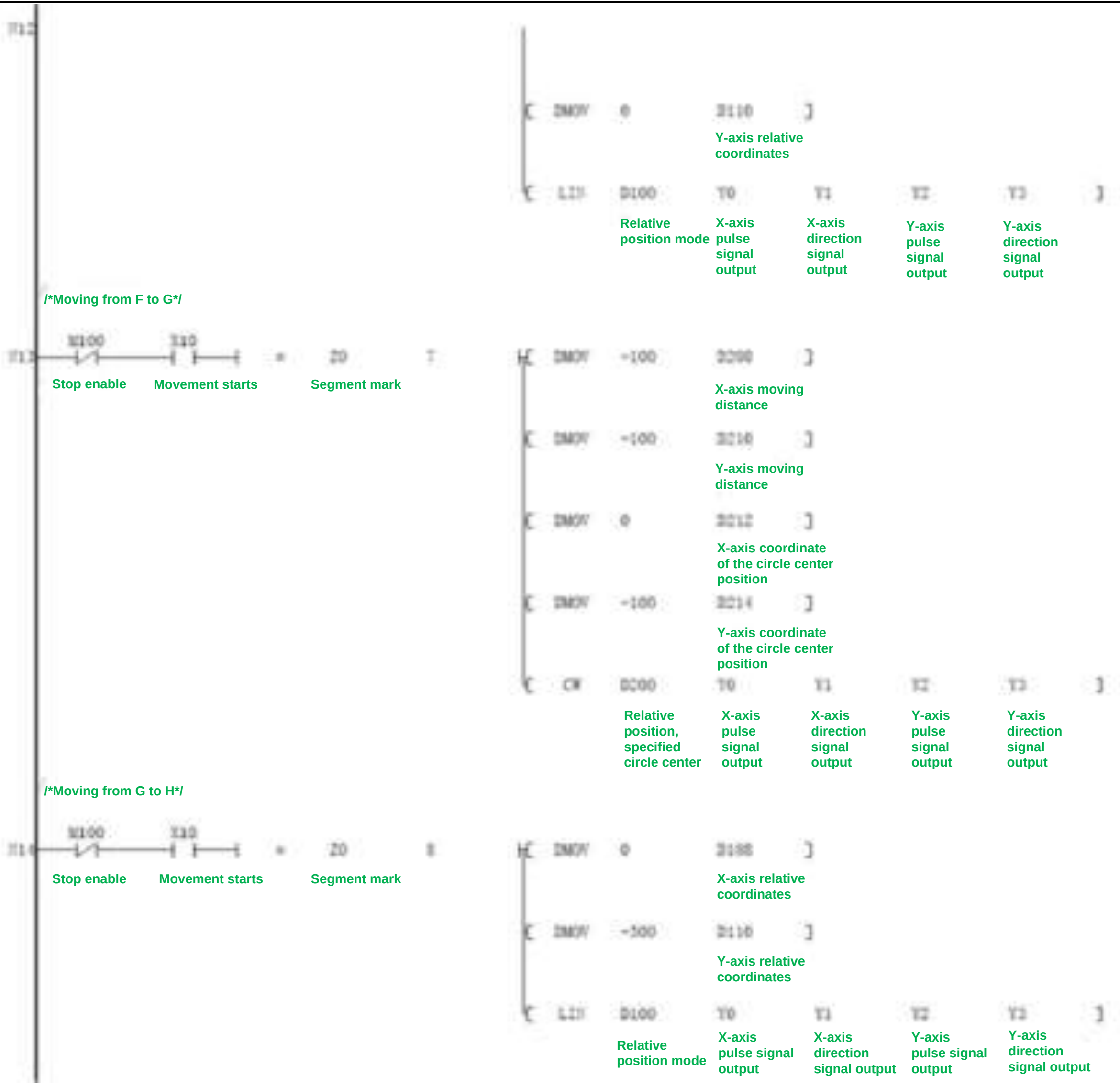
1. Operating the ZRN button, and the table automatically searches for the origin and stops running when reaching the origin.
2. Entering the running state, operating the start button, and running continuously according to the illustrated track.
3. Pressing the stop button when the tableis running, then the table stops running.


A
B
C
D
E
F
G
H
-200
-100
100
200
300
-100
0
100
200
300
400
500
600
700
X
Y


/*Initialize soft elements relating to motion control instruction*/
N0
SM1
DMOV 0 D100
Relative position mode
DMOV 10000 D102
Initial speed of linear interpolation
DMOV 10000 D104
Resultant speed of linear interpolation
DMOV 100 D106
Deceleration time of linear interpolation
DMOV 0 D108
X-axis relative coordinates
DMOV 0 D110
Y-axis relative coordinates
N1
SM1
DMOV 0 D200
Relative position, specified circle center
DMOV 10000 D202
Arc interpolation reserved
DMOV 10000 D204
Resultant speed of arc interpolation
DMOV 100 D206
Reserved
DMOV 0 D208
X-axis moving distance
DMOV 0 D210
Y-axis moving distance
DMOV 0 D212
Y-axis coordinate of the circle center position
DMOV 0 D214
Y-axis coordinate of the circle center position
MOV 1 Z0
Segment mark
SET SM72
Y2 pulse output completion interrupt
SET SM69
Interrupt enable control
EI
N2
SM0
DMOV 200000 SD202
Y0 max. speed
MOV 10 SD204
Y0 base speed
MOV 100 SD205
Y0 acceleration/ deceleration time
MOV 10 SD207
Y0 crawling speed
DMOV 2000 SD208
Y0 zero return speed
N3
SM0
DMOV 200000 SD322
Y2 max. speed
MOV 10 SD324
Y2 base speed
MOV 100 SD325
Y2 acceleration/ deceleration time
MOV 10 SD327
Y2 crawling speed
DMOV 2000 SD328
Y2 zero return speed


/*X-axis return to zero point*/
X-axis zero return button
X-axis DOG signal
X-axis DOG signal
X-axis pulse signal output
X-axis direction signal output
/*X-axis return to zero point*/
X-axis zero return button
Y-axis DOG signal
Y-axis DOG signal
Y-axis pulse signal output
Y-axis direction signal output
Stop button
Stop enable
/*Startup*/
/*Move from origin to start point A*/
Stop enable
Movement starts
Segment mark
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from A to B*/
Stop enable
Movement starts
Segment mark
X-axis relative coordinates
Y-axis relative coordinates
Relative position mode
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from B to C*/
Stop enable
Movement starts
Segment mark
X-axis moving distance
Y-axis moving distance


X-axis coordinate of the circle center position
Y-axis coordinate of the circle center position
Relative position, specified circle center
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from C to D*/
Stop enable
Movement starts
Segment mark
X-axis relative coordinates
Y-axis relative coordinates
Relative position mode
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from D to E*/
Stop enable
Movement starts
Segment mark
X-axis moving distance
Y-axis moving distance
X-axis coordinate of the circle center position
Y-axis coordinate of the circle center position
Relative position, specified circle center
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from E to F*/
Stop enable
Movement starts
Segment mark
X-axis relative coordinates



Y-axis relative coordinates
Relative position mode
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from F to G*/
Stop enable
Movement starts
Segment mark
X-axis moving distance
Y-axis moving distance
X-axis coordinate of the circle center position
Y-axis coordinate of the circle center position
Relative position, specified circle center
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output
/*Moving from G to H*/
Stop enable
Movement starts
Segment mark
X-axis relative coordinates
Y-axis relative coordinates
Relative position mode
X-axis pulse signal output
X-axis direction signal output
Y-axis pulse signal output
Y-axis direction signal output

Appendix A Special auxiliary relay

All special auxiliary relays are initialized when the PLCs change from STOP to RUN. The special auxiliary relays that have been set in the system setting are set to the preset value in the system block after the initialization.

Note

The reserved SD and SM elements are not listed in the table. The reserved SM elements are by default read only (R).

1. PLC working state flag

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM0	Monitoring run bit	This bit is always high in RUN, and 0 in STOP.	R	√	√	√	√
SM1	Initial run pulse bit	This bit is set high when the user program changes from STOP to RUN, and set low after a scan cycle	R	√	√	√	√
SM2	Power-on flag bit	This bit is set high after the system is powered-on, and set low after a scan cycle	R	√	√	√	√
SM3	System error	This bit is set when a system error occurs after power-on or after PLC changes from STOP to RUN, or reset if no system error occurs	R	√	√	√	√
SM4	Battery voltage low	This bit is set when the battery voltage is too low, or reset if the battery voltage is detected to be higher than 2.4 V.	R		√	√	√
SM5	AC power failure check bit	This bit is set when the PLC detects AC power loss (detecting time 40 ms), or reset if the power is on after the delay of power failure detecting time (set in SD05).	R	√	√	√	√
SM6	24 Vdc power failure	This bit is set when the PLC detects the 24 Vdc power failure (detecting time 50 ms), or reset if the power is detected to be back within the following 50 ms.	R	√	√	√	√
SM7	No battery working mode	If this bit is set to 1 (configurable only through the system block), the system does not report errors caused by loss of battery backup data or loss of forcing tables upon backup battery failure.	R		√	√	√
SM8	Constant scan mode	Setting this bit, and the scan time is constant (configurable only through the system block)	R	√	√	√	√
SM9	Startup mode of the input point	Setting this bit, and the PLC can changes from STOP to RUN when the designated X input point is ON (configurable only through the system block)	R	√	√	√	√

2. Clock running bit

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM10	10 ms clock	Clock oscillation in 10 ms cycle (reversing every half cycle. The first half cycle is 0 during the operation of the user program)	R	√	√	√	√
SM11	100 ms clock	Clock oscillation in 100 ms cycle (reversing every half cycle. The first half cycle is 0 during the operation of the user program)	R	√	√	√	√
SM12	1 s clock	Clock oscillation in 1 s cycle (reversing every half cycle. The first half cycle is 0 during the operation of the user program)	R	√	√	√	√
SM13	1 min clock	Clock oscillation in 1 min cycle (reversing every half cycle. The first half cycle is 0 during the operation of the user program)	R	√	√	√	√
SM14	1 hour clock	Clock oscillation in 1 hour cycle (reversing every half cycle. The first half cycle is 0 during the operation of the user program)	R	√	√	√	√

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM15	Scan cycle oscillation bit	This bit is reversed once every scan cycle (The first cycle is 0 during the operation of the user program)	R	√	√	√	√
SM16	High-speed ring counter enable flag bit	Unit: 0.1 ms, 16-bit Set: The high-speed ring counter starts counting Reset: The high-speed ring counter stops counting	R/W	√			√

3. User program execution error

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM20	Instruction execution error	This bit is set upon the instruction execution error, and the corresponding error type code is written into SD20. This bit is reset after the application instruction is correctly executed.	R	√	√	√	√
SM21	Instruction element number subscript overflow	This bit is set upon the instruction execution error, and the corresponding error type code is written into SD20.	R	√	√	√	√
SM22	Instruction parameter illegal	This bit is set upon the instruction execution error, and the corresponding error type code is written into SD20. This bit is reset after the application instruction is correctly executed.	R	√	√	√	√
SM30	Instruction execution completion flag	This bit is connected when the MODRW instruction action is completed.	R	√			√

4. Interrupt control

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM40	X0 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM41	X1 input rising/falling edge interrupt flag bit	Enable when set as 1	R/W	√	√	√	√
SM42	X2 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM43	X3 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM44	X4 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM45	X5 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM46	X6 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM47	X7 input rising/falling edge interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM48	COM 0 character transmission interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM49	COM 0 character receiving interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM50	COM 0 frame transmission interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM51	COM 0 frame receiving interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM52	COM 1 character transmission interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM53	COM 1 character receiving interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM54	COM 1 frame transmission interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM55	COM 1 frame receiving interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM56	AC power loss interrupt	Enable when set as 1	R/W	√	√	√	√
SM57	COM 2 character transmission interrupt enable flag bit	Enable when set as 1	R/W		√	√	√
SM58	COM 2 character receiving interrupt enable flag bit	Enable when set as 1	R/W		√	√	√
SM59	COM 2 frame transmission interrupt enable flag bit	Enable when set as 1	R/W		√	√	√
SM60	COM 2 frame receiving interrupt enable flag bit	Enable when set as 1	R/W		√	√	√
SM61	Enable flag bit for position-based interrupt 0 in the positioning instruction	Enable when set as 1	RW			√	
SM62	Enable flag bit for position-based interrupt 1 in the positioning instruction	Enable when set as 1	RW			√	
SM63	PTO (Y0) output completion interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM64	PTO (Y1) output completion interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM65	High-speed counter interrupt enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM66	Timed interrupt 0 enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM67	Timed interrupt 1 enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM68	Timed interrupt 2 enable flag bit	Enable when set as 1	R/W	√	√	√	√
SM69	Interpolation completion interrupt 1 enable flag bit	Enable when set as 1	R/W			√	
SM72	PTO (Y2) output completion interrupt enable flag bit	Enable when set as 1	R/W			√	√
SM73	PTO (Y3) output completion interrupt enable flag bit	Enable when set as 1	R/W			√	√
SM74	PTO (Y4) output completion interrupt enable flag bit	Enable when set as 1	R/W			√	
SM75	PTO (Y5) output completion interrupt enable flag bit	Enable when set as 1	R/W			√	
SM76	PTO (Y6) output completion interrupt enable flag bit	Enable when set as 1	R/W			√	
SM77	PTO (Y7) output completion interrupt enable flag bit	Enable when set as 1	R/W			√	
SM78	Interpolation completion interrupt 2 enable flag bit	Enable when set as 1	R/W			√	
SM79	Interpolation completion interrupt 3 enable flag bit	Enable when set as 1	R/W			√	
SM105	Enable flag bit for position-based interrupt 2 in the positioning instruction	Enable when set as 1	R/W			√	
SM106	Enable flag bit for position-based interrupt 3 in the positioning instruction	Enable when set as 1	R/W			√	

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM107	Enable flag bit for position-based interrupt 4 in the positioning instruction	Enable when set as 1	R/W			√	
SM108	Enable flag bit for position-based interrupt 5 in the positioning instruction	Enable when set as 1	R/W			√	
SM98	Enable flag bit for position-based interrupt 6 in the positioning instruction	Enable when set as 1	R/W			√	
SM99	Enable flag bit for position-based interrupt 7 in the positioning instruction	Enable when set as 1	R/W			√	

5. Peripheral instruction

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM70	Printing mode selection	When this bit is set to 1, there are 1-16 characters. When it is set to 0, there areregularly 8 characters.	R/W		√	√	
SM71	Printing	When this bit is set to 1, the printing is in progress	R		√	√	

6. High-speed pulse output control

Address	Name	Function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM80	Y000 pulse output stop instruction	When this bit is set to on, Y000 pulse is disabled	R/W	√	√	√	√
SM81	Y001 pulse output stop instruction	When this bit is set to on, Y001 pulse is disabled	R/W	√	√	√	√
SM82	Y000 pulse output monitoring (busy/ready)	This bit is set to on when Y000 outputs pulse	R	√	√	√	√
SM83	Y001 pulse output monitoring (busy/ready)	This bit is set to on when Y001 outputs pulse	R	√	√	√	√
SM84	Enabling the output through PWM instruction with the unit of us	When this bit is set to ON, the unit of the output through PWM instruction is us.	R/W	√		√	√
SM85	Enabling the clear function	When this bit is set to ON, the ZRN CLR signal output is enabled.	R/W	√			
SM86	Enabling Y00 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprograms. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow.	R/W	√		√	√
SM87	Enabling Y001 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprograms. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow.	R/W	√		√	√
SM88	Envelop line cyclic execution	When it is ON, envelope line executes cyclically	R/W				
SM89	PLSV gradual frequency conversion	When it is ON, the frequency changes gradually.	R/W	√		√	√
SM382	Enabling Y002 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprograms. If the	R/W			√	√

		instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow.					
SM383	Enabling Y003 pulse output when driven by an interrupt program	When this bit is set to ON, the PLSY instruction can be called in the interrupt program and subprograms. If the instruction is called in the main program, it is driven repeatedly and continuously along with the energy flow.	R/W			√	√

7. Pulse capture monitoring bit

Address	Name	Function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM90	Input X0 pulse capture monitoring bit	(1) All the elements in the table are cleared when the PLC changes from STOP to RUN; (2) The pulse capture is valid when the HCNT and SPD instructions are being executed at the same input port. For details, refer to section 6.10.9 "SPD: frequency measurement instruction" and section 6.10.1 "HCNT: high-speed counter drive instruction".	R/W	√	√		√
SM91	Input X0 pulse capture monitoring bit		R/W	√	√		√
SM92	Input X0 pulse capture monitoring bit		R/W	√	√		√
SM93	Input X0 pulse capture monitoring bit		R/W	√	√		√
SM94	Input X0 pulse capture monitoring bit		R/W	√	√		√
SM95	Input X0 pulse capture monitoring bit		R/W	√	√		√
SM96	Input X0 pulse capture monitoring bit		R/W	√	√		√
SM97	Input X0 pulse capture monitoring bit		R/W	√	√		√

Note:
1. All the elements in the table are cleared when the PLC changes from STOP to RUN. The pulse capture is valid when the HCNT and SPD instructions are being executed at the same input port. For details, refer to section 6.10.9"SPD: frequency measurement instruction"and section 6.10.1 "HCNT: high-speed counter drive instruction".
2. For hardware counters, the total pulse frequency input through X0–X7 (using pulse capture, SPD instruction or HCNT, but no high-speed comparison instruction) does not exceed 80 k. For software, that frequency (using the DHSCS, DHSCI, DHSZ, DHSP, or DHST instructions for the driven high-speed counters) does not exceed 30 k.

8. Quad-frequency

Address	Name	function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM100	Switching between one/four times of AB phase input of X0 and X1	Clearing when from STOP —>RUN;	R/W	√	√	√	√
SM101	Switching between one/four times of AB phase input of X2 and X3		R/W			√	
SM102	Switching between one/four times of AB phase input of X3 and X4		R/W		√	√	√
SM103	Switching between one/four times of AB phase input of X4 and X5		R/W			√	
SM104	Switching between one/four times of AB phase input of X6 and X7		R/W			√	

9. Free port (PORT0)

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM110	PORT0 transmission enable flag bit	This bit is set when the XMT instruction is used, and cleared after the transmission is over. You can manually clear this bit to suspend the current transmission at PORT0. The transmission can	R/W	√	√	√	√

		continue when the energy flow is on again					
SM111	PORT0 receiving enable flag bit	This bit is set when the RCV instruction is used, and cleared after the transmission is over. You can manually clear this bit to suspend the current transmission at PORT0. The transmission can continue when the energy flow is on again	R/W	√	√	√	√
SM112	PORT0 transmission completion flag bit	This bit is set after the transmission is over	R/W	√	√	√	√
SM113	PORT0 receiving completion flag bit	This bit is set after the receiving is over	R/W	√	√	√	√
SM114	PORT0 idle flag bit	This bit is set when the port is idle	R	√	√	√	√

Note

SM112-SM114 are flagsfor receiving, completion and idle states in all communication protocols that are supported by PORT0. For example, PORT0 of IVC1 series PLCssupports N:N, Modbus and Free-port protocols. No matter what protocol is used, the functions of M112-SM114 remain the same.

10. Free port (PORT1)

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM120	PORT1 transmission enable flag bit	This bit is set when the XMT instruction is used, and cleared after the transmission is over. You can manually clear this bit to suspend the current transmission at PORT1. The transmission can continue when the energy flow is on again	R/W	√	√	√	√
SM121	PORT1 receiving enable flag bit	This bit is set when the RCV instruction is used, and cleared after the transmission is over. You can manually clear this bit to suspend the current transmission at PORT1. The transmission can continue when the energy flow is on again	R/W	√	√	√	√
SM122	PORT1 transmission completion flag bit	This bit is set after the transmission is over	R/W	√	√	√	√
SM123	PORT1 receiving completion flag bit	This bit is set after the receiving is over	R/W	√	√	√	√
SM124	PORT1 idle flag bit	This bit is set when the port is idle	R	√	√	√	√

Note

SM122-SM124 are flags for receiving, completion and idle states in all communication protocols that are supported by PORT1. For example, PORT1 of IVC1 series PLCs supports N:N, Modbus and Free-port protocols. No matter what protocol is used, the functions of M122-SM124 remain the same.

11. Extended free port (PORT2)

Address	Name	Action and function	R/W	IVC2L	IVC3	IVC1L
SM130	PORT2 transmission enable flag bit	This bit is set when the XMT instruction is used, and cleared after the transmission is over. You can manually clear this bit to suspend the current transmission at PORT2. The transmission can continue when the energy flow is on again	R/W	√	√	√

Address	Name	Action and function	R/W	IVC2L	IVC3	IVC1L
SM131	PORT1 receiving enable flag bit	This bit is set when the RCV instruction is used, and cleared after the transmission is over. You can manually clear this bit to suspend the current transmission at PORT2. The transmission can continue when the energy flow is on again	R/W	√	√	√
SM132	PORT2 transmission completion flag bit	This bit is set after the transmission is over	R/W	√	√	√
SM133	PORT2 receiving completion flag bit	This bit is set after the receiving is over	R/W	√	√	√
SM134	PORT2 idle flag bit	This bit is set when the port is idle	R	√	√	√

Note

SM132-SM134 are flags for receiving, completion and idle states in all communication protocols that are supported by PORT2. For example, 485 communication module of IVC2L series PLCsupports Modbus and Free-port protocols. No matter what protocol is used, the functions of M132-SM134 remain the same.

12. Modbus communication

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM135	PORT1 Modbus communication completed	This bit is set after the communication is over	R/W	√	√	√	√
SM136	PORT1 Modbus communication error	This bit is set upon the communication error	R/W	√	√	√	√
SM137	PORT2 Modbus communication completed	This bit is set after the communication is over	R/W	√	√	√	√
SM138	PORT2 Modbus communication error	This bit is set upon the communication error	R/W	√	√	√	√

13. N:N communication

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM140	Station 0 communication error flag		R	√	√	√	√
SM141	Station 1 communication error flag		R	√	√	√	√
SM142	Station 2 communication error flag		R	√	√	√	√
SM143	Station 3 communication error flag		R	√	√	√	√
SM144	Station 4 communication error flag		R	√	√	√	√
SM145	Station 5 communication error flag		R	√	√	√	√
SM146	Station 6 communication error flag		R	√	√	√	√
SM147	Station 7 communication error flag		R	√	√	√	√
SM148	Station 8 communication error flag		R	√	√	√	√
SM149	Station 9 communication error flag		R	√	√	√	√
SM150	Station 10 communication error flag		R	√	√	√	√
SM151	Station 11 communication error flag		R	√	√	√	√
SM152	Station 12 communication error flag		R	√	√	√	√
SM153	Station 13 communication		R	√	√	√	√

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
	error flag						
SM154	Station 14 communication error flag		R	√	√	√	√
SM155	Station 15 communication error flag		R	√	√	√	√
SM156	Station 16 communication error flag		R	√	√	√	√
SM157	Station 17 communication error flag		R	√	√	√	√
SM158	Station 18 communication error flag		R	√	√	√	√
SM159	Station 19 communication error flag		R	√	√	√	√
SM160	Station 20 communication error flag		R	√	√	√	√
SM161	Station 21 communication error flag		R	√	√	√	√
SM162	Station 22 communication error flag		R	√	√	√	√
SM163	Station 23 communication error flag		R	√	√	√	√
SM164	Station 24 communication error flag		R	√	√	√	√
SM165	Station 25 communication error flag		R	√	√	√	√
SM166	Station 26 communication error flag		R	√	√	√	√
SM167	Station 27 communication error flag		R	√	√	√	√
SM168	Station 28 communication error flag		R	√	√	√	√
SM169	Station 29 communication error flag		R	√	√	√	√
SM170	Station 30 communication error flag		R	√	√	√	√
SM171	Station 31 communication error flag		R	√	√	√	√

14. Enable flag of integrated analog channel

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM172	Enable flag of AD channel 1	Sampling at AD channel 1 is enabled when this bit is set to 1	R/W	√			√
SM173	Enable flag of AD channel 2	Sampling at AD channel 2 is enabled when this bit is set to 1	R/W	√			√
SM174	Voltage/current enable flag of AD channel 1	This bit is set to 1 for current input and 0 for voltage input	R/W	√			√
SM175	Voltage/current enable flag of AD channel 2	This bit is set to 1 for current input and 0 for voltage input	R/W	√			√
SM176	Enable flag of AD channel 3	Sampling at AD channel 3 is enabled when this bit is set to 1	R/W	√			√
SM177	Voltage/current enable flag of AD channel 3	This bit is set to 1 for current input and 0 for voltage input	R/W	√			√
SM34	Enable flag of AD channel 4	Sampling at AD channel 4 is enabled when this bit is set to 1	R/W	√			√
SM35	Voltage/current enable flag of AD channel 4	This bit is set to 1 for current input and 0 for voltage input	R/W	√			√
SM36	Enable flag of AD channel 5	Sampling at AD channel 5 is enabled when this bit is set to 1	R/W	√			√
SM37	Voltage/current enable flag of	This bit is set to 1 for current input	R/W	√			√

	AD channel 5	and 0 for voltage input					
SM38	Enable flag of AD channel 6	Sampling at AD channel 6 is enabled when this bit is set to 1	R/W	√			√
SM39	Voltage/current enable flag of AD channel 6	This bit is set to 1 for current input and 0 for voltage input	R/W	√			√
SM178	Enable flag of DA channel 1	Sampling at DA channel 1 is enabled when this bit is set to 1	R/W	√			√

15. Operation flag bit

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM180	Zero flag bit	This bit is set when the related calculated result is zero. You can clear this bit manually	R/W	√	√	√	√
SM181	Carry/overflow flag bit	This bit is set when the related calculated result is a carry. You can clear this bit manually	R/W	√	√	√	√
SM182	Borrow flag bit	This bit is set when the related calculated result is a borrow. You can clear this bit manually	R/W	√	√	√	√
SM185	Table comparison flag	This bit is set when the whole table record is completed	R/W	√	√	√	√
SM188	Data block comparison set	This bit is set when the comparison results in the data block are 1	R/W			√	

16. ASCII code conversion instruction flag

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM186	ASC instruction storage mode flag	0: MSB/LSB of each word stores one ASCII code 1: LSB of each word stores one ASCII code	R/W	√	√	√	√
SM187	Instruction execution completed	This bit is set to ON after the initial cyclic action of the MTR instruction	R				

17. System bus error flag

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM190	Main module bus error flag bit	1. This bit is reset upon correct powering-on addressing 2. This bit is reset when the PLC changes from STOP to RUN 3. This bit is reset when a new program is downloaded 4. This bit incurs system stop	R	√	√	√	√
SM191	General module bus error flag bit	1. This bit is set and the system raises an alarm when a general module bus operation error occurs 2. This bit is reset automatically when the system error is removed	R	√	√	√	√
SM192	Special module bus error flag bit	1. This bit is set and the system raises an alarm when a special module bus operation error occurs 2. This bit is reset automatically when the system error is removed	R	√	√	√	√

18. Real-time clock error flag

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM193	R/W real-time clock error	This bit is set upon a real-time clock error This bit is automatically cleared if the system error is removed	R	√	√	√	√

19. Memory card connection flag bit

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM197	Whether memory card is connected	This bit is set to 1 when a memory card is connected, and reset to 0 when no memory card is connected.	R				

20. Count direction of bidirectional counters

Address no.	Corresponding counter address no.	Function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM200	C200	When SM2 _ _ is of high level, its corresponding C2_ _ becomes a decrement counter When SM2 _ _ is of low level, its corresponding C2_ _becomes an increment counter	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	√	√	√	√
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 of high level, its corresponding C2_ _ becomes a decrement counter When SM2 _ _ is of low level, its corresponding C2_ _becomes an increment counter	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	√	√	√	√

21. Count direction and monitoring of high-speed counters

Differentiation	Address no.	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L
Single-phase one point	SM236	C236	The high and low levels of SM2 _ _correspond to counting up and down of the counter respectively When C2 _ _ of single-phase bidirectional count input counter and two-phase count input counter is in down mode, its corresponding SM2 _ _ changes	R/W	√	√	√	√
	SM237	C237		R/W	√	√	√	√
	SM238	C238		R/W	√	√	√	√
	SM239	C239		R/W	√	√	√	√
	SM240	C240		R/W	√	√	√	√

Differentiation	Address no.	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L
count input	SM241	C241	to high level. When it is in up mode, its corresponding SM2 _ _ changes to low level.	R/W	√	√	√	√
	SM301	C301		R/W			√	
	SM302	C302		R/W			√	
	SM242	C242		R/W	√	√	√	√
	SM243	C243		R/W	√	√	√	√
	SM244	C244		R/W	√	√	√	√
	SM245	C245		R/W	√	√	√	√
Single-phase bidirectional count input	SM246	C246	Register content The high and low levels of SM2 _ _correspond to counting up and down of the counter respectively	R	√	√	√	√
	SM247	C247		R	√	√	√	√
	SM303	C303		R			√	
	SM248	C248		R	√	√	√	√
	SM249	C249		R	√	√	√	√
	SM250	C250		R	√	√	√	√
Two-phase count input	SM251	C251		R	√	√	√	√
	SM304	C304		R			√	
	SM305	C305		R			√	
	SM306	C306		R			√	
	SM252	C252		R	√	√	√	√
	SM253	C253		R	√	√	√	√
	SM254	C254		R	√	√	√	√
	SM255	C255		R	√	√	√	√

22. Enhanced positioning

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM260	Enabling the input-based interrupt function	Y0 and Y1 are applicable to DVIT. When this function is disabled, Y0 corresponds to X0 and Y1 corresponds to X1 in the interruption. When this function is enabled, the bits specified in the input are set to ON, and every 4 bits of SD240 correspond to one output (Y).	R/W		√		√
SM262	Y002 pulse output stop instruction	When this bit is set, Y002 pulse is disabled	R/W			√	√
SM263	Y003 pulse output stop instruction	When this bit is set, Y003 pulse is disabled	R/W			√	√
SM264	Y004 pulse output stop instruction	When this bit is set, Y004 pulse is disabled	R/W			√	
SM265	Y005 pulse output stop instruction	When this bit is set, Y005 pulse is disabled	R/W			√	
SM266	Y006 pulse output stop instruction	When this bit is set, Y006 pulse is disabled	R/W			√	
SM267	Y007 pulse output stop instruction	When this bit is set, Y007 pulse is disabled	R/W			√	
SM272	Y002 pulse output monitoring (busy/ready)	This bit is set when Y002 outputs pulse	R			√	√
SM273	Y003 pulse output monitoring (busy/ready)	This bit is set when Y003 outputs pulse	R			√	√
SM274	Y004 pulse output monitoring (busy/ready)	This bit is set when Y004 outputs pulse	R			√	
SM275	Y005 pulse output monitoring (busy/ready)	This bit is set when Y005 outputs pulse	R			√	
SM276	Y006 pulse output monitoring (busy/ready)	This bit is set when Y006 outputs pulse	R			√	
SM277	Y007 pulse output	This bit is set when Y007 outputs pulse	R			√	

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
	monitoring (busy/ready)						
SM280	Enabling the clear function	The CLR signal output function Y0 of the ZRN instruction is valid.	R/W		√	√	√
SM281	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD206 is used to indicate the clear signal. If not specified, Y0 is Y10 by default, which is applicable to DSZR and ZRN.	R/W		√	√	√
SM282	The ZRN direction	Y0 is applicable to DSZR	R/W		√	√	√
SM283	FWD limit	Y0 is applicable to DSZR/DVIT	R/W		√	√	√
SM284	REV limit	Y0 is applicable to DSZR/DVIT	R/W		√	√	√
SM285	Logic reversal of the DOG signal	Y0 is applicable to DSZR	R/W		√	√	√
SM286	Logic reversal of the zero signal	Y0 is applicable to DSZR	R/W		√	√	√
SM287	Logic reversal of the interrupt signal	Y0 is applicable to DVIT, but not applicable to the user interrupt input instruction	R/W		√	√	√
SM288	The positioning instruction is driving	Y0 is applicable to DSZR/DVIT	R/W		√	√	√
SM289	User interrupt input instruction	Y0 is applicable to DVIT	R/W		√	√	√
SM290	Enabling the clear function	Y1 is applicable to DSZR/ZRN	R/W		√	√	√
SM291	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD216 is used to indicate the clear signal. If not specified, Y1 is Y11 by default, which is applicable to DSZR and ZRN.	R/W		√	√	√
SM292	The ZRN direction	Y1 is applicable to DSZR	R/W		√	√	√
SM293	FWD limit	Y1 is applicable to DSZR/DVIT	R/W		√	√	√
SM294	REV limit	Y1 is applicable to DSZR/DVIT	R/W		√	√	√
SM295	Logic reversal of the DOG signal	Y1 is applicable to DSZR	R/W		√	√	√
SM296	Logic reversal of the zero signal	Y1 is applicable to DSZR	R/W		√	√	√
SM297	Logic reversal of the interrupt signal	Y1 is applicable to DVIT	R/W		√	√	√
SM298	The positioning instruction is driving	Y1 is applicable to DSZR/DVIT	R/W		√	√	√
SM299	User interrupt input instruction	Y1 is applicable to DVIT	R/W			√	√
SM320	Enabling the clear function	Y2 is applicable to DSZR/ZRN	R/W			√	√
SM321	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD326 is used to indicate the clear signal. If not specified, Y2 is Y12 by default, which is applicable to DSZR.	R/W			√	√
SM322	The ZRN direction	Y2 is applicable to DSZR	R/W			√	√
SM323	FWD limit	Y2 is applicable to DSZR/DVIT	R/W			√	√
SM324	REV limit	Y2 is applicable to DSZR/DVIT	R/W			√	√
SM325	Logic reversal of the DOG signal	Y2 is applicable to DSZR	R/W			√	√
SM326	Logic reversal of the zero signal	Y2 is applicable to DSZR	R/W			√	√
SM327	Logic reversal of the interrupt signal	Y2 is applicable to DVIT	R/W				√
SM328	The positioning instruction is driving	Y2 is applicable to DSZR/DVIT	R/W			√	√

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM329	User interrupt input instruction	Y2 is applicable to DVIT	R/W				√
SM330	Enabling the clear function	Y3 is applicable to DSZR/ZRN	R/W			√	√
SM331	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD336 is used to indicate the clear signal. If not specified, Y3 is Y13 by default, which is applicable to DSZR and ZRN.	R/W			√	√
SM332	The ZRN direction	Y3 is applicable to DSZR	R/W			√	√
SM333	FWD limit	Y3 is applicable to DSZR/DVIT	R/W			√	√
SM334	REV limit	Y3 is applicable to DSZR/DVIT	R/W			√	√
SM335	Logic reversal of the DOG signal	Y3 is applicable to DSZR	R/W			√	√
SM336	Logic reversal of the zero signal	Y3 is applicable to DSZR	R/W			√	√
SM337	Logic reversal of the interrupt signal	Y3 is applicable to DVIT	R/W			√	√
SM338	The positioning instruction is driving	Y3 is applicable to DSZR/DVIT	R/W			√	
SM339	User interrupt input instruction	Y3 is applicable to DVIT	R/W				√
SM341	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD346 is used to indicate the clear signal. If not specified, Y4 is Y14 by default, which is applicable to DSZR and ZRN.	R/W			√	
SM342	The ZRN direction	Y4 is applicable to DSZR	R/W			√	
SM343	FWD limit	Y4 is applicable to DSZR/DVIT	R/W			√	
SM344	REV limit	Y4 is applicable to DSZR/DVIT	R/W			√	
SM345	Logic reversal of the DOG signal	Y4 is applicable to DSZR	R/W			√	
SM346	Logic reversal of the zero signal	Y4 is applicable to DSZR	R/W			√	
SM347	Logic reversal of the interrupt signal	Y4 is applicable to DVIT	R/W			√	
SM348	The positioning instruction is driving	Y4 is applicable to DSZR/DVIT	R/W			√	
SM350	Enabling the clear function	Y5 is applicable to DSZR/ZRN	R/W			√	
SM351	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD356 is used to indicate the clear signal. If not specified, Y5 is Y15 by default, which is applicable to DSZR and ZRN.	R/W			√	
SM352	The ZRN direction	Y5 is applicable to DSZR	R/W			√	
SM353	FWD limit	Y5 is applicable to DSZR/DVIT	R/W			√	
SM354	REV limit	Y5 is applicable to DSZR/DVIT	R/W			√	
SM355	Logic reversal of the DOG signal	Y5 is applicable to DSZR	R/W			√	
SM356	Logic reversal of the zero signal	Y5 is applicable to DSZR	R/W			√	
SM357	Logic reversal of the interrupt signal	Y5 is applicable to DVIT	R/W			√	
SM360	Enabling the clear function	Y6 is applicable to DSZR/ZRN	R/W			√	
SM361	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD366 is used to indicate the clear signal. If not specified, Y6 is Y16	R/W			√	

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
		by default, which is applicable to DSZR.					
SM362	The ZRN direction	Y6 is applicable to DSZR	R/W			√	
SM363	FWD limit	Y6 is applicable to DSZR/DVIT	R/W			√	
SM364	REV limit	Y6 is applicable to DSZR/DVIT	R/W			√	
SM365	Logic reversal of the DOG signal	Y6 is applicable to DSZR	R/W			√	
SM366	Logic reversal of the zero signal	Y6 is applicable to DSZR	R/W			√	
SM370	Enabling the clear function	Y7 is applicable to DSZR/ZRN	R/W			√	
SM371	The element specified by the clear signal is valid	Y element Y (N) corresponding to the value N in SD376 is used to indicate the clear signal. If not specified, Y7 is Y17 by default, which is applicable to DSZR and ZRN.	R/W			√	
SM372	The ZRN direction	Y7 is applicable to DSZR	R/W			√	
SM373	FWD limit	Y7 is applicable to DSZR/DVIT	R/W			√	
SM374	REV limit	Y7 is applicable to DSZR/DVIT	R/W			√	
SM375	Logic reversal of the DOG signal	Y7 is applicable to DSZR	R/W			√	
SM376	Logic reversal of the zero signal	Y7 is applicable to DSZR	R/W			√	

23. Signal alarm

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM400	Signal alarm is valid	After setting SM400 to ON, the following SM401 and SD401 work	R/W			√	
SM401	Signal alarm action	Setting SM401 to ON when there is any action in state S900-S999	R/W			√	

24. Timed output instruction

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM430	Timer clock output 1	For use by the DUTY instruction	R/W			√	
SM431	Timer clock output 2	For use by the DUTY instruction	R/W			√	
SM432	Timer clock output 3	For use by the DUTY instruction	R/W			√	
SM433	Timer clock output 4	For use by the DUTY instruction	R/W			√	
SM434	Timer clock output 5	For use by the DUTY instruction	R/W			√	

25. CANopen instruction

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SM440	CANopen instruction completed				√	√	
SM441	CANopen instruction is wrong				√	√	
SM442	CANopen instruction is running				√	√	

Appendix B Special data register

Note

1. All special data registers except SD50 - SD55 are initialized when the PLC changes from STOP to RUN.
2. The reserved SD and SM elements are not listed in the table. The reserved SD elements are by default read only.

1. PLC working state data

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD00	PLC type	20 indicates IVC2L 10 indicates IVC1, 110 indicates IVC1A, 112 indicates IVC1L 111 indicates EVC－UPRG 30 indicates IVC3	R	√	√	√		
SD01	Version no.	For example, 100 indicates 1.00	R	√	√	√	√	
SD02	User program capacity	For example, 8 indicates an 8 k step program	R	√	√	√	√	
SD03	System error code	Storing the occurred system error codes	R	√	√	√	√	
SD04	Battery voltage value	Taking 0.1 V as unit, and 3.6V indicates 36	R		√	√	√	
SD07	Number of extension I/O modules		R	√	√	√	√	
SD08	Number of special modules		R	√	√	√	√	
SD09	Setting the input points for operation control. Decimal (X0 is displayed as 0 and X10 is displayed as 8. Maximum: 15) (Configurable through the system block only)		R	√	√	√	√	0-15
SD10	Number of main module I/O points	MSB: input; LSB: output	R	√	√	√	√	
SD11	Number of extension module I/O points	MSB: input; LSB: output	R	√	√	√	√	
SD12	Number of main module analog I/O points	MSB: input; LSB: output	R	√		√	√	
SD16	High-speed ring counter	0-20971 (Unit: 0.1 ms, 16-bit) increment ring counter.	R/W	√				0-32767

2. Operation error code FIFO area

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD20	Reserved operation error code 0	In the order of arrival, the latest five operation error codes are reserved. SD20 always stores the latest error codes.	R	√	√	√	√	
SD21	Reserved operation error code 1		R	√	√	√	√	
SD22	Reserved operation error code 2		R	√	√	√	√	
SD23	Reserved operation error code 3		R	√	√	√	√	

SD24	Reserved operation error code 4		R	√	√	√	√	

3. FROM/TO error

Address	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD25	Special module's numbering is wrong (starts from 0) when the FROM/TO instruction is used	R	√	√	√	√	Initial value: 255
SD26	The I/O chip's numbering is wrong (starts from 0) when I/O is refreshed	R	√	√	√	√	Initial value: 255

4. Scan time

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD30	Current scan value	Current scan time (unit: ms)	R	√	√	√	√	
SD31	Min. scan time	Min. scan time (unit: ms)	R	√	√	√	√	
SD32	Max. scan time	Max. scan time (unit: ms)	R	√	√	√	√	
SD33	Set value of constant scan time	Initial value: 0ms (configurable only through the system block), Unit: 1 ms. When the constant scan time is longer than the set value of user monitoring timeout, the user program timeout alarm is raised. When a scan cycle of the user program is longer than the constant scan time, the constant scan mode of this cycle is invalid automatically and no alarm is raised. SD33 is regarded as 1000 ms when it is set bigger than 1000 ms.	R	√	√	√	√	0–1000 ms
SD34	Set value of user program timeout	Initial value:200 ms (configurable only through the system block). SD34 is regarded as 100 when it is set smaller than 100. SD34 is regarded as 1000 when it is set smaller than 1000.	R	√	√	√	√	100–1000 ms

Note

1. The error tolerance of SD30, SD31 and SD32 is 1ms.
2. It is recommended to set the user program overtime (SD34) at least 5ms bigger than the constant scan time (SD33). Otherwise,when the value of SD33 closes to SD34, the system is apt to report user program overtime errordue to the influence of the system operation and user program.

5. Input filtering constant setup

Address	Name	Action and function	R/W	IVC1	IVC2L	IVC3	IVC1L
SD35	Input filtering constant	(configurable only through the system block)	R	0,2, 4,8, 16,32,64 ms	0-60 ms		0,2,4,8, 16,32,64 ms
SD36	Input filtering constant	(configurable only through the system block)	R	0,2, 4,8, 16,32,64 ms	-	-	0,2,4,8, 16,32,64 ms

Note

1. IVC2 has only one group of filtering, SD35, and the supported input ports is X0–X17.
2. IVC1 and IVC1S are divided into two groups of filtering: SD35 and SD36, and their supported inputs are X0–X3 and X4–X7 respectively. Range: 0,2,4,8,16,32, and 64 ms.

3. IVC1 and IVC1S are divided into two groups of filtering: SD35 and SD36, and their supported inputs are X0–X3 and X4–X7 respectively. Range: 0 to 60 ms.
4. IVC3 supports to independently set the filtering parameters for each port (X0–X7), ranging from 0us to 60ms.

6. High-speed pulse output monitoring

Address	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD50	Accumulated total number of high-speed pulses outputted through Y0 (MSB)	R/W	√	√	√	√	
SD51	Accumulated total number of high-speed pulses outputted through Y0 (LSB)	R/W	√	√	√	√	
SD52	Accumulated total number of high-speed pulses outputted through Y1 (MSB)	R/W	√	√	√	√	
SD53	Accumulated total number of high-speed pulses outputted through Y1 (LSB)	R/W	√	√	√	√	
SD54	Accumulated total number of high-speed pulses outputted throughY1 and Y0 (MSB)	R/W	√	√			
SD55	Accumulated total number of high-speed pulses outputted through Y1 and Y0 (LSB)	R/W	√	√			
SD56	Number of segment that is being outputted through Y0 by the PLS instruction	R	√		√	√	
SD57	Number of segment that is being outputted through Y1 by the PLS instruction	R	√		√	√	
SD160	Total number of pulses outputted through Y2 by the PLSR/PLSY instruction (MSB)	R/W			√	√	
SD161	Total number of pulses outputted through Y2 by the PLSR/PLSY instruction (LSB)	R/W			√	√	
SD162	Total number of pulses outputted through Y3 by the PLSR/PLSY instruction (MSB)	R/W			√	√	
SD163	Total number of pulses outputted through Y3 by the PLSR/PLSY instruction (LSB)	R/W			√	√	
SD164	Total number of pulses outputted through Y4 by the PLSR/PLSY instruction (MSB)	R/W			√		
SD165	Total number of pulses outputted through Y4 by the PLSR/PLSY instruction (LSB)	R/W			√		
SD166	Total number of pulses outputted through Y5 by the PLSR/PLSY instruction (MSB)	R/W			√		
SD167	Total number of pulses outputted through Y5 by the PLSR/PLSY instruction (LSB)	R/W			√		
SD168	Total number of pulses outputted through Y6 by the PLSR/PLSY instruction (MSB)	R/W			√		
SD169	Total number of pulses outputted through Y6 by the PLSR/PLSY instruction (LSB)	R/W			√		
SD170	Total number of pulses outputted through Y7 by the PLSR/PLSY instruction (MSB)	R/W			√		
SD171	Total number of pulses outputted through Y7 by the PLSR/PLSY instruction (LSB)	R/W			√		
SD252	Number of segment that is being outputted through Y2 by the PLS instruction	R			√	√	
SD253	Number of segment that is being outputted through Y3 by the PLS instruction	R				√	
SD254	Number of segment that is being outputted through Y4 by the PLS instruction	R			√		
SD255	Number of segment that is being outputted through Y5 by the PLS instruction	R			√		
SD256	Number of segment that is being outputted through Y6 by the PLS instruction	R			√		
SD257	Number of segment that is being outputted through Y7 by the PLS instruction	R			√		

7. Timed interrupt cycle

Address	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD66	Cycle of timed interrupt 0	The interrupt is not triggered when the value is not within 1–32767	R/W	√	√	√	√	1–32767 ms
SD67	Cycle of timed interrupt 1	The interrupt is not triggered when the value is not within 1–32767	R/W	√	√	√	√	1–32767 ms
SD68	Cycle of timed interrupt 2	The interrupt is not triggered when the value is not within 1–32767	R/W	√	√	√	√	1–32767 ms

An error of ±1ms may occur when the system processes a user timed interrupt. To ensure the normal operation of the timed interrupt, it is recommended to set the cycle of timed interrupts to be bigger or equal to 5ms.

8. Positioning instruction

Address	Name	Initial value	Function	R/W	IVC1	IVC2L	IVC3	IVC1L
SD80	Current position value in the positioning instruction outputted through Y0 (MSB)	0	Used as the current value data register of Y000 output positioning instruction.	R/W	√			
SD81	Current position value in the positioning instruction outputted through Y0 (LSB)			R/W	√			
SD82	Current position value in the positioning instruction outputted through Y1 (MSB)	0	Used as the current value data register of Y001 output positioning instruction.	R/W	√			
SD83	Current position value in the positioning instruction outputted through Y1 (LSB)			R/W	√			
SD84	The base speed at which the ZRN, DRVI and DRVA instructions are executed	5000	Base speed at which the ZRN, PLSV, DRVI and DRVA instructions are executed (below 1/10 of the max. speed)	R/W	√			
SD85	Max. speed at which the ZRN, DRVI and DRVA instructions are executed (MSB)	100000	Max. speed at which the ZRN,PLSV,DRVI and DRVA instructions are executed (10-100000)	R/W	√			
SD86	Max. speed at which the ZRN, DRVI and DRVA instructions are executed (LSB)			R/W	√			
SD87	Acceleration/deceleration time at which the ZRN, DRVI and DRVA instructions are executed	1000	Acceleration/deceleration timefor accelerating from the base speed (SD84) to the max. speed (SD85,SD86) when the ZRN,DRVI and DRVA instructions are executed (50 ms－5000 ms)	R/W	√			
SD240	Specified interrupt input	0	Specified interrupt input	R/W		√		√

9. Real-time clock

Address	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD100	Year	For real-time clock	R	√	√	√	√	2000–2099
SD101	Month	For real-time clock	R	√	√	√	√	1–12 month
SD102	Day	For real-time clock	R	√	√	√	√	1–31 day
SD103	Hour	For real-time clock	R	√	√	√	√	0–23 hour
SD104	Minute	For real-time clock	R	√	√	√	√	0–59 minute
SD105	Second	For real-time clock	R	√	√	√	√	0–59 second
SD106	Week	For real-time clock	R	√	√	√	√	0 (Sunday)–6 (Saturday)
You can set these elements only with the TWR instruction or through the host computer								

10. Free-port receiving control and state (PORT0)

<table>
<tr><th>Address</th><th colspan="2">Name</th><th>Register content</th><th>R/W</th><th>IVC1</th><th>IVC2L</th><th>IVC3</th><th>IVC1L</th><th>Range</th></tr>
<tr><td rowspan="8">SD110</td><td rowspan="8">Free-port 0 mode state word</td><td>SD110.0–SD110.2 Port baud rate</td><td>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</td><td rowspan="8">R</td><td rowspan="8">√</td><td rowspan="8">√</td><td rowspan="8">√</td><td rowspan="8">√</td><td rowspan="8"></td></tr>
<tr><td>SD110.3 Stop bit</td><td>0: 1 stop bit
1: 2 stop bits</td></tr>
<tr><td>SD110.4 Parity check</td><td>0: even parity; 1: odd parity</td></tr>
<tr><td>SD110.5 Parity check enabling</td><td>0: no parity check;
1: parity check</td></tr>
<tr><td>SD110.6 Character data bit</td><td>Data bit of each character
0: 8-bit character
1: 7-bit character</td></tr>
<tr><td>SD110.7 Free-port receiving start character mode</td><td>1: has specific start character
0: no specific start character</td></tr>
<tr><td>SD110.8 Free-port receiving end character mode</td><td>1: has specific end character
0: no specific end character</td></tr>
</table>

Address	Name		Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
		SD110.9 Free-port intercharacter timout enabling	1: has valid intercharacter timeout 0: no valid intercharacter timeout						
		SD110.10 Free-port interframe timeout enabling	1: has interframe timeout 0: no interframe timeout						
		SD110.11	Reserved						
		SD110.12 MSB/LSB valid	0: LSB of the word element is valid 1: MSB/LSB of the word element is valid						
		SD110.13–SD110.15	Reserved						
SD111	Start character			R/W	√	√	√	√	
SD112	End character			R/W	√	√	√	√	
SD113	Intercharacter timeout		Default: 0 ms (intercharacter timeout is omitted)	R/W	√	√	√	√	1–32767 ms
SD114	Frame timeout		Default: 0 ms (frame timeout is omitted)	R/W	√	√	√	√	1–32767 ms
SD115	Receiving completion message code		Bit 0: set when the receiving ends Bit 1: set when the specified end word is received Bit 2: set when the max. character number is received Bit 3: set upon intercharacter timeout Bit 4: set upon frame timout Bit 5: set upon the parity check error Bits 6-15: reserved	R	√	√	√	√	
SD116	Currently received characters			R	√	√	√	√	
SD117	Total number of currently received characters			R	√	√	√	√	
SD118	Currently transmitted characters			R	√		√	√	

11. Free-port receiving control and state (PORT1)

Address	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range

<table>
<tr><th>Address</th><th colspan="2">Name</th><th>Register content</th><th>R/W</th><th>IVC1</th><th>IVC2L</th><th>IVC3</th><th>IVC1L</th><th>Range</th></tr>
<tr><td rowspan="13">SD120</td><td rowspan="13">Free-port 1 mode state word</td><td>SD120.0 – SD120.2 Port baud rate</td><td>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</td><td rowspan="13">R/W</td><td rowspan="13">√</td><td rowspan="13">√</td><td rowspan="13">√</td><td rowspan="13">√</td><td rowspan="13"></td></tr>
<tr><td>SD120.3 Stop bit</td><td>0: 1 stop bit
1: 2 stop bits</td></tr>
<tr><td>SD120.4 Parity check</td><td>0: even parity
1: odd parity</td></tr>
<tr><td>SD120.5 Parity check enabling</td><td>0: no parity check
1: parity check</td></tr>
<tr><td>SD120.6 Character data bit</td><td>Data bit of each character
0: 8-bit character
1: 7-bit character</td></tr>
<tr><td>SD120.7 Free-port receiving start character mode</td><td>1: has specific start character
0: no specific start character</td></tr>
<tr><td>SD120.8 Free-port receiving end character mode</td><td>1: has specific end character
0: no specific end character</td></tr>
<tr><td>SD120.9 Free-port intercharacter timout enabling</td><td>1: has valid intercharacter timeout
0: no valid intercharacter timeout</td></tr>
<tr><td>SD120.10 Free-port interframe timeout enabling</td><td>1: has interframe timeout
0: no interframe timeout</td></tr>
<tr><td>SD120.11</td><td>Reserved</td></tr>
<tr><td>SD120.12 MSB/LSB valid</td><td>0: LSB of the word element is valid
1: MSB/LSB of the word element is valid</td></tr>
<tr><td>SD120.13–SD120.15</td><td>Reserved</td></tr>
<tr><td colspan="0" style="display:none"></td></tr>
<tr><td>SD121</td><td colspan="2">Start character</td><td></td><td>R/W</td><td>√</td><td>√</td><td>√</td><td>√</td><td></td></tr>
<tr><td>SD122</td><td colspan="2">End character</td><td></td><td>R/W</td><td>√</td><td>√</td><td>√</td><td>√</td><td></td></tr>
<tr><td>SD123</td><td colspan="2">Intercharacter</td><td>Default: 0 ms</td><td>R/W</td><td>√</td><td>√</td><td>√</td><td>√</td><td>0–32767ms</td></tr>
</table>

Address	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
	timeout	(intercharacter timeout is omitted)						
SD124	Frame timeout	Default: 0 ms (frame timeout is omitted)	R/W	√	√	√	√	0–32767ms
SD125	Receiving completion message code	Bit 0: set when the receiving ends Bit 1: set when the specified end word is received Bit 2: set when the max. character number is received Bit 3: set upon intercharacter timeout Bit 4: set upon frame timout Bit 5: set upon the parity check error Bits 6-15: reserved	R	√	√	√	√	
SD126	Currently received characters		R	√	√	√	√	
SD127	Total number of currently received characters		R	√	√	√	√	
SD128	Currently transmitted characters		R	√	√	√	√	

12. Modbus/N:N setup

Address	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD130	Local station No. (PORT0)	R/W	√	√	√	√	MOD (1–32), EMR (0–31)
SD131	Max. timeout time of PORT0 (post-sending and pre-receiving)/N:N additional delay	R/W	√		√	√	
SD132	Retry times of PORT0	R/W	√	√	√	√	
SD133	N:N network refreshing mode (PORT0)	R/W	√	√	√	√	1–13
SD134	Modbus master station error code (COM0)	R					
SD135	Local station No(PORT1)	R/W	√	√	√	√	MOD (1–32), EMR (0–31)
SD136	Max. timeout time of PORT1 (post-sending and pre-receiving)/N:N additional delay	R/W	√	√	√	√	
SD137	Retry times of PORT1	R/W	√	√	√	√	0–100
SD138	N:N network refreshing mode (PORT1)	R/W	√	√	√	√	1–13
SD139	Error code of Modbus master station (COM1)	R	√	√	√	√	

13. Extended free-port receiving control and state (PORT2)

Address	Name	Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range

Address	Name		Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD140	Free-port 2 mode state 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/W			√	√	
		SD140.3Stop bit	0: 1 stop bit 1: 2 stop bits						
		SD140.4Parity check	0: even parity 1: odd parity						
		SD140.5Parity check enabling	0: no parity check 1: parity check						
		SD140.6Character data bit	Data bit of each character 0: 8-bit character 1: 7-bit character						
		SD140.7Free-port receiving start character mode	1: has specific start character 0: no specific start character						
		SD140.8Free-port receiving end character mode	1: has specific end character 0: no specific end character						
		SD140.9Free-port intercharacter timout enabling	1: has valid intercharacter timeout 0: no valid intercharacter timeout						
		SD140. 10 Free-port interframe timeout enabling	1: has interframe timeout 0: no interframe timeout						
SD140		SD140.11	Reserved	R/W			√	√	
		SD140.12 MSB/LSB valid	0: LSB of the word element is valid 1: MSB/LSB of the word element is valid						

Address	Name		Register content	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
		SD140.13 – SD140.15	Reserved						
SD141	Start character			R/W			√	√	
SD142	End character			R/W			√	√	
SD143	Intercharacter timeout		Default: 0 ms (intercharacter timeout is omitted)	R/W			√	√	0–32767ms
SD144	Frame timeout		Default: 0 ms (frame timeout is omitted)	R/W			√	√	0–32767ms
SD145	Receiving completion message code		Bit 0: set when the receiving ends Bit 1: set when the specified end word is received Bit 2: set when the max. character number is received Bit 3: set upon intercharacter timeout Bit 4: set upon frame timout Bit 5: set upon the parity check error Bits 6-15: reserved	R			√	√	
SD146	Currently received characters			R			√	√	
SD147	Total number of currently received characters			R			√	√	
SD148	Currently transmitted characters			R			√	√	

14. Modbus (communication port 2) setup

Address	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD150	Local station No. (PORT2)	R/W			√	√	MOD (1–32)
SD151	Max. timeout time of PORT2 (post-sending and pre-receiving)/N:N additional delay	R/W			√	√	
SD152	Retry times of PORT2	R/W			√	√	0–100
SD153	N:N communication COM2 refreshing mode	R			√	√	1-18(default: 3)
SD154	N:N communication COM0 polling cycle	R			√	√	
SD155	N:N communication COM1 polling cycle	R			√	√	
SD156	N:N communication COM2 polling cycle	R			√	√	
SD159	Error code of Modbus master station (PORT2)	R			√	√	

15. Setting and reading of integrated analog channel

Address no.	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD172	Average sampling value of A/D channel 1	R	√			√	−10000–+10000
SD173	Sampling times of A/D channel 1	R/W	√			√	1–1000
SD174	Average sampling value of A/D channel 2	R	√			√	−10000–+10000
SD175	Sampling times of A/D channel 2	R/W	√			√	1–1000
SD176	Average sampling value of A/D channel 3	R	√			√	−10000–+10000
SD177	Sampling times of A/D channel 3	R/W	√			√	1–1000
SD185	Average sampling value of A/D channel 4	R	√			√	−10000–+10000
SD186	Sampling times of A/D channel 4	R/W	√			√	1–1000
SD187	Average sampling value of A/D channel 5	R	√			√	−10000–+10000
SD188	Sampling times of A/D channel 5	R/W	√			√	1–1000
SD189	Average sampling value of A/D channel 6	R	√			√	−10000–+10000
SD190	Sampling times of A/D channel 6	R/W	√			√	1–1000
SD178	Output value of D/A channel 1	R/W	√			√	−10000–+10000
Note: The values of SD173, SD175, SD177, SD186, SD188, and SD190 are 8 by default.							

16. Usage of DHSP and DHST instructions

Address	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD180	MSB of DHSP table comparison output data	R/W	√		√	√	
SD181	LSB of DHSP table comparison output data	R/W	√		√	√	
SD182	MSB of DHST or DHSP table comparison data	R/W	√	√	√	√	
SD183	LSB of DHST or DHSP table comparison data	R/W	√	√	√	√	
SD184	Record no. of the table being executed	R/W	√	√	√	√	

17. Error flag

Address	Name	R/W	IVC1	IVC2L	IVC3	IVC1L	Range
SD191	Number of the general module where bus error occurred	R	√	√	√	√	
SD192	Number of the special module where bus error occurred	R	√	√	√	√	
SD193	Detailed Modbus error (COM0)	R	√	√	√	√	
SD194	Detailed Modbus error (COM1)	R	√	√		√	
SD195	Detailed Modbus error (COM2)	R				√	

18. Enhanced positioning instruction

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD200 SD201	Used as the current value data register of Y000 output positioning instruction.	0	R/W		√	√	√
SD202 SD203	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y0)	100000 200000(IVC3)	R/W		√	√	√
SD204	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y0)	5000	R/W		√	√	√

SD205	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y0) (50ms－5000ms)	1000	R/W		√	√	√
SD206	Y0 is specified as the soft element of the clear signal		R/W		√	√	√
SD207	Crawling speed Y0 is applicable to DSZR	1000	R/W		√	√	√
SD208	The zero return speed Y0is applicable to DSZR	50000	R/W		√	√	√
SD209							
SD220	Y0 is specified as the soft element of the clear signal				√		
SD260	Deceleration time for decelerating from max. speed to base speed when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed Y0 (50 ms－5000 ms)	1000	R/W				√

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD210	Used as the current value data register of Y001 output positioning instruction.	0	R/W		√	√	√
SD211							
SD212	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y1)	100000 200000(IVC3)	R/W		√	√	√
SD213							
SD214	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y1)	5000	R/W		√	√	√
SD215	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y0) (50 ms－5000 ms)	1000	R/W		√	√	√
SD216	Y1 is specified as the soft element of the clear signal		R/W		√	√	√
SD217	Crawling speed Y1 is applicable to DSZR	1000	R/W		√	√	√
SD218	The zero return speed Y1 is applicable to DSZR	50000	R/W		√	√	√
SD219							
SD230	Y1 is specified as the soft element of the clear signal	0			√		
SD261	Deceleration time for decelerating from max. speed to base speed when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed Y0 (50 ms－5000 ms)	1000	R/W				√

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD320	Used as the current value data register of Y002 output positioning instruction.	0	R/W			√	√
SD321							
SD322	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y2)	100000 200000(IVC3)	R/W			√	√
SD323							
SD324	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y2)	5000	R/W			√	√

SD325	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y2) (50 ms－5000 ms)	1000	R/W			√	√
SD326	Y2 is specified as the soft element of the clear signal		R/W			√	√
SD327	Crawling speed Y2 is applicable to DSZR	1000	R/W			√	√
SD328 SD329	The zero return speed Y2 is applicable to DSZR	50000	R/W			√	√
SD262	Deceleration time for decelerating from max. speed to base speed when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed Y2 (50 ms－5000 ms)	1000	R/W				√

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD330 SD331	Used as the current value data register of Y003 output positioning instruction.	0	R/W			√	√
SD332 SD333	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y3)	100000 200000(IVC3)	R/W			√	√
SD334	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y3)	5000	R/W			√	√
SD335	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y3) (50 ms－5000 ms)	1000	R/W			√	√
SD336	Y3 is specified as the soft element of the clear signal		R/W			√	√
SD337	Crawling speed Y3 is applicable to DSZR	1000	R/W			√	√
SD338 SD339	The zero return speed Y3 is applicable to DSZR	50000	R/W			√	√
SD263	Deceleration time for decelerating from max. speed to base speed when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed Y3 (50 ms－5000 ms)	1000	R/W				√

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD340 SD341	Used as the current value data register of Y004 output positioning instruction.	0	R/W			√	
SD342 SD343	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y4)	100000 200000(IVC3)	R/W			√	
SD344	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y4)	5000	R/W			√	

SD345	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y4) (50 ms－5000 ms)	1000	R/W			√	
SD346	Y4 is specified as the soft element of the clear signal		R/W			√	
SD347	Crawling speed Y4 is applicable to DSZR	1000	R/W			√	
SD348	The zero return speed Y4 is applicable to DSZR	50000	R/W			√	
SD349							

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD350	Used as the current value data register of Y005 output positioning instruction.	0	R/W			√	
SD351							
SD352	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y5)	100000 200000(IVC3)	R/W			√	
SD353							
SD354	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y5)	5000	R/W			√	
SD355	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y5) (50 ms－5000 ms)	1000	R/W			√	
SD356	Y5 is specified as the soft element of the clear signal		R/W			√	
SD357	Crawling speed Y5 is applicable to DSZR	1000	R/W			√	
SD358	The zero return speed Y5 is applicable to DSZR	50000	R/W			√	
SD359							

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD360	Used as the current value data register of Y006 output positioning instruction.	0	R/W			√	
SD361							
SD362	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y6)	100000 200000(IVC3)	R/W			√	
SD363							
SD364	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y6)	5000	R/W			√	
SD365	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y6) (50 ms－5000 ms)	1000	R/W			√	
SD366	Y6 is specified as the soft element of the clear signal		R/W			√	
SD367	Crawling speed Y6 is applicable to DSZR	1000	R/W			√	
SD368	The zero return speed Y6 is applicable to DSZR	50000	R/W			√	
SD369							

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L

SD370 SD371	Used as the current value data register of Y007 output positioning instruction.	0	R/W			√	
SD372 SD373	Max. speed at which the ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (10- 100000) (y7)	100000 200000(IVC3)	R/W			√	
SD374	Base speed when ZRN, PLSV, DRVI, DRVA, DSZR and DVIT instructions are executed (1/10 of max. speed) (y7)	5000	R/W			√	
SD375	Acceleration/deceleration time for accelerating from base speed (SD204) to max. speed (SD202, SD203) when the ZRN, DRVI, DRVA, DSZR and DVIT instructions are executed (y7) (50 ms－5000 ms)	1000	R/W			√	
SD376	Y7 is specified as the soft element of the clear signal		R/W			√	
SD377	Crawling speed Y7 is applicable to DSZR	1000	R/W			√	
SD378 SD379	The zero return speed Y7 is applicable to DSZR	50000	R/W			√	

19. Signal alarm instruction

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD401	Keeping the min. No. of actions in S900-S999	0	R/W			√	

20. Timed output instruction

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD430	Scan times used by the timed clock output 1		R/W			√	
SD431	Scan times used by the timed clock output 2		R/W			√	
SD432	Scan times used by the timed clock output 3		R/W			√	
SD433	Scan times used by the timed clock output 4					√	
SD434	Scan times used by the timed clock output 5					√	

21. CAN communication

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD440	Bits 0 to 15 indicate the configuration of network nodes (stations 1 to 16). The value 1 of a bit indicates the station corresponding to the bit is configured. Bit 0 corresponds to station 1, and bit 15 corresponds to station 16.	0	R			√	
SD441	It indicates whether stations 17-32 are configured. The value 1 of a bit indicates the station corresponding to the bit is configured. Bit 0 indicates station 17 and bit 15 indicates station 32.	0	R			√	
SD442	Network baud rates 1-8 correspond to 10 kbps, 20 kbps,50 kbps, 125 kbps, 250 kbps, 500 kbps, 800 kbps, and 1 Mbps.	0	R			√	

SD443	Synchronization COB-ID	0	R			√	
SD444	Synchronization cycleperiod (1-1000ms)	0	R			√	
SD445	Start address of the mapping area (D1000 is displayed as1000)	0	R			√	
SD446							
SD447							
SD448							
SD449							
SD450	When a node is online, the value of the bit corresponding to the node is 1. For stations 1-16, bit 0 indicates station 1 and bit 15 indicates station 16.	0	R			√	
SD451	When a node is online, the value of the bit corresponding to the node is 1. For stations 17-32, bit 0 indicates station 17 and bit 15 indicates station 32.	0	R			√	
SD452	CANopen network state Bit 0 Errors of optional modules 0: No error; 1: At least one module does not match the network configuration Bit 1 Errors of mandatory modules 0: No error; 1: At least one configured module is not on the network Bit 2 Errors of mandatory modules are found in network monitoring Reserved Bit 3 Configuration process error 0: No error; 1: Error Bit 4 Network communication error0: No error; 1: Error Bit 5 One or more slave stations are not in operation 0: No error; 1: Error Bit 6 The PDO received by the CANopen master is too short 0: No error; 1: Error Bit 7–bit 10 Reserved Bit 11 Whether the master station is on the bus alone 0: No; 1: Yes Bit 15–bit 12 Reserved	0	R			√	
SD453	CANOPEN instruction error state	0	R			√	
SD454	EMCY ID	0	R			√	
SD455	EMCY DATA	0	R			√	

SD460	CANopen master station state information Bit 0 The system self-tests successfully 0: Initialization fails; 1: Succeed Bit 0 The system self-tests successfully 0: Initialization fails; 1: Succeed Bit 1 Network initialization / configuration startup 0: Failed; 1: Start Bit 2 Error occurred in slave station configuration 0: No error; 1: At least one module does not match the network configuration Bit 3 Flag bit for serious error 0: No error; 1: Serious error, restart is required Bit 4–bit 7 Error codes 0: OK; 1: Download error; 2: Initialization error Bit 8–bit 14 Master station state =0x01, Initialize =0x02, Reset node =0x04, Reset communication =0x10, Preoperate =0x20, Operate =0x30, Stop Bit 15Reserved Reserved	AN	R			√	
SD461 to SD492	State information of CANopen stations 1 to 32 Bit 0 Whether a user is configured 1: Yes; 0: No Bit 1 Whether the slave station is on the CANopen network 0: No; 1: Yes Bit 2 Whether the slave station is ready to be started 0: No; 1: Yes Bit 3 Whether the slave station is configured completely 0: No; 1: Yes Bit 4–bit 7 Error codes 0: OK bit 4=1 EMCYerror; bit 5=1 configuration error; bit 6=1 PDO is too short; bit 7=1 Life Guard or Heartbeat error =F Other errors =Others Reserved Bit8–bit15 Slave station state =0x00 Initialize =0x04 Stop =0x7F Preoperate =0x05 Operate =0xFF Unknown (the monitoring state is configured as None)	0	R			√	
SD500 to SD515	For IVC3, SD500 to SD515 are used to receive data for CANopen slave stations	0	R			√	
SD516 to SD531	For IVC3, SD516 to SD531 are used to transmit data for CANopen slave stations	0	R/W			√	

22. Ethernet communication

Address	Action and function	Initial value	R/W	IVC1	IVC2L	IVC3	IVC1L
SD560	IP0		R			√	
SD561	IP1		R			√	
SD562	IP2		R			√	
SD563	IP3		R			√	
SD564	Ethernet slave station listen port		R			√	
SD565	MAC address 0		R			√	
SD566	MAC address 1		R			√	
SD567	MAC address 2		R			√	
SD568	MAC address 3		R			√	
SD569	MAC address 4		R			√	
SD570	MAC address 5		R			√	
SD571	Number of accepted connection attempts		R			√	
SD572	Number of rejected connection attempts		R			√	
SD573	TCP error code		R			√	
SD574	Communication times		R			√	
SD575	TCP Modbus slave error code		R			√	
SD576	Bits 0 to 15 (corresponding to stations 1 to 16) indicate whether the corresponding stations are offline. The value 1 of a bit indicates the station corresponding to the bit is offline.		R			√	

Appendix C Reserved elements

Description	IVC1/IVC2L		IVC1L/IVC3/IVC1S	
Inverter instruction sending buffer zone	D7940	D7969	D7940	D7969
Inverter instruction receiving buffer zone	D7970	D7999	D7970	D7999
N:N network sharing area	D7700	D7763	D7700	D7763
N:N enhanced mode (mode 14-18) network sharing area	D7500	D7755	D7500	D7755
N:N network sharing area	M1400	M1911	M1400	M1911
EROMWR instruction operation area	D6000	D6999	—	—

Appendix D Modbus communication error codes

Abnormal code	Meaning of abnormal code
0x01	Illegal function code
0x02	Illegal register address
0x03	Data number error
0x10	Communication timeout, namely communication time exceeds the max communication time set by the user
0x11	Receiving data frame error
0x12	Parameter error, setting parameter (mode or master/slave) error
0x13	The station no. is the same as that set by the instruction, and an error occurred
0x14	Element address overflows (the data quantity received or sent is beyond the element storage space)
0x15	Instruction execution failed
0x16	The received slave address does not match the requested one, and detailed error code element stores the received slave address
0x17	The received function code does not match the requested one, and detailed error code element stores the received function code
0x18	Information frame error: the start address of element does not match, and detailed error code element stores the received element start address
0x19	The received data length does not match the one specified by the protocol or the element number exceeds the max. number specified by this function code.
0x20	CRC/LRC parity error
0x21	Reserved
0x22	Start address of instruction parameter element is set wrong
0x23	Instruction parameter sets unsupported function code or illegal function code
0x24	The number of instruction parameter elements are set wrong
0x25	Reserved
0x26	Parameter is non-modifiable during running.
0x27	Parameter is protected by a password

Appendix E Error codes of inverter instructions

Abnormal code	Meaning of abnormal code
0x1	Illegal function code
0x2	Illegal register address
0x3	Data error, namely data exceeds the upper or lower limit
0x4	Slave operation failed, including the error caused by the invalid data which is within upper/lower limit
0x5	Instruction is valid and in processing, mainly applied in storing data to nonvolatile memory
0x6	The slave station is busy, try again later, mainly applied in storing data to nonvolatile memory
0x18	Information frame error: including information length error, parity error
0x20	The parameter is non-modifiable
0x21	The parameter is non-modifiable during running.
0x22	The parameter is protected by a password

Appendix F System error codes

Error code	Meaning	Error type	Description	IVC1	IVC2L
0	No error occurred			√	√
1–9	Reserved			√	√
System hardware error					
10	SRAM error	System error	User program stops ERR indicator turns on. To remove this error, it is required to power off the PLC and check the hardware.	√	
11	FLASH error	System error	User program stops ERR indicator turns on. To remove this error, it is required to power off the PLC and check the hardware.	√	
12	Communication port error	System error	User program stops ERR indicator turns on. To remove this error, it is required to power off the PLC and check the hardware.	√	
13	Real-time clock error	System error	User program stops ERR indicator turns on. To remove this error, it is required to power off the PLC and check the hardware.	√	
14	I2C error	System error	User program stops ERR indicator turns on. To remove this error, it is required to power off the PLC and check the hardware.	√	
Peripheral error (20–23)					
20	Serious local I/O error	System error	User program stops ERR indicator turns on. To remove this error, it is required to power off the PLC and check the hardware.	√	
21	Serious extension I/O error	System error	ERR indicator blinks The alarm is cleared automatically upon the removal of the error	√	
22	Serious special module error	System error	ERR indicator blinks The alarm is cleared automatically upon the removal of the error	√	
23	Refreshing error of real-time clock (incorrect time is read during the system refreshing)	System error	ERR indicator blinks The alarm is cleared automatically upon the removal of the error	√	
24	EEPROM read/read operation error	System error	ERR indicator blinks The alarm is cleared automatically upon the removal of the error	√	
25	Local analog signal error	System error	ERR indicator blinks The alarm is cleared automatically upon the removal of the error	√	
26	System special module configuration error	System error	ERR indicator blinks The alarm is cleared automatically upon the removal of the error	√	
Storage error (40–45)					
40	User program file error	System error	User program stops, and ERR indicator turns on To remove this error, it is required to download new program or format the disk	√	√

Error code	Meaning	Error type	Description	IVC1	IVC2L
41	System configuration file error	System error	User program stops ERR indicator turns on To remove this error, it is required to download new system configuration files or format the disk	√	√
42	Data block file error	System error	User program stops, and ERR indicator turns on To remove this error, it is required to download new data block file or format the disk	√	√
43	Battery-backed data loss error	System error	User program keeps running, and ERR indicator blinks To remove this error, it is required to clear the elements, format the disk, or reset.	√	√
44	Forcing table loss error	System error	User program keeps running, and ERR indicator blinks To remove this error, it is required to clear the elements, force, format the disk, or reset.	√	√
45	User information file error	System error	User program stops, and ERR indicator turns on To remove this error, it is required to download new program and data block files or format the disk	√	√
46–59	Reserved				
Instruction execution error (60–75)					
60	User program compilation error	Execution error	User program stops, and ERR indicator turns on.	√	√
61	User program operation timeout	Execution error	User program stops, and ERR indicator turns on.	√	√
62	Illegal user program instruction execution error	Execution error	User program stops, and ERR indicator turns on.	√	√
63	Illegal element type of instruction operand	Execution error	User program stops, and ERR indicator turns on.	√	√
64	Illegal instruction operand value	Execution error	User program keeps running, and ERR indicator keeps off. The corresponding error code is prompted in SD20	√	√
65	Outside the instruction operand element range	Execution error		√	√
66	Subprogram stack overflow	Execution error		√	√
67	User interrupt request queue overflow	Execution error		√	√
68	Illegal label jump or subprogram call	Execution error		√	√
69	Divide by 0 error	Execution error		√	√
70	Definition error of the stack operated	Execution error	When the stack size, or number of stack elements are smaller than zero, or number of the stack elements exceed the limit of the stack size	√	√
71	Reserved	Execution error		√	√
72	Undefined user subprogram or interrupt subprogram	Execution error		√	√
73	Invalid special module address	Execution error			√
74	Error occurred when assess the special module	Execution error			√
75	I/O immediate refreshing error	Execution error		√	√
76	Clock setup error	Execution error		√	√

Error code	Meaning	Error type	Description	IVC1	IVC2L
77	PLSR instruction parameter error	Execution error		√	√
78	BFM buffer of assessed special module exceeds the range	Execution error			√
79	ABS data read timeout	Execution error		√	
80	ABS data read and parity error	Execution error		√	

Appendix G Modbus communication protocols (the IVC1, IVC2, and IVC3 series)

1. Overview of Modbus communication protocol

IVC series micro-PLCS are configured with two communication ports: PORT0(also as a programming port) supports the Modbus slave station, and PORT1 supports Modbus master and slave stations (configurable through the background software).

1. Using RS485 or RS232 interface, with RS232 3-line system as the physical interface.

2. Supporting RTU and ASCII modes (change of the ASCII ending character is not supported).

3. Being as an Modbus slave station with the address ranging from 1 to 31.

4. Supportingthe broadcast mode. The broadcast is effective for write and sub-function codes of diagnosis.

5. Supporting baud rates including 38,400 bps, 19,200 bps, 9,600 bps, 4,800 bps, 2,400 bps and 1,200 bps.(Default: 19200, 8 bits, 1 stop bit, and even check)

6.Supporting data field 2 ×252 bytes (ASII mode) or 252 bytes (RTU mode).

2. Supported Modbus function code and element addressing mode

The slave station supports function codes 01, 02, 03, 05, 06, 08, 15, and 16 (decimal).

- Relationship between read/write element function code and the element

Function code	Name of function code	Modicon data addresss	Type of operable element	Note
01	Read coil	0[Note 1]:xxxx	Y, X, M, SM, S, T, and C	Bit read
02	Read discrete input	1[Note 2]:xxxx	X	Bit read
03	Read register	4[Note 3]:xxxx[Note 4]	D, SD, Z, T, and C	Word read
05	Write single coil	0:xxxx	Y, M, SM, S, T, and C	Bit write
06	Write single register	4:xxxx	D, SD, Z, T, and C	Word write
15	Write multiple coils	0:xxxx	Y, M, SM, S, T, and C	Bit write
16	Write multiple registers	4:xxxx	D, SD, Z, T, and C	Word write

Notes:
1. 0 indicates "coil".
2. 1 indicates "discrete input".
3. 4 indicates "register".
4. xxxx indicates range "1–9999". Each type has an independent logic address range of 1 to 9999 (protocol address starts from 0).
5. 0, 1 and 4 do not have the physical meaning and are not involved in actual addressing.
6. You shall not write data in X element with function codes 05 and 15; otherwise, the system does not feed back the error information if the
written operand and data are correct, but the system does not perform any operation on the write instruction.

- Relationship between PLC element and Modbus communication protocol address

Element	Type	Physical element	Protocol address	Supported function code	Note
Y	Bit	Y0–Y377 (octal code) 256 points in total	0000–0255	01, 05, and 15	Output state, element code: Y0–Y7 and Y10–Y17
X	Bit	X0–X377 (octal code) 256 points in total	1200–01455 0000–0255	01, 05, and 15 02	Input state. It supports two kinds of addresses, and the element code is same as the above
M	Bit	M0–M10239	2000–4047	01, 05, and 15	

Element	Type	Physical element	Protocol address	Supported function code	Note
			12000-20191		
SM	Bit	SM0–SM511	4400–4655 30000-30255	01, 05, and 15	
S	Bit	S0–S4095	6000-7023 31000-34071	01, 05, and 15	
T	Bit	T0–T511	8000–8255 11000-11255	01, 05, and 15	State of T element
C	Bit	C0–C306	9200–9455 10000-10050	01, 05, and 15	State of C element
D	Word	D0–D7999	0000–7999	03, 06, and 16	
SD	Word	SD0–SD511	8000–8255 12000-12255	03, 06, and 16	
Z	Word	Z0–Z15	8500–8515	03, 06, and 16	
T	Word	T0–T511	9000–9255 11000-11255	03, 06, and 16	Current value of T element
C	Word	C0–C199	9500–9699	03, 06, and 16	Current value of C element (WORD)
C	Double word	C200–C255	9700–9811	03 and 16	Current value of C element (DWORD)
C	Double word	C256–C306	10000-10101	03 and 16	Current value of C element (DWORD)
R	Word	R0–R32767	13000-45767	03, 06, and 16	

Note:

The protocol address is the address used on data transmission and corresponds with the logic address of Modicon data. The protocol address starts from 0 while the logic address of Modicon data starts from 1, that is, protocol address + 1 = logic address of Modicon data. For example, if M0 protocol address is 2000, and its corresponding logic address of Modicon data is 0:2001. In fact, the read and write of M0 is completed through the protocol address, for example: read M0 element frame (sent from the master station).

01 01 07 D0 00 01 FD 47

- 01 — Station No.
- 01 — Function code
- 07 D0 — Start address. The decimal value of D7D0 is 2000.
- 00 01 — CRC check code
- FD 47 — CRC check code

- Abnormal response description:

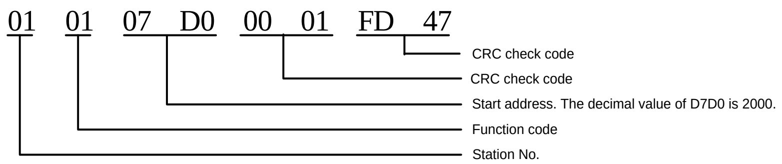
Abnormal code	Definition
0x01	Illegal function code
0x02	Illegal register address
0x03	Illegal data

Relationship between the IVC3 series PLC element and Modbus communication protocol address

Element	Type	Physical element	Protocol address	Supported function code	Note
Y	Bit	Y0–Y777 (octal code) 512 points in total	0000–0511	01, 05, and 15	Output state, element code: Y0–Y7 and Y10–Y17
X	Bit	X0–X777 (octal code) 512 points in total	1200–01711 0000–0511	01, 05, and 15 02	Input state. It supports two kinds of addresses, and the element code is same as the above
M	Bit	M0–M2047 M2048–M10240	2000–4047 12000-20191	01, 05, and 15	
SM	Bit	SM0–SM255 SM256–SM1023	4400–4655 30000-30767	01, 05, and 15	
S	Bit	S0–S1023 S1024–S4095	6000-7023 31000-34071	01, 05, and 15	
T	Bit	T0–T255 T256–T511	8000–8255 11000-11255	01, 05, and 15	State of T element
C	Bit	C0–C255 C256–C511	9200–9455 10000-10511	01, 05, and 15	State of C element
D	Word	D0–D7999	0000–7999	03, 06, and 16	

Element	Type	Physical element	Protocol address	Supported function code	Note
SD	Word	SD0–SD255 SD256–SD1023	8000–8255 12000-12767	03, 06, and 16	
Z	Word	Z0–Z15	8500–8515	03, 06, and 16	
T	Word	T0–T255 T256–T511	9000–9255 11000-11255	03, 06, and 16	Current value of T element
C	Word	C0–C199	9500–9699	03, 06, and 16	Current value of C element (WORD)
C	Double word	C200–C255	9700–9811	03 and 16	Current value of C element (DWORD)
C	Double word	C256–C306	10000-10101	03 and 16	Current value of C element (DWORD)
R	Word	R0–R32767	13000-45767	03, 06, and 16	

Note:
The protocol address is the address used on data transmission and corresponds with the logic address of Modicon data. The protocol address starts from 0 while the logic address of Modicon data starts from 1, that is, protocol address + 1 = logic address of Modicon data. For example, if M0 protocol address is 2000, and its corresponding logic address of Modicon data is 0:2001. In fact, the read and write of M0 is completed through the protocol address, for example: read M0 element frame (sent from the master station).



- Notes

1. X and Y elements use the octal system. There are 256 points in total from X0 to X377, 256 points from Y0 to Y377, with the combinations ofY0–Y7, Y10–Y17, Y20–Y27, etc.

2. Two addressing methods are available for X element. One is the protocol address of 1200-1455 with corresponding function codes of 01, 05 and 15; the other is the protocol address of 0-255 with function code 02.

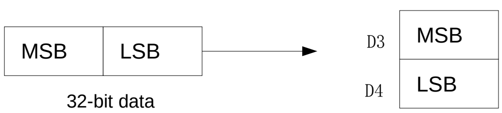
3. Processing of double-word elements: C element is a counter. It has its state and current value. C200–C255 are 32-bit elements, but each C element in the range occupies two protocol addresses during the protocol addressing. For

example: The protocol address of C200 is 9700–9701. When reading the elements though Modbus, both the startprotocol address and the number of the read elements are even number.

4. For most SM and SD elements, the real value cannot be written through Modbus, but PLC salve station still returns "OK" to indicate the completion of write operation, which is allowable.

5. In addition, the Modbus communication protocol of the IVC2Lseries supports the read and write of double word element, LONGINT variable and floating-point number. In IVC2L series PLCs, 32-bit data are stored with high bits at high address. For

example, a 32-bit data is stored in two D elements (D3 and D4), with MSB in D3 and LSB in D4, as shownin the following figure: (Refer to the description for the specific example)



3. Modbus function code description

3.1 Reading the coil state (0x01)

A maximum of 256 bit elements can be read in the IVC series PLCs.

1. Request frame

Address	Function code (01H)	Start address		Number of elements		
		H	L	H	L	

2. Response frame

If the read address is not a multiple of 8, the remaining bits are filled with 0 (starting from the high bits).

Address	Function code (01H)	Number of read elements (n) (Number of bytes)	Read data No.1	……	Read data No.n	Check code (CRC or LRC)

B7	B6	B5	B4	B3	B2	B1	B0

3.2 (0x02)Reading the discrete input state (0x02)

In the IVC series PLC, it specially refers to X element. The function code only supports the read function of X element with the max. read number of 256.

1. Request frame

Address	Function code (02H)	Start address		Number of elements		Check code (CRC or LRC)
		H	L	H	L	

2. Response frame

If the read address is not a multiple of 8, the remaining bits are filled with 0 (starting from the high bits).

Address	Function code (02H)	Number of read elements (n) (Number of bytes)	Read dataNo.1	……	Read data No.n	Check code (CRC or LRC)

B7	B6	B5	B4	B3	B2	B1	B0

3.3 Reading holding registers (0x03)

Reading holding registers refers to reading the value of data (word) register at the slave station, with a maximum of 125 registers to be read. It does not support the broadcast.

1. Request frame

Address	Function code (03H)	Start address		Number of elements		Check code (CRC or LRC)
		H	L	H	L	

2. Response frame

Address	Function code (03H)	Number of read elements (n) (Number of bytes)	Read dataNo.1		……	Read data No.n		Check code (CRC or LRC)
			H	L		H	L	

3.4 Forcing (writing) single coil (0x05)

Forcing (writing) single coil writes the bit element value to the slave station and supports broadcast, that is, writing the same element to all slave stations. It supports a maximum of 1 bit element.

1. Request frame

Address	Function code (05H)	Start address		Written element value		Check code (CRC or LRC)
		H	L	H	L	

Note: The written value of the element is 0xFF00(ON,1) or 0x0000(OFF,0).

2. Response frame

Response frame is the repetition of request frame.

Address	Function code (05H)	Start address		Written element value		Check code (CRC or LRC)
		H	L	H	L	

3.5 Presetting (writing) single register (0x06)

Presetting (writing) single register (0x06) writes the word element value to the slave station andsupports broadcast, that is, writing the same element to all slave stations. It supports a maximum of 1 word element.

1. Request frame

Address	Function code (06H)	Start address		Written element value		Check code (CRC or LRC)
		H	L	H	L	

2. Response frame

Response frame is the repetition of request frame.

Address	Function code (06H)	Start address		Written element value		Check code (CRC or LRC)
		H	L	H	L	

3.6 Returning diagnostic check (0x08)

Diagnostic registers and communication error information can be obtained through the returning diagnostic check.

Diagnostic code	Meaning
0x00	Returning the request frame
0x 01	Restarting the communication option
0x 04	Listen only mode of the slave station
0x0a	Clearing counters and diagnostic registers
0x0b	Returning the bus message count
0x0c	Returning the bus CRC error count
0x0d	Returning the slave error count
0x0e	Returning the slave message count
0x0f	Returning the slave no response count
0x12	Returning the bus character overrun count

The frame description of sub-function code is as follows.

- Returning the request frame(0x00):

  1. Request frame

Address	Function code (0x08H)	Function word code		Any character		Check code (CRC or LRC)
		(0x00H)	(0x00H)	H	L	

  2. Response frame

  Returning the request frame intact.

Address	Function code (0x08H)	Function word code		Any character		Check code (CRC or LRC)
		(0x00H)	(0x00H)	H	L	

- Restarting the communication option (0x01):

  After receiving the frame, the PLC exit from the listen only mode (Broadcast frame is supported).

  1. Request frame

  The normal data field is 0x00 00 or 0xff 00.

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		0x00H	0x01H	H	L	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		0x00H	0x01H	H	L	

- Listen only mode of the slave station(0x04):

The slave station enters the listen only mode. None of the instruction is executed or responded. The slave station can only recognize the restarting communication option instruction and enters the online mode after receiving the instruction(Broadcast frame is supported).

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x04H)	0x00H	0x00H	

2. Response frame

No return.

- Clearing counters and diagnostic registers(0x0A):

Clearing all counters (Broadcast frame is supported).

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0AH)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0AH)	0x00 H	0x00 H	

- Returning the bus message count(0x0B):

Recording the total number of the messages to all master stations from the slave stations since the last starting, clearing and power-on of counters, which excludes the message of CRC error.

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0BH)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0BH)	H	L	

- CRC error count (0x0C):

Recording the number of CRC errors received by the slave station since the last starting, clearing and power-on of counters.

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0CH)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0CH)	H	L	

- Returningthe slave error count (0x0D):

Recording the number of errors that are detected by the slave station since the last starting, clearing and power-onof counters, which includes the error detected in the broadcast message.

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0DH)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0DH)	H	L	

- Returning the slave message count(0x0E):

Recording the number of the addressing messages received by the slave station since the last starting, clearing andpower-on of counters.

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0EH)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0EH)	H	L	

- Returning the slave no response count(0x0F):

Recording the number of messages that have not returned to the slave station since the last starting, clearing andpower-on of counters.

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0FH)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x0FH)	H	L	

- Returning thebus character overrun count (0x12)

Recording the number of the messages that cannot be addressed due to the character overrun since the last starting,clearing and power-on of counters.

1. Request frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x12H)	0x00 H	0x00 H	

2. Response frame

Address	Function code (0x08H)	Function word code		Data field		Check code (CRC or LRC)
		(0x00H)	(0x12H)	H	L	

3.7 Forcing (Writing) multiple coils(0x0F)

A maximum of 1968 bit elements (0x07b0) can be written and the number is changeable according to the defined range.

1. Request frame

Address	Function	Start address	Number of elements	Number of bytes	Written element	……	Written element value No.N	Check code (CRC or LRC)

	code (0FH)	H	L	H	L	(n)	value No.1			

B7	B6	B5	B4	B3	B2	B1	B0

2. Response frame

Add ress	Function code (0FH)	Start address		Number of elements		Check code (CRC or LRC)
		H	L	H	L	

3.8 Presetting (writing) multiple registers (0x10 Hex)

A maximum of 120 registers can be written (0x78).

1. Request frame

Add ress	Function code (0x10H)	Start address		Number of elements		Number of bytes (n)	Written element value No.1		……	Written element value No.N		Check code (CRC or LRC)
		H	L	H	L		H	L		H	L	

2. Response frame

Add ress	Function code (0x10H)	Start address		Number of elements		Check code (CRC or LRC)
		H	L	H	L	

3.9 Fault response frame (0x80+function code)

Response frame:

Address	Function code	Error code (Refer to 2. "Supported Modbus function code and element addressing mode"of appendix G	Check code (CRC or LRC)

Function code refers to the function code of the captured request frame ＋0x80.

3.10 Note

1. Refer to the address classification of elements, the read soft elements each time shall be of the same type. For example, X and Y elements cannot be read in one frame.

2. The address and data range of the element shall be within the range specified by the protocol. For example:

For Y element, the protocol address range is 0000﹣0255 (Y0-Y377):

- If the read start address is 1 and 256 elements are read, address error is returned (error code 02),because there are only 255 Y elements that start from 1.
- If the read start address is 0 and 257 elements are read, data error is returned (error code 03), becausethe actual defined number of Y elements is only 256.
- If the read start address is 0 and 256 elements are read, the states of 256 elements are returned.

In other words, the number of the read elements must be within the actually defined range. It is true for read/write ofbit/word elements.

4. Example of Modbus communication control

Modbus slave station responds to the message from the master station according to the received message of local addressing, rather than sending out message actively. The slave station only supports Modbus function codes 01, 02, 03, 05, 06, 08, 15, and 16, and the other codes are "illegal function codes"(except broadcast frame).

- Reading and writing elements

All supported function codes, except 08, are used for reading and writing elements. In principle, in oneframe, there are 2000 bit elements and 125 word elements for reading, 1968 bit elements and 120 word elements for writing at most. However, theactual protocol addresses for different types of elements are different and discontinuous (for example, Y377's protocol addressis 255 while X0's protocol address is 1200). Therefore, when reading or writing an element, the element read at one timecan only be the same type, and the max. number of the read/written elements depends on the number of the elements thatare actually defined. For example, when reading Y elements(Y0 – Y377), the protocol addressranges from 0 to 255, the corresponding logic address of Modicon data is from 1 to 256, and the max. number of the Y elements that can be readis 256.The examples are as follows:

1. Sending from the master station: 0101 000001003D 9A

01 – address; 01- function code; 00 00 – start address; 01 00 – number of the read elements; 3D 9A – check

Response of the slave station: providing correct response

2. Sending from the master station: 01 01 00 00 01 01 FC 5A

The master station reads 01 01 (257) elements, which is beyond thedefined number of Y elements.

Response of the slave station: 01 81 03 00 51

The response of the slave station is illegal address, because 257>256, and 256 is the allowed max. number of Y elements.

3. Sending from the master station: 01 01 00 64 00 A0 7D AD

00 64 (decimal: 100) is the start address for the master station to read, and 00 A0 (decimal: 160) is the number of the elements.

Response of the slave station: 01 81 02 C1 91

The response of the slave station is illegal address, because the protocol address 300 has no definition of bit elements.

The response of the slave station is illegal address, because there are only 156 Y elements which are defined to start from 100, and 160 Y elements have exceeded the specified number.

4. Sending from the master station: 01 01 01 2C 00 0A 7C 38

The master station read 10 bit elements of 01 2C (decimal: 300).

Response of slave station: 01 81 02 C1 91

The response of the slave station is illegal address, because the protocol address 300 has no definition of bit elements.

5. Sending from the master station: 01 04 00 02 00 0A D1 CD

The master station sends the frame of function code 04

Response of the slave station: 01 84 01 82 C0

The response of the slave station is illegal function code.

6. Sending from the master station: 01 02 00 00 00 0A F8 0D

The master station reads 10 (X0–X9) input elements (X elements) from the start address.

Response of the slave station:01 02 02 00 00 B9 B8

The slave station responds with correct information, which has 02 bytes, and the content is 00 00.

7. Sending from the master station: 01 01 04 B0 00 0A BC DA

The master station reads 10 bit elements (X0–X9) starting from 04 B0 (demical:1200).

Response of the slave station:01 01 02 00 00 B9 FC

The slave station responds with 02 bytes, and the content is 00 00.

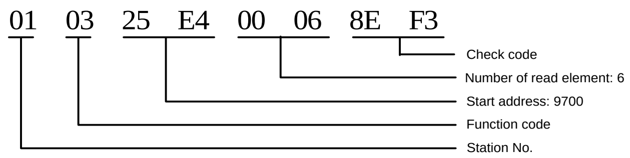
Note

1. The address of the slave station is 01, the last two bytes are CRC check codes and the second byte is function code.
2. X element does not support the write operation.

- Processing of double word elements

1. The current count value of C element is word or double word element. The values from C200 to C255 and C301 to C306 are double word elements, which are read and written through the function codes (03 and 16) of read/write registers. The address of every two registers corresponds to one C double word element, and the registers can only be read or written in pair.

For example, reading the RTU frame of three C double word elements from C200 to C202.



In the returned data, two addresses 9700 and 9701 indicate the contents of C200. 9700 is MSB, and 9701 is LSB.

2. When reading a double word element, if the start address for reading is not an even number, then the system responds with error code of illegal address. For example:

Sending from the master station: 0103 25E500 045E F2

The start address for the reading sent by the master station is 25 E5 (four word elements, decimal:9701).

Response of the slave station: 01 8302 C0 F1

Response of the slave station: Illegal data address

3. If the number of the read registers is not an even number, the system responds with error code of illegal data. For example:

Sending from the master station: 01 03 25 E4 00 05 CE F2

The start address for the master station reading is 25 E4 (5 word elements).

Response of the slave station: 01 83 03 01 31

The slave station returns the illegal data.

- Processing of LONG INT data

Based on the storage method of IVC series PLCs, one LONG INT data can be saved in two D elements. For example, if a LONG INT data is saved in D3 and D4, INVT PLC think that MSB are stored in D3 and LSB are stored in D4. When the master station reads the LONG INT data through Modbus, the 32-bit data shall be regrouped based on the LONG INTstorage principle of INVT PLC after reading the data.

The storage principle of FLOAT is the same as the storage principle of LONG INT.

5. Description of broadcast

The slave station supports broadcast but not all the function codes. The slave station supports function codes 01, 02,03, 05, 06, 08, 15 and 16 (decimal), among which, 01, 02 and 03 can read element but do not support broadcast, noresponse is gotten after sending out the broadcast; 05, 06, 15 and 16 are can write element and support broadcast,no response is gotten after sending out the broadcast, but the slave station processes the received data; 08 is thediagnostic function code, it does not support the broadcast except its sub-function codes 0x01, 0x04 and 0x0A(hex).

Appendix H TCP communication error codes

Abnormal code	Meaning
0	No error
-1	Memory allocation error
-2	Buffer error
-3	Communication timeout, namely the communication time exceeds the max. communication time set by the user
-4	Line fault
-5	In operation
-6	Illegal data
-7	Operation blockage
-8	The address is occupied
-9	Connected
-10	The memory is already in use
-11	Not connected
-12	The connection is interrupted
-13	The connection is reset
-14	The connection is already closed

Appendix I CAN communication error codes

Abnormal code	Meaning
0x0040	Unknown NMT instruction
0x0041	Heartbeat message or life protection is enabled
0x0042	Unable to send start message in this nmt state
0x0043	Heartbeat consumer configuration is wrong
0x0044	Slave node already exists
0x0050	SDO client initialization error
0x0051	One of the SDO client parameters is invalid
0x0052	SDO client does not exist
0x0053	SDO client is busy
0x0054	An error occurred during the SDO client transmission
0x0060	SDO server initialization error
0x0061	SDO server does not exist
0x0062	SDO server is busy

Appendix J ASCII code character encoding table

ASCII HEX code		Most significant 3 bits							
		0	1	2	3	4	5	6	7
Least significant 4 bits	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
	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

Appendix K Instruction order index table

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
A	ABS	Current value reading instruction	8	Zero, carry, and borrow	√	√		√	243
	ACOS	Instruction for obtaining ACOS of a floating-point number	7	Zero, carry, and borrow				√	114
	ADD	Integer addition instruction	7	Zero, carry, and borrow	√	√	√	√	98
	ANB	Energy flow block and instruction	1		√	√	√	√	68
	AND	NO contact and instruction	1		√	√	√	√	66
	AND<	Integer comparison AND< instruction	5		√	√	√	√	212
	AND<=	Integer comparison AND<= instruction	5		√	√	√	√	212
	AND<>	Integer comparison AND<> instruction	5		√	√	√	√	212
	AND=	Integer comparison AND= instruction	5		√	√	√	√	212
	AND>	Integer comparison AND> instruction	5		√	√	√	√	212
	AND>=	Integer comparison AND>= instruction	5		√	√	√	√	212
	ANDD<	Long integer comparison AND< instruction	7		√	√	√	√	215
	ANDD<=	Long integer comparison AND<= instruction	7		√	√	√	√	215
	ANDD<>	Long integer comparison AND<> instruction	7		√	√	√	√	215
	ANDD=	Long integer comparison AND= instruction	7		√	√	√	√	215
	ANDD>	Long integer comparison AND> instruction	7		√	√	√	√	215
	ANDD>=	Long integer comparison AND>= instruction	7		√	√	√	√	215
	ANDR<	Floating-point number comparison AND> instruction	7		√	√	√	√	219
	ANDR<=	Floating-point number comparison AND<= instruction	7		√	√	√	√	219
	ANDR<>	Floating-point number comparison AND<> instruction	7		√	√	√	√	219
	ANDR=	Floating-point number comparison AND= instruction	7		√	√	√	√	219
	ANDR>	Floating-point number comparison AND> instruction	7		√	√	√	√	219
	ANDR>=	Floating-point number comparison AND>= instruction	7		√	√	√	√	219
	ANI	NC contact and instruction	1		√	√	√	√	66
	ANR	Signal alarm reset instruction	1	Zero, carry, and borrow				√	262
	ANS	Signal alarm set instruction	7	Zero, carry, and borrow				√	261
	ASC	ASCII code conversion instruction	19	Zero, carry, and borrow	√	√	√	√	122
	ASIN	Instruction for obtaining ASIN of a floating-point number	7	Zero, carry, and borrow				√	113

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	ATI	Instruction for converting an ASCII code to a 16-bit hex data	7	Zero, carry, and borrow	√	√	√	√	123
	ATAN	Instruction for obtaining ATAN of a floating-point number	7	Zero, carry, and borrow				√	114
	ALT	Alternate output instruction	11	Zero, carry, and borrow	√		√	√	190
	ABSD	Cam absolute control instruction	9	Zero, carry, and borrow	√			√	187
B	BAND	Word bit contact AND instruction	5		√	√	√	√	209
	BANI	Word bit contact ANI instruction	5		√	√	√	√	209
	BCD	Instruction for converting a word to a 16-bit BCD code	5	Zero, carry, and borrow	√	√	√	√	118
	BIN	BIN: Instruction for converting a 16-bit BCD code to a word	5	Zero, carry, and borrow	√	√	√	√	119
	BITS	Instruction for counting on bit in word	5		√	√	√	√	207
	BKADD	Bulk data addition operation instruction	9	Zero, carry, and borrow				√	222
	BKCMP=,>,<,<>,<=,>=	Bulk data comparison instruction	9					√	223
	BKSUB	Bulk data subtraction operation instruction	9	Zero, carry, and borrow				√	223
	BLD	Word bit contact LD instruction	5		√	√	√	√	208
	BLDI	Word bit contact LDI instruction	5		√	√	√	√	208
	BMOV	Block data transmission instruction	7		√	√	√	√	92
	BON	Instruction for judging on bit in word	7					√	208
	BOR	Word bit contact OR instruction	5		√	√	√	√	210
	BORI	Word bit contact ORI instruction	5		√	√	√	√	210
	BOUT	Word bit coil output instruction	5		√	√	√	√	211
B	BRST	Word bit coil reset instruction	5		√	√	√	√	211
	BSET	Word bit coil set instruction	5		√	√	√	√	211
	BTOW	Byte-unit data combination instruction	7	Zero, carry, and borrow				√	258
C	CALL	Instruction for calling the user subprogram	Dependent on the parameter of the subprogram		√	√	√	√	90
	CCITT	CCITT check instruction	7		√	√	√	√	203
	CCW	Counterclockwise arc trajectory interpolation	12	Zero, carry, and borrow				√	251
	CFEND	Instruction for conditional return of user main program	1		√	√	√	√	88
	CIRET	Instruction for conditional return of user interrupt program	1		√	√	√	√	89
	CJ	Conditional jump instruction	3		√	√	√	√	88
	COS	Instruction for obtaining COS of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	111
	CRC16	CRC16 check instruction	7		√	√	√	√	204
	CSRET	Instruction for conditional return of user subprogram	1		√	√	√	√	90
	CTR	16-bit cyclic counting instruction	5		√	√	√	√	78
	CTU	16-bit increment counter instruction	5		√	√	√	√	77

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	CMP	Instruction for setting integer comparison to ON	7		√*			√	221
	CW	Clockwise arc trajectory interpolation	12	Zero, carry, and borrow				√	250
D	DADD	Long integer addition instruction	10	Zero, carry, and borrow	√	√	√	√	102
	DBAND	Deadband control instruction	9	Zero, carry, and borrow				√	225
	DBCD	Instruction for converting a double word to a 32-bit BCD code	7	Zero, carry, and borrow	√	√	√	√	119
	DBIN	Instruction for converting a 32-bit BCD code to a double word	7		√	√	√	√	119
	DBITS	Instruction for counting on bit in double word	6		√	√	√	√	207
	DCMP<	Date < comparison instruction	7		√	√		√	151
	DCMP<=	Date <= comparison instruction	7		√	√		√	151
	DCMP<>	Date <> comparison instruction	7		√	√		√	151
	DCMP=	Date = comparison instruction	7		√	√		√	151
	DCMP>	Date > comparison instruction	7		√	√		√	151
	DCMP>=	Date >= comparison instruction	7		√	√		√	151
	DCNT	32-bit increment and decrement counting instruction	7		√	√	√	√	79
	DDEC	Long integer minus one instruction	4	Zero, carry, and borrow	√	√	√	√	105
	DDIV	Long integer division instruction	10	Zero, carry, and borrow	√	√	√	√	103
	DEC	Integer minus one instruction	3	Zero, carry, and borrow	√	√	√	√	101
	DEG	Instruction for floating-point number radian-angle conversion	7	Zero, carry, and borrow				√	115
	DECO	Decoding instruction	5		√	√	√	√	206
	DFLT	Instruction for converting a long integer to a floating-point number	7	Zero, carry, and borrow	√	√		√	117
	DFMOV	Data block double word fill instruction	9		√	√	√	√	93
	DFROM	Instruction for reading double words from a special module buffer register	10		√			√	139
	DGBIN	Instruction for converting a 32-bit gray code to a double word	7	Zero, carry, and borrow	√	√	√	√	121
	DGRY	Instruction for converting a double word to a 32-bit gray code	7	Zero, carry, and borrow	√	√	√	√	120
	DHSCI	Instruction for triggering interrupt based on comparison of high-speed count	10		√	√	√	√	155
	DHSCR	Instruction for resetting the high-speed count comparison	10		√	√	√	√	157
	DHSCS	Instruction of setting the high-speed count comparison	10		√	√	√	√	154
	DHSP	Instruction for pulse output based on high-speed count table comparison	10		√	√	√	√	162
	DHSPI	Instruction for triggering interrupt based on comparison of absolute high-speed output positions	10					√	157
	DHST	High-speed count table comparison instruction	10		√	√	√	√	160
	DHSZ	High-speed count range comparison instruction	13		√	√	√	√	159

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	DI	Disable interrupt instruction	1		√	√	√	√	89
	DINC	Long integer plus one instruction	4	Zero, carry, and borrow	√	√	√	√	104
	DINT	Instruction for convert a floating-point number to a long integer	7	Zero, carry, and borrow	√	√	√	√	118
D	DIS	Instruction for separating 4 bitsof 16-bit data	7	Zero, carry, and borrow				√	260
D	DIV	Integer division instruction	7		√	√	√	√	99
D	DMOV	Double word data transmission instruction	7		√	√	√	√	91
D	DMUL	Long integer multiplication instruction	10	Zero, carry, and borrow	√	√	√	√	103
D	DNEG	Long integer negation instruction	7	Zero, carry, and borrow	√	√	√	√	105
D	DRCL	Instruction for 32-bit rotate left with carry flag bit	9	Carry	√	√	√	√	134
D	DRCR	Instruction for 32-bit rotate right with carry flag bit	9	Carry	√	√	√	√	133
D	DROL	32-bit rotate left instruction	9	Carry	√	√	√	√	133
D	DROR	32-bit rotate right instruction	9	Carry	√	√	√	√	132
D	DRVA	Absolute position control instruction	11	Zero, carry, and borrow	√*	√	√	√	241
D	DRVI	Relative position control instruction	11	Zero, carry, and borrow	√*	√	√	√	241
D	DSHL	32-bit shift left instruction	9		√	√	√	√	136
D	DSHR	32-bit shift right instruction	9		√	√	√	√	135
D	DSQT	Instruction for extracting the square root of a long integer	7	Zero, carry, and borrow	√	√	√	√	104
D	DSUB	Long integer subtraction instruction	10	Zero, carry, and borrow	√	√	√	√	102
D	DSUM	Long integer accumulation instruction	9	Zero, carry, and borrow	√	√	√	√	107
D	DTI	Instruction for converting a long integer to an integer	6	Zero, carry, and borrow	√	√	√	√	116
D	DTO	Instruction for writing double words from a special module buffer register	10		√			√	141
D	DUTY	Instruction for generating timed pulses	7					√	263
D	DVABS	Instruction for obtaining the absolute value of a long integer	7	Zero, carry, and borrow	√	√	√	√	105
D	DWAND	Double word AND instruction	10		√	√	√	√	128
D	DWINV	Double word negation instruction	10		√	√	√	√	129
D	DWOR	Double word OR instruction	10		√	√	√	√	128
D	DWXOR	Double word XOR instruction	10		√	√	√	√	129
D	DXCH	Double word swop instruction	7		√	√	√	√	94
D	DABSD	Cam absolute control instruction	11	Zero, carry, and borrow	√			√	189
D	DSZR	Instruction for zero return with DOG	9	Zero, carry, and borrow	√		√	√	244
D	DVIT	Interrupt positioning instruction	11	Zero, carry, and borrow	√		√	√	246
E	ED	Falling edge detection instruction	1		√	√	√	√	70
E	EI	Enable interrupt instruction	1		√	√	√	√	89
E	ENCO	Encoding instruction	5		√	√	√	√	206
E	EROMWR	EEPROM write instruction	7		√	√		√	143
E	EU	Rising edge detection instruction	2		√	√	√	√	70

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	IVDFWD	Inverter jogging forward instruction	6		√*	√	√	√	192
	IVDREV	Inverter jogging reverse instruction	6		√*	√	√	√	193
	IVFRQ	Inverter frequency setting instruction	8		√*	√	√	√	194
	IVFWD	Inverter forward instruction	6		√*	√	√	√	191
	IVRD	Inverter single register value reading instruction	10		√*	√	√	√	196
	IVRDST	Inverter state reading instruction	10		√*	√	√	√	195
	IVREV	Inverter reverse instruction	6		√*	√	√	√	192
	IVSTOP	Inverter stop instruction	8		√*	√	√	√	193
	IVWRT	Single register value writing instruction	10		√*	√	√	√	194
	EXP	Instruction for obtaining the natural number power of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	112
F	FIFO	First-in-first-out instruction	7		√	√	√	√	95
	FLT	Instruction for converting an integer to a floating-point number	6	Zero, carry, and borrow	√	√	√	√	116
	FMOV	Data block fill instruction	7		√	√	√	√	92
	FOR	Cycle instruction	3		√	√	√	√	86
	FROM	Instruction for reading words from a special module buffer register	9		√			√	139
G	GBIN	Instruction for converting a 16-bit gray code to a word	5	Zero, carry, and borrow	√	√	√	√	121
	GRY	Instruction for converting a word to a 16-bit gray code	5	Zero, carry, and borrow	√	√	√	√	120
H	HACKLE	Sawtooth wave signal output instruction	12		√	√	√	√	185
	HCNT	Instruction for driving the high-speed counter	7		√	√	√	√	154
	HOUR	Chronograph instruction	8		√	√	√	√	150
	HTOS	Instruction for converting hour-minute-second data to seconds	5					√	153
I	INC	Integer plus one instruction	3	Zero, carry, and borrow	√	√	√	√	100
	INITR	Instruction for initializing an extension register	5	Zero, carry, and borrow				√	235
	INT	Instruction for converting a floating-point number to an integer	6	Zero, carry, and borrow	√	√	√	√	117
	INV	Energy flow negation instruction	1		√	√	√	√	71
	ITA	Instruction for converting a 16-bit hex data to an ASCII code	7	Zero, carry, and borrow	√	√	√	√	123
	ITD	Instruction for converting an integer to a long integer	6	Zero, carry, and borrow	√	√	√	√	116
L	LBL	Jump label definition instruction	3		√	√	√	√	87
	LCNV	Project conversion instruction	9	Zero, carry, and borrow	√			√	124
	LD	NO contact instruction	1		√	√	√	√	212
	LD<	Integer comparison LD<instruction	5		√	√	√	√	212
	LD<=	Integer comparison LD<=instruction	5		√	√	√	√	212
	LD<>	Integer comparison LD<>instruction	5		√	√	√	√	212

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	LD=	Integer comparison LD=instruction	5		√	√	√	√	212
	LD>	Integer comparison LD=instruction	5		√	√	√	√	212
	LD>=	Integer comparison LD>=instruction	5		√	√	√	√	212
	LDD<	Long integer comparison LD<instruction	7		√	√	√	√	215
	LDD<=	Long integer comparison LD<=instruction	7		√	√	√	√	215
	LDD<>	Long integer comparison LD<>instruction	7		√	√	√	√	215
	LDD=	Long integer comparison LD=instruction	7		√	√	√	√	215
	LDD>	Long integer comparison LD>instruction	7		√	√	√	√	215
	LDD>=	Long integer comparison LD>=instruction	7		√	√	√	√	215
	LDI	NC contact instruction	1		√	√	√	√	65
	LDR<	Floating-point number comparison LD<instruction	7		√	√	√	√	217
	LDR<=	Floating-point number comparison LD<=instruction	7		√	√	√	√	217
	LDR<>	Floating-point number comparison LD<>instruction	7		√	√	√	√	217
	LDR=	Floating-point number comparison LD=instruction	7		√	√	√	√	217
	LDR>	Floating-point number comparison LD>instruction	7		√	√	√	√	217
	LDR>=	Floating-point number comparison LD>=instruction	7		√	√	√	√	217
	LIFO	Last-in-first-out instruction	7		√	√	√	√	96
	LIMIT	Upper/lower limit control instruction	9	Zero, carry, and borrow				√	224
	LIN	Linear trajectory interpolation instruction	12	Zero, carry, and borrow				√	248
	LN	Instruction for obtaining the natural logarithm of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	112
	LOADR	Instruction for reading data from an extension file register	5	Zero, carry, and borrow				√	233
	LOG	Instruction for obtaining the common logarithm of a floating-point number	7	Zero, carry, and borrow				√	115
	LOGR	Instruction for logging on an extension register	11	Zero, carry, and borrow				√	236
	LRC	LRC check instruction	7		√	√	√	√	205
M	MC	Main control instruction	3		√	√	√	√	72
	MCR	Main control reset instruction	1		√	√	√	√	73
	MEAN	Mean instruction	7	Zero, carry, and borrow				√	257
	Modbus	Modbus master station communication instruction	8		√	√	√	√	190
	MOV	Word data transmission instruction	5		√	√	√	√	91
	MOVLINK	Synchronous control instruction	17	Zero, carry, and borrow				√	251

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	MPP	Output energy flow stack pop instruction	1		√	√	√	√	70
	MPS	Output energy flow push instruction	1		√	√	√	√	69
	MRD	Instruction for reading output energy flow stack top value	1		√	√	√	√	69
	MUL	Integer negation instruction	8	Zero, carry, and borrow	√	√	√	√	99
	MODRW	Modbus read/write instruction	14		√*	√	√	√	198
N	NEG	Integer negation instruction	5	Zero, carry, and borrow	√	√	√	√	101
	NEXT	Cycle return instruction	1		√	√	√	√	86
	NOP	No operation instruction	1		√	√	√	√	72
O	OR	NO contact or instruction	1		√	√	√	√	67
	OR<	Integer comparison OR<instruction	5		√	√	√	√	214
	OR<=	Integer comparison OR<=instruction	5		√	√	√	√	214
	OR<>	Integer comparison OR<>instruction	5		√	√	√	√	214
	OR=	Integer comparison OR=instruction	5		√	√	√	√	214
	OR>	Integer comparison OR>instruction	5		√	√	√	√	214
	OR>=	Integer comparison OR>=instruction	5		√	√	√	√	214
	ORB	Energy flow block or instruction	1		√	√	√	√	68
	ORD<	Long integer comparison OR<instruction	7		√	√	√	√	216
	ORD<=	Long integer comparison OR<=instruction	7		√	√	√	√	216
	ORD<>	Long integer comparison OR<>instruction	7		√	√	√	√	216
	ORD=	Long integer comparison OR=instruction	7		√	√	√	√	216
	ORD>	Long integer comparison OR>instruction	7		√	√	√	√	216
	ORD>=	Long integer comparison OR>=instruction	7		√	√	√	√	216
	ORI	NO contact or instruction	1		√	√	√	√	67
	ORR<	Floating-point number comparison OR>instruction	7		√	√	√	√	220
	ORR<=	Floating-point number comparison OR<=instruction	7		√	√	√	√	220
	ORR<>	Floating-point number comparison OR<>instruction	7		√	√	√	√	220
	ORR=	Floating-point number comparison OR=instruction	7		√	√	√	√	220
	ORR>	Floating-point number comparison OR>instruction	7		√	√	√	√	220
	ORR>=	Floating-point number comparison OR>=instruction	7		√	√	√	√	220
	OUT	Coil output instruction	1		√	√	√	√	68
	OUT Sxx	SFC state jump instruction	3		√	√	√	√	74
P	PID	PID function instruction	9		√	√	√	√	180
	PLS	Envelop line pulse output instruction	7		√	√	√	√	171
	PLSR	Instruction for count pulse output with acceleration/deceleration	10		√	√	√	√	167

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	PLSV	Variable speed pulse output instruction	8	Zero, carry, and borrow	√	√	√	√	240
	PLSY	High-speed pulse output instruction	9		√	√	√	√	164
	POWER	Instruction for exponentiation of a floating-point number	10	Zero, carry, and borrow	√	√	√	√	111
	PUSH	Data push instruction	7		√	√	√	√	94
	PWM	Pulse output instruction	7		√	√	√	√	177
	PLSB	Instruction for count pulse output with base frequency and acceleration/deceleration	12	Zero, carry, and borrow		√		√	174
	RAD	Instruction for floating-point number angle-radian conversion	7	Zero, carry, and borrow				√	115
	RADD	Floating-point number addition instruction	10	Zero, carry, and borrow	√	√	√	√	107
	RAMP	Ramp signal output instruction	12		√	√	√	√	184
	RCL	Instruction for 16-bit rotate left with carry flag bit	7	Carry	√	√	√	√	132
	RCR	Instruction for 16-bit rotate right with carry flag bit	7	Carry	√	√	√	√	131
	RCV	Free-port receiving instruction	7		√	√	√	√	197
	RDIV	Floating-point number division instruction	10	Zero, carry, and borrow	√	√	√	√	109
	REF	Instruction for immediately refreshing I/O	5		√	√	√	√	142
	REFF	Instruction for setting input filtering constant	3		√	√	√	√	142
R	RET	SFC program segment end instruction	1		√	√	√	√	74
	RLCNV	Floating-point project conversion instruction	12	Zero, carry, and borrow	√			√	125
	RMOV	Floating-point data transmission instruction	7		√	√	√	√	92
	RMUL	Floating-point number multiplication instruction	10	Zero, carry, and borrow	√	√	√	√	108
	RND	Instruction for generating random numbers	3	Zero				√	262
	RNEG	Floating-point number negation instruction	7	Zero, carry, and borrow	√	√	√	√	110
	ROL	16-bit rotate left instruction	7	Carry	√	√	√	√	130
	ROR	16-bit rotate right instruction	7	Carry	√	√	√	√	130
	RSQT	Instruction for extracting the square root of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	109
	RST	Coil reset instruction	1		√	√	√	√	72
	RST Sxx	SFC state reset instruction	3		√	√	√	√	74
	RSUB	Floating-point number subtraction instruction	10	Zero, carry, and borrow	√	√	√	√	108
	RSUM	Floating-point number accumulation instruction	9	Zero, carry, and borrow	√	√	√	√	113
R	RVABS	Instruction for obtaining the absolute value of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	109
	RCMP	Instruction for setting floating-point number comparison to ON	9		√*			√	222
S	SAVER	Instruction for writing data to an extension file register	7	Zero, carry, and borrow				√	234

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	SCL	Coordinate setting instruction	7	Zero, carry, and borrow				√	226
	SEG	Instruction for converting a word to a 7-segment code	5	Zero, carry, and borrow	√	√	√	√	121
	SER	Data search instruction	9	Zero, carry, and borrow				√	227
	SET	Coil set instruction	1		√	√	√	√	72
	SET Sxx	SFC state transition instruction	3		√	√	√	√	74
	SFTL	Bit string shift left instruction	9		√	√	√	√	137
	SFTR	Bit string shift right instruction	9		√	√	√	√	136
	SHL	16-bit shift left instruction	7		√	√	√	√	135
	SHR	16-bit shift right instruction	7		√	√	√	√	134
	SIN	Instruction for obtaining SIN of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	110
	SPD	Frequency measuring instruction	7		√	√	√	√	163
	SQT	Instruction for extracting the square root of an integer	5	Zero, carry, and borrow	√	√	√	√	100
	STL	SFC state loading instruction	3		√	√	√	√	73
	STOH	Instruction for converting seconds to hour-minute-second data	5					√	153
	STOP	Instruction for stopping the user program	1		√	√	√	√	89
	STRADD	String combination instruction	7	Zero, carry, and borrow				√	228
	STRINSTR	String search instruction	9	Zero, carry, and borrow				√	232
	STRLEFT	Instruction for reading a string from left	7	Zero, carry, and borrow				√	230
	STRLEN	Instruction for detecting the string length	5	Zero, carry, and borrow				√	229
	STRMIDR	Instruction for reading any characters of a string	7	Zero, carry, and borrow				√	230
	STRMIDW	Instruction for replacing any characters of a string	7	Zero, carry, and borrow				√	231
	STRMOV	String transmission instruction	5	Zero, carry, and borrow				√	233
	STRRIGHT	Instruction for reading a string from right	7	Zero, carry, and borrow				√	229
	SUB	Integer subtraction instruction	7	Zero, carry, and borrow	√	√	√	√	99
	SUM	Integer accumulation instruction	8	Zero, carry, and borrow	√	√	√	√	106
	SWAP	MSB/LSB swop instruction	3		√	√	√	√	93
T	TADD	Clock addition instruction	7	Zero and carry	√	√	√	√	148
	TAN	Instruction for obtaining TAN of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	111
	TCMP<	Time < comparison instruction	7		√	√	√	√	152
	TCMP<=	Time >= comparison instruction	7		√	√	√	√	152
	TCMP<>	Time <> comparison instruction	7		√	√	√	√	152
	TCMP=	Time = comparison instruction	7		√	√	√	√	152
	TCMP>	Time > comparison instruction	7		√	√	√	√	152
	TCMP>=	Time >= comparison instruction	7		√	√	√	√	152
	TKY	Numeric key input instruction	7					√	144
	TMON	Non-retriggerable monostable timing instruction	5		√	√	√	√	76
	TO	Instruction for writing words from a special module buffer register	9		√			√	140

Instruction		Instructionfunction	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	TOF	Turn-off delay timing instruction	5		√	√	√	√	76
	TON	Turn-on delay timing instruction	5		√	√	√	√	75
	TONR	Memory-type turn-on delay timing instruction	5		√	√	√	√	75
	TRD	Real-time clock read instruction	3		√	√	√	√	146
	TRIANGLE	Triangle wave signal output instruction	12		√	√	√	√	186
	TSUB	Clock subtraction instruction	7	Zero and borrow	√	√	√	√	149
	TWR	Real-time clock write instruction	3		√	√	√	√	147
U	UNI	Instruction for combining 4 bits of 16-bit data	7	Zero, carry, and borrow				√	259
V	VABS	Instruction for obtaining the absolute value of an integer	5	Zero, carry, and borrow	√	√	√	√	101
	VRRD	Instruction for reading the value of an analog potentiometer	5		√	√			141
W	WAND	Word AND instruction	7		√	√	√	√	126
	WDT	Instruction for watchdog reset of user program	1		√	√	√	√	88
	WINV	Word INV instruction	5		√	√	√	√	128
	WOR	Word OR instruction	7		√	√	√	√	127
	WSFL	Word string shift left instruction	9	Zero, carry, and borrow	√	√	√	√	97
	WSFR	Word string shift right instruction	9	Zero, carry, and borrow	√	√	√	√	96
	WTOB	Byte-unit data separation instruction	7	Zero, carry, and borrow				√	257
	WXOR	Word XOR instruction	7		√	√	√	√	127
X	XCH	Word swop instruction	5		√	√	√	√	94
	XMT	Free-port sending instruction	7		√	√	√	√	196
Z	ZONE	Zone control instruction	9	Zero, carry, and borrow				√	225
	ZRN	Zero return instruction	11	Zero, carry, and borrow	√	√		√	239
	ZRST	Instruction for resetting bits to 0 in batch	5		√	√	√	√	205
	ZSET	Instruction for resetting bits in batch	5		√	√	√	√	206

Note:
*: Applicable only to IVC2L

Appendix L Instruction classification index table

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
Basic instructions	LD	NO contact instruction	1		√	√	√	√	212
	LDI	NC contact instruction	1		√	√	√	√	65
	AND	NO contact and instruction	1		√	√	√	√	66
	ANI	NC contact and instruction	1		√	√	√	√	66
	OR	NO contact or instruction	1		√	√	√	√	67
	ORI	NC contact or instruction	1		√	√	√	√	67
	OUT	Coil output instruction	1		√	√	√	√	68
	SET	Coil set instruction	1		√	√	√	√	72
	RST	Coil reset instruction	1		√	√	√	√	72
	ANB	Energy flow block and instruction	1		√	√	√	√	68
	ORB	Energy flow block or instruction	1		√	√	√	√	68
	INV	Energy flow negation instruction	1		√	√	√	√	71
	NOP	No operation instruction	1		√	√	√	√	72
	MPS	Output energy flow push instruction	1		√	√	√	√	69
	MRD	Instruction for reading output energy flow stack top value	1		√	√	√	√	69
	MPP	Output energy flow stack pop instruction	1		√	√	√	√	70
	MC	Main control instruction	3		√	√	√	√	72
	MCR	Main control reset instruction	1		√	√	√	√	73
	EU	Rising edge detection instruction	2		√	√	√	√	70
	ED	Falling edge detection instruction	2		√	√	√	√	70
	TON	Turn-on delay timing instruction	5		√	√	√	√	75
	TOF	Turn-off delay timing instruction	5		√	√	√	√	76
	TMON	Non-retriggerable monostable timing instruction	5		√	√	√	√	76
	TONR	Memory-type turn-on delay timing instruction	5		√	√	√	√	75
	CTU	16-bit increment counter instruction	5		√	√	√	√	77
	CTR	16-bit cyclic counting instruction	5		√	√	√	√	78
	DCNT	32-bit increment and decrement counting instruction	7		√	√	√	√	79
Program flow control instructions	LBL	Jump label definition instruction	3		√	√	√	√	87
	CJ	Conditional jump instruction	3		√	√	√	√	88
	CALL	Instruction for calling the user subprogram	Determined by the program		√	√	√	√	90

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	CSRET	Instruction for conditional return of user subprogram	1		√	√	√	√	90
	CFEND	Instruction for conditional return of user main program	1		√	√	√	√	88
	CIRET	Instruction for conditional return of user interrupt program	1		√	√	√	√	89
	FOR	Cycle instruction	3		√	√	√	√	86
	NEXT	Cycle return instruction	1		√	√	√	√	86
	WDT	Instruction for watchdog reset of user program	1		√	√	√	√	88
	STOP	Instruction for stopping the user program	1		√	√	√	√	89
	EI	Enable interrupt instruction	1		√	√	√	√	89
	DI	Disable interrupt instruction	1		√	√	√	√	89
SFC instructions	STL	SFC state loading instruction	3		√	√	√	√	73
	SET Sxx	SFC state transition instruction	3		√	√	√	√	74
	OUT Sxx	SFC state jump instruction	3		√	√	√	√	74
	RST Sxx	SFC state reset instruction	3		√	√	√	√	74
	RET	SFC program segment end instruction	1		√	√	√	√	74
Data transmission instructions	MOV	Word data transmission instruction	5		√	√	√	√	91
	DMOV	Double word data transmission instruction	7		√	√	√	√	91
	RMOV	Floating-point data transmission instruction	7		√	√	√		92
	BMOV	Block data transmission instruction	7		√	√	√	√	92
	SWAP	MSB/LSB swop instruction	3		√	√	√	√	93
Data flow	XCH	Word swop instruction	5		√	√	√	√	94
	DXCH	Double word swop instruction	7		√	√	√	√	94
	FMOV	Data block fill instruction	7		√	√	√	√	92
	DFMOV	Data block double word fill instruction	9		√	√	√	√	93
	WSFR	Word string shift right instruction	9	Zero, carry, and borrow	√	√	√	√	96
	WSFL	Word string shift left instruction	9	Zero, carry, and borrow	√	√	√	√	97
	PUSH	Data push instruction	7		√	√	√	√	94
	FIFO	First-in-first-out instruction	7		√	√	√	√	95
	LIFO	Last-in-first-out instruction	7		√	√	√	√	96
Integer/long integer arithmetic operation instructions	ADD	Integer addition instruction	7	Zero, carry, and borrow	√	√	√	√	98
	DADD	Long integer addition instruction	10	Zero, carry, and borrow	√	√	√	√	102
	SUB	Integer subtraction instruction	7	Zero, carry, and borrow	√	√	√	√	99
	DSUB	Long integer subtraction instruction	10	Zero, carry, and borrow	√	√	√	√	102
	INC	Integer plus one instruction	3	Zero, carry, and borrow	√	√	√	√	100
	DINC	Long integer plus one instruction	4	Zero, carry, and borrow	√	√	√	√	104

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	DEC	Integer minus one instruction	3	Zero, carry, and borrow	√	√	√	√	101
	DDEC	Long integer minus one instruction	4	Zero, carry, and borrow	√	√	√	√	105
	MUL	Integer multiplication instruction	8	Zero, carry, and borrow	√	√	√	√	99
	DMUL	Long integer multiplication instruction	10	Zero, carry, and borrow	√	√	√	√	103
	DIV	Integer division instruction	7		√	√	√	√	99
	DDIV	Long integer division instruction	10	Zero, carry, and borrow	√	√	√	√	103
	VABS	Instruction for obtaining the absolute value of an integer	5	Zero, carry, and borrow	√	√	√	√	101
	DVABS	Instruction for obtaining the absolute value of a long integer	7	Zero, carry, and borrow	√	√	√	√	105
	NEG	Integer negation instruction	5	Zero, carry, and borrow	√	√	√	√	101
	DNEG	Long integer negation instruction	7	Zero, carry, and borrow	√	√	√	√	105
	SQT	Instruction for extracting the square root of an integer	5	Zero, carry, and borrow	√	√	√	√	100
	DSQT	Instruction for extracting the square root of a long integer	7	Zero, carry, and borrow	√	√	√	√	104
	SUM	Integer accumulation instruction	8	Zero, carry, and borrow	√	√	√	√	106
	DSUM	Long integer accumulation instruction	9	Zero, carry, and borrow	√	√	√	√	107
Floating-point arithmetic operation instructions	RADD	Floating-point number addition instruction	10	Zero, carry, and borrow	√	√	√	√	107
	RSUB	Floating-point number subtraction instruction	10	Zero, carry, and borrow	√	√	√	√	108
	RMUL	Floating-point number multiplication instruction	10	Zero, carry, and borrow	√	√	√	√	108
	RDIV	Floating-point number division instruction	10	Zero, carry, and borrow	√	√	√	√	109
	RVABS	Instruction for obtaining the absolute value of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	109
	RNEG	Floating-point number negation instruction	7	Zero, carry, and borrow	√	√	√	√	110
	RSQT	Instruction for extracting the square root of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	109
	SIN	Instruction for obtaining SIN of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	110
	COS	Instruction for obtaining COS of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	111
	TAN	Instruction for obtaining TAN of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	111
	LN	Instruction for obtaining the natural logarithm of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	112

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	EXP	Instruction for obtaining the natural number power of a floating-point number	7	Zero, carry, and borrow	√	√	√	√	112
	POWER	Instruction for exponentiation of a floating-point number	10	Zero, carry, and borrow	√	√	√	√	111
Floating-point arithmetic operation instructions	RSUM	Floating-point number accumulation instruction	9	Zero, carry, and borrow	√	√	√	√	113
	ASIN	Instruction for obtaining ASIN of a floating-point number	7	Zero, carry, and borrow				√	113
	ACOS	Instruction for obtaining ACOS of a floating-point number	7	Zero, carry, and borrow				√	114
	ATAN	Instruction for obtaining ATAN of a floating-point number	7	Zero, carry, and borrow				√	114
	RAD	Instruction for floating-point number angle-radian conversion	7	Zero, carry, and borrow				√	115
	DEG	Instruction for floating-point number radian-angle conversion	7	Zero, carry, and borrow				√	115
	LOG	Instruction for obtaining the common logarithm of a floating-point number	7	Zero, carry, and borrow				√	115
Word/double word logic operation instructions	WAND	Word AND instruction	7		√	√	√	√	126
	DWAND	Double word AND instruction	10		√	√	√	√	128
	WOR	Word OR instruction	7		√	√	√	√	127
	DWOR	Double word OR instruction	10		√	√	√	√	128
	WXOR	Word XOR instruction	7		√	√	√	√	128
	DWXOR	Double word XOR instruction	10		√	√	√	√	129
	WINV	Word INV instruction	5		√	√	√	√	128
	DWINV	Double word negation instruction	7		√	√	√	√	129
Bit shift and rotate instructions	ROR	16-bit rotate right instruction	7	Carry	√	√	√	√	130
	DROR	32-bit rotate right instruction	9	Carry	√	√	√	√	132
	ROL	16-bit rotate left instruction	7	Carry	√	√	√	√	130
	DROL	32-bit rotate left instruction	9	Carry	√	√	√	√	133
	RCR	Instruction for 16-bit rotate right with carry flag bit	7	Carry	√	√	√	√	131
	DRCR	Instruction for 32-bit rotate right with carry flag bit	9	Carry	√	√	√	√	133
	RCL	Instruction for 16-bit rotate left with carry flag bit	7	Carry	√	√	√	√	132
	DRCL	Instruction for 32-bit rotate left with carry flag bit	9	Carry	√	√	√	√	134
	SHR	16-bit shift right instruction	7		√	√	√	√	134
	DSHR	32-bit shift right instruction	9		√	√	√	√	135
	SHL	16-bit shift left instruction	7		√	√	√	√	135
	DSHL	32-bit shift left instruction	9		√	√	√	√	136
	SFTL	Bit string shift left instruction	9		√	√	√	√	137
	SFTR	Bit string shift right instruction	9		√	√	√	√	136

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
Enhanced bit processing instructions	DECO	Decoding instruction	5		√	√	√	√	206
	ENCO	Encoding instruction	5		√	√	√	√	206
	BITS	Instruction for counting on bit in word	5		√	√	√	√	207
	DBITS	Instruction for counting on bit in double word	6		√	√	√	√	207
	ZRST	Instruction for resetting bits to 0 in batch	5		√	√	√	√	205
	ZSET	Instruction for resetting bits in batch	5		√	√	√	√	206
	BON	Instruction for judging on bit in word	7					√	208
High-speed I/O instructions	HCNT	High-speed counter drive instruction	7		√	√	√	√	154
	DHSCS	High-speed counting comparison set instruction	10		√	√	√	√	154
	DHSCR	High-speed counting comparison reset instruction	10		√	√	√	√	157
	DHSCI	High-speed counting comparison interrupt trigger instruction	10		√	√	√	√	155
	DHSZ	High-speed count range comparison instruction	13		√	√	√	√	159
	DHST	High-speed count table comparison instruction	10		√	√	√	√	160
	DHSP	Instruction for pulse output based on high-speed count table comparison	10		√	√	√	√	162
	SPD	Frequency measuring instruction	7		√	√	√	√	163
	PLSY	High-speed pulse output instruction	9		√	√	√	√	164
	PLSR	Instruction for count pulse output with acceleration/deceleration	10		√	√	√	√	167
	PLSB	Instruction for count pulse output with base frequency and acceleration/deceleration	12	Zero, carry, and borrow		√		√	174
	PWM	PWM pulse output instruction	7		√	√	√	√	177
	PLS	Envelop line pulse output instruction	7		√	√	√	√	177
Control calculation instructions	PID	Function instruction	9		√	√	√	√	180
	RAMP	Ramp signal output instruction	12		√	√	√	√	184
	TRIANGLE	Triangle wave signal output instruction	12		√	√	√	√	186
	HACKLE	Sawtooth wave signal output instruction	12		√	√	√	√	185
	ABSD	Cam absolute control instruction	9	Zero, carry, and borrow	√			√	187
	DABSD	Double word cam absolute control instruction	11	Zero, carry, and borrow	√			√	189
	ALT	Alternate output instruction	3	Zero, carry, and borrow	√		√	√	190

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
Peripheral instructions	FROM	Instruction for reading words from a special module buffer register	9		√			√	139
	DFROM	Instruction for reading double words from a special module buffer register	10		√			√	139
	TO	Instruction for writing words from a special module buffer register	9		√			√	140
	DTO	Instruction for writing double words from a special module buffer register	10		√			√	141
	VRRD	Instruction for reading the value of an analog potentiometer	5		√	√			141
	REFF	Instruction for setting input filtering constant	3		√	√	√	√	142
	REF	Instruction for immediately refreshing I/O	5		√	√	√	√	142
	EROMWR	EEPROM write instruction	7		√	√		√	143
	PR	Printing instruction	5		√*			√	143
	TKY	Numeric key input instruction	7					√	144
Positioning instructions	ABS	Current value reading instruction	8	Zero, carry, and borrow	√	√		√	243
	ZRN	Zero return instruction	11	Zero, carry, and borrow	√	√	√	√	239
	PLSV	Variable speed pulse output instruction	8	Zero, carry, and borrow	√	√	√	√	240
	DRVI	Relative position control instruction	11	Zero, carry, and borrow	√*	√	√	√	241
	DRVA	Absolute position control instruction	11	Zero, carry, and borrow	√*	√	√	√	241
	DSZR	Instruction for zero return with DOG	9	Zero, carry, and borrow	√		√	√	244
	DVIT	Interrupt positioning instruction	11	Zero, carry, and borrow	√		√	√	246
	LIN	Linear trajectory interpolation instruction	12	Zero, carry, and borrow				√	248
	CW	Clockwise arc trajectory interpolation	12	Zero, carry, and borrow				√	250
	CCW	Counterclockwise arc trajectory interpolation instruction	12	Zero, carry, and borrow				√	251
	MOVLINK	Synchronous control instruction	17	Zero, carry, and borrow				√	251
Real-time clock instructions	TRD	Real-time clock read instruction	3		√	√	√	√	146
	TWR	Real-time clock write instruction	3		√	√	√	√	147
	TADD	Clock addition instruction	7	Zero and carry	√	√	√	√	148
	TSUB	Clock subtraction instruction	7	Zero and borrow	√	√	√	√	149
	HOUR	Chronograph instruction	8		√	√	√	√	150
	HTOS	Instruction for converting hour-minute-second data to seconds	5					√	153

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	STOH	Instruction for converting seconds to hour-minute-second data	5					√	153
Comparison contact instructions	LD=	Integer comparison LD= instruction	5		√	√	√	√	212
	LDD=	Long integer comparison LD= instruction	7		√	√	√	√	215
	LDR=	Floating-point number comparison LD= instruction	7		√	√	√	√	217
	LD>	Integer comparison LD= instruction	5		√	√	√	√	212
	LDD>	Long integer comparison LD> instruction	7		√	√	√	√	215
	LDR>	Floating-point number comparison LD> instruction	7		√	√	√	√	217
	LD>=	Integer comparison LD>= instruction	5		√	√	√	√	212
	LDD>=	Long integer comparison LD>= instruction	7		√	√	√	√	215
Comparison contact instructions	LD<	Integer comparison LD<instruction	5		√	√	√	√	212
	LDD<	Long integer comparison LD< instruction	7		√	√	√	√	215
	LDR<	Floating-point number comparison LD< instruction	7		√	√	√	√	217
	LD<=	Integer comparison LD<= instruction	5		√	√	√	√	212
	LDD<=	Long integer comparison LD<= instruction	7		√	√	√	√	215
	LDR<=	Floating-point number comparison LD<= instruction	7		√	√	√	√	217
	LD<>	Integer comparison LD<> instruction	5		√	√	√	√	212
	LDD<>	Long integer comparison LD<> instruction	7		√	√	√	√	215
	LDR<>	Floating-point number comparison LD<> instruction	7		√	√	√	√	217
	AND=	Integer comparison AND= instruction	5		√	√	√	√	212
	ANDD=	Long integer comparison AND= instruction	7		√	√	√	√	215
	ANDR=	Floating-point number comparison AND= instruction	7		√	√	√	√	219
	AND>	Integer comparison AND> instruction	5		√	√	√	√	212
	ANDD>	Long integer comparison AND> instruction	7		√	√	√	√	215
	ANDR>	Floating-point number comparison AND> instruction	7		√	√	√	√	219
	AND>=	Integer comparison AND>= instruction	5		√	√	√	√	212
	ANDD>=	Long integer comparison AND>= instruction	7		√	√	√	√	215

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	ANDR>=	Floating-point number comparison AND>= instruction	7		√	√	√	√	219
	AND<	Integer comparison AND< instruction	5		√	√	√	√	212
	ANDD<	Long integer comparison AND< instruction	7		√	√	√	√	215
	ANDR<	Floating-point number comparison AND> instruction	7		√	√	√	√	219
	AND<=	Integer comparison AND<= instruction	5		√	√	√	√	212
	ANDD<=	Long integer comparison AND<= instruction	7		√	√	√	√	215
	ANDR<=	Floating-point number comparison AND<= instruction	7		√	√	√	√	219
Comparison contact instructions	AND<>	Integer comparison AND<> instruction	5		√	√	√	√	212
	ANDD<>	Long integer comparison AND<> instruction	7		√	√	√	√	215
	ANDR<>	Floating-point number comparison AND<> instruction	7		√	√	√	√	219
	OR=	Integer comparison OR= instruction	5		√	√	√	√	214
	ORD=	Long integer comparison OR= instruction	7		√	√	√	√	216
	ORR=	Floating-point number comparison OR= instruction	7		√	√	√	√	220
	OR>	Integer comparison OR> instruction	5		√	√	√	√	214
	ORD>	Long integer comparison OR> instruction	7		√	√	√	√	216
	ORR>	Floating-point number comparison OR> instruction	7		√	√	√	√	220
	OR>=	Integer comparison OR>= instruction	5		√	√	√	√	214
	ORD>=	Long integer comparison OR>= instruction	7		√	√	√	√	216
	ORR>=	Floating-point number comparison OR>= instruction	7		√	√	√	√	220
	OR<	Integer comparison OR< instruction	5		√	√	√	√	214
	ORD<	Long integer comparison OR< instruction	7		√	√	√	√	216
	ORR<	Floating-point number comparison OR> instruction	7		√	√	√	√	220
	OR<=	Integer comparison OR<= instruction	5		√	√	√	√	214
	ORD<=	Long integer comparison OR<= instruction	7		√	√	√	√	216
	ORR<=	Floating-point number comparison OR<= instruction	7		√	√		√	220

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	OR<>	Integer comparison OR<> instruction	5		√	√	√	√	214
	ORD<>	Long integer comparison OR<> instruction	7		√	√	√	√	216
	ORR<>	Floating-point number comparison OR<> instruction	7		√	√	√	√	220
	CMP	Instruction for setting integer comparison to ON	7		√*			√	221
	LCMP	Instruction for setting long integer comparison to ON	9		√*			√	221
	RCMP	Instruction for setting floating-point number comparison to ON	9		√*			√	222
Value conversion instructions	ITD	Instruction for converting an integer to a long integer	6	Zero, carry, and borrow	√	√	√	√	116
	DTI	Instruction for converting a long integer to an integer	6	Zero, carry, and borrow	√	√	√	√	116
	FLT	Instruction for converting an integer to a floating-point number	6	Zero, carry, and borrow	√	√	√	√	116
	DFLT	Instruction for converting a long integer to a floating-point number	7	Zero, carry, and borrow	√	√	√	√	117
	INT	Instruction for converting a floating-point number to an integer	6	Zero, carry, and borrow	√	√	√	√	117
	DINT	Instruction for convert a floating-point number to a long integer	7	Zero, carry, and borrow	√	√	√	√	118
	BCD	Instruction for converting a word to a 16-bit BCD code	5	Zero, carry, and borrow	√	√	√	√	118
	DBCD	Instruction for converting a double word to a 32-bit BCD code	7	Zero, carry, and borrow	√	√	√	√	119
	BIN	Instruction for converting a 16-bit BCD code to a word	5	Zero, carry, and borrow	√	√	√	√	119
	DBIN	Instruction for converting a 32-bit BCD code to a double word	7	Zero, carry, and borrow	√	√	√	√	119
	GRY	Instruction for converting a word to a 16-bit gray code	5	Zero, carry, and borrow	√	√	√	√	120
	DGRY	Instruction for converting a double word to a 32-bit gray code	7	Zero, carry, and borrow	√	√	√	√	120
	GBIN	Instruction for converting a 16-bit gray code to a word	5	Zero, carry, and borrow	√	√	√	√	121
	DGBIN	Instruction for converting a 32-bit gray code to a double word	7	Zero, carry, and borrow	√	√	√	√	121
	SEG	Instruction for converting a word to a 7-segment code	5	Zero, carry, and borrow	√	√	√	√	121
	ASC	ASCII code conversion instruction	19	Zero, carry, and borrow	√	√	√	√	122
	ITA	Instruction for converting a 16-bit hex data to an ASCII code	7	Zero, carry, and borrow	√	√	√	√	123

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	ATI	Instruction for converting an ASCII code to a 16-bit hex data	7	Zero, carry, and borrow	√	√	√	√	123
	LCNV	Project conversion instruction	9	Zero, carry, and borrow	√			√	124
	RLCNV	Floating-point project conversion instruction	12	Zero, carry, and borrow	√			√	125
Word contact instructions	BLD	Word bit contact LD instruction	5		√	√	√	√	208
	BLDI	Word bit contact LDI instruction	5		√	√	√	√	208
	BAND	Word bit contact AND instruction	5		√	√	√	√	209
	BANI	Word bit contact ANI instruction	5		√	√	√	√	209
	BOR	Word bit contact OR instruction	5		√	√	√	√	210
	BORI	Word bit contact ORI instruction	5		√	√	√	√	210
	BSET	Word bit coil set instruction	5		√	√	√	√	211
	BRST	Word bit coil reset instruction	5		√	√	√	√	211
	BOUT	Word bit coil output instruction	5		√	√	√	√	211
Communication instructions	Modbus	Master station communication instruction	8		√	√	√	√	190
	XMT	Free-port sending instruction	7		√	√	√	√	196
	RCV	Free-port receiving instruction	7		√	√	√	√	197
	IVFWD	Inverter forward instruction	6		√*	√	√	√	191
	IVREV	Inverter reverse instruction	6		√*	√	√	√	192
	IVDFWD	Inverter jogging forward instruction	6		√*	√	√	√	192
	IVDREV	Inverter jogging reverse instruction	6		√*	√	√	√	193
	IVSTOP	Inverter stop instruction	8		√*	√	√	√	193
	IVFRQ	Inverter frequency setting instruction	8		√*	√	√	√	194
	IVWRT	Single register value writing instruction	10		√*	√	√	√	194
	IVRDST	Inverter state reading instruction	10		√*	√	√	√	195
	IVRD	Inverter single register value reading instruction	10		√*	√	√	√	196
	MODRW	Modbus read/write instruction	14		√*	√	√	√	198
Check instructions	CCITT	CCITT check instruction	7		√	√	√	√	203
	CRC16	CRC16 check instruction	7		√	√	√	√	204
	LRC	LRC check instruction	7		√	√	√	√	205
Data comparison instructions	DCMP=	Date = comparison instruction	7		√	√	√	√	151
	DCMP>	Date > comparison instruction	7		√	√	√	√	151
	DCMP<	Date < comparison instruction	7		√	√	√	√	151
	DCMP>=	Date >= comparison instruction	7		√	√	√	√	151

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	DCMP<=	Date <= comparison instruction	7		√	√	√	√	151
	DCMP<>	Date <> comparison instruction	7		√	√	√	√	151
Time comparison instructions	TCMP=	Time = comparison instruction	7		√	√	√	√	152
	TCMP>	Time > comparison instruction	7		√	√	√	√	152
	TCMP<	Time < comparison instruction	7		√	√	√	√	152
	TCMP>=	Time >= comparison instruction	7		√	√	√	√	152
	TCMP<=	Time <= comparison instruction	7		√	√	√	√	152
	TCMP<>	Time <> comparison instruction	7		√	√	√	√	152
Data processing instructions	MEAN	Mean instruction	7	Zero, carry, and borrow				√	257
	WTOB	Byte-unit data separation instruction	7	Zero, carry, and borrow				√	257
	BTOW	Byte-unit data combination instruction	7	Zero, carry, and borrow				√	258
	UNI	Instruction for combining 4 bits of 16-bit data	7	Zero, carry, and borrow				√	259
	DIS	Instruction for separating 4 bitsof 16-bit data	7	Zero, carry, and borrow				√	260
	ANS	Signal alarm set instruction	7	Zero, carry, and borrow				√	261
	ANR	Signal alarm reset instruction	1	Zero, carry, and borrow				√	262
Bulk data processing instructions	BKADD	Bulk data addition operation instruction	9	Zero, carry, and borrow				√	222
	BKSUB	Bulk data subtraction operation instruction	9	Zero, carry, and borrow				√	223
	BKCMP=,>,<,<>,<=,>=	Bulk data comparison instruction	9	Zero, carry, and borrow				√	223
Datasheet instructions	LIMIT	Upper/lower limit control instruction	9	Zero, carry, and borrow				√	224
	DBAND	Deadband control instruction	9	Zero, carry, and borrow				√	225
	ZONE	Zone control instruction	9	Zero, carry, and borrow				√	225
	SCL	Coordinate setting instruction	7	Zero, carry, and borrow				√	226
	SER	Data search instruction	9	Zero, carry, and borrow				√	227
Character string processing instructions	STRADD	String combination instruction	7	Zero, carry, and borrow				√	228
	STRLEN	Instruction for detecting the string length	5	Zero, carry, and borrow				√	229
	STRRIGHT	Instruction for reading a string from right	7	Zero, carry, and borrow				√	229
	STRLEFT	Instruction for reading a string from left	7	Zero, carry, and borrow				√	230
	STRMIDR	Instruction for reading any characters of a string	7	Zero, carry, and borrow				√	230
	STRMIDW	Instruction for replacing any characters of a string	7	Zero, carry, and borrow				√	231

Instruction		Instruction function	Step length	Influenced flag bit	IVC2L	IVC1	IVC1L	IVC3	Page
	STRINSTR	String search instruction	9	Zero, carry, and borrow				√	232
	STRMOV	String transmission instruction	5	Zero, carry, and borrow				√	233
Extension file register instruction	LOADR	Instruction for reading data from an extension file register	5	Zero, carry, and borrow				√	233
	SAVER	Instruction for writing data to an extension file register	7	Zero, carry, and borrow				√	234
	INITR	Instruction for initializing an extension register	5	Zero, carry, and borrow				√	235
	LOGR	Instruction for logging on an extension register	11	Zero, carry, and borrow				√	236
	INITER	Instruction for initializing an extension file register	5	Zero, carry, and borrow				√	237
Other instruction	RND	Instruction for generating random numbers	3	Zero				√	262
	DUTY	Instruction for generating timed pulses	7					√	263
Note: *: Applicable only to IVC2L									