MENU

APPENDIX

1. Introduction to M, VB and VH Series PLC

1-1 PLC User Guideline

1-1-1 Structure of a PLC (Programmable Logic Controller)

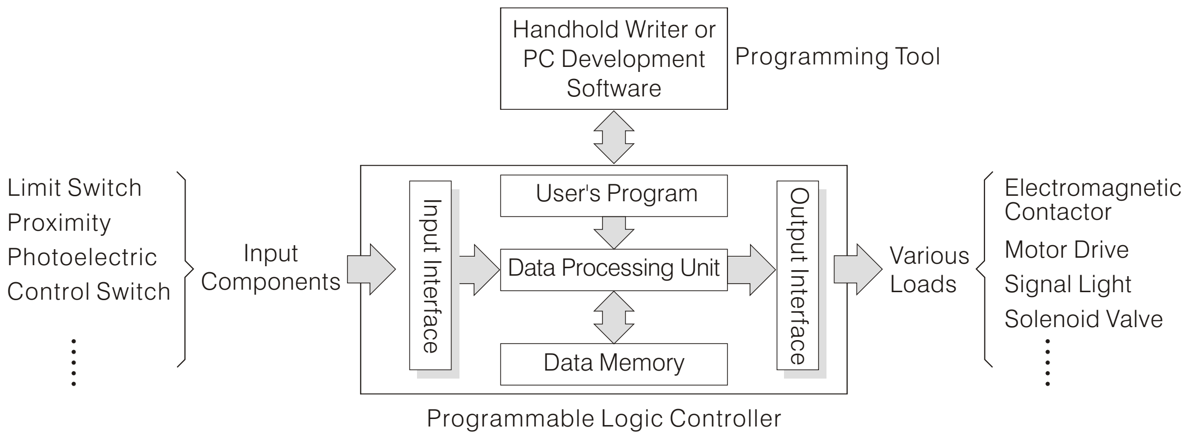


Fig. 1-2

1-1-2 Operation and Scanning Time of a PLC

The PLC operates in the Data Central Processing Unit. The operation is processed as follows:

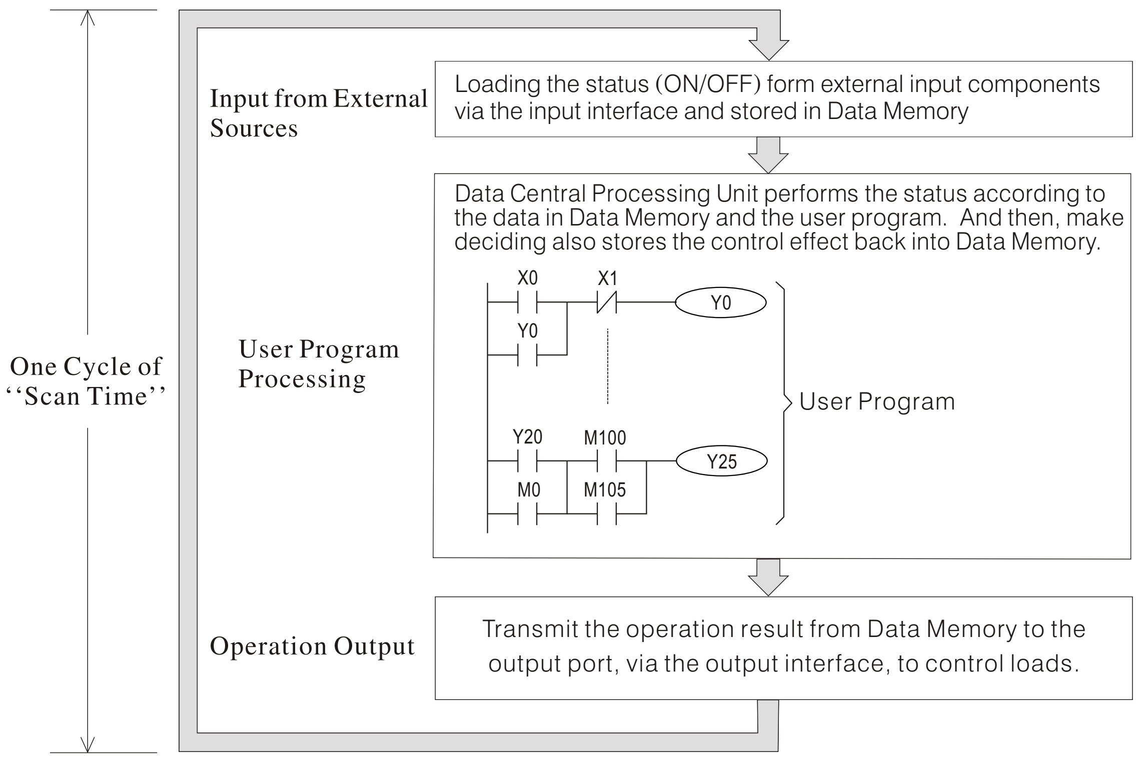


Fig. 1-2

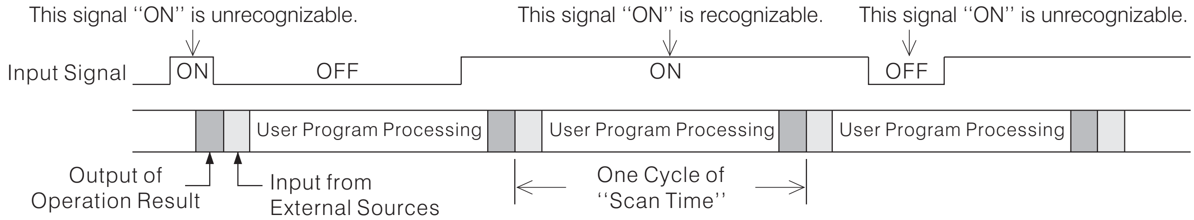
The PLC achieves to simulation the conventional relay switchboard by using the Microcomputer technology. The Microcomputer scans all I/O status and user program to calculate the control results, and the outcome will follow the user desires. The brief process is shown as in Fig. 1-2. One cycle of “Scan Time” including spend time to handle: (① loading the input status from resources, ② to process the user program, ③ the operation result output.) An important concept about the "Scan Time" is the most significant difference between a PLC and a conventional relay switchboard, and it is a critical concept that we need to understand.

1-1-3 Input / Output Delay of a PLC

- Input Delay of a PLC
  The environment of a PLC is always full of interference and noise. For protected a PLC and filtering input noise, the PLC usually uses Photo-couplers at its input ends to isolate the noise, and also adds a noise filtering circuit on the input circuit. That will cause an approximately 10 ms input signal delay. And if the input signals including some excessively narrow-short-signal-waves, that may cause those signals transmit to the PLC internal operating circuit incorrectly.

- Output Delay of a PLC
  The PLC conveyed operation result usually through relays or transistors to loadings. There will be a 10 ms mechanic delay for relays or below a 1 ms delay for transistors.

- A PLC can't capture the swiftly changing input signal
  The PLC input signal (ON or OFF) duration time must be longer than a cycle of Scan Time; Otherwise the PLC can't be recognized the correct signal properly.



1-1-4 Memory Required for a PLC

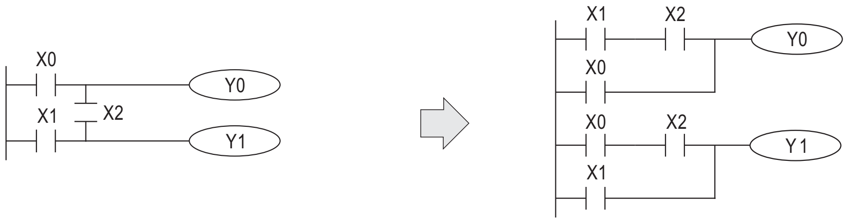
As a result of the PLC user'S program and some data must be kept, saved and updated anytime. To prevent the PLC program and data will not lost by a power failure, there are two types of data storage protect below:

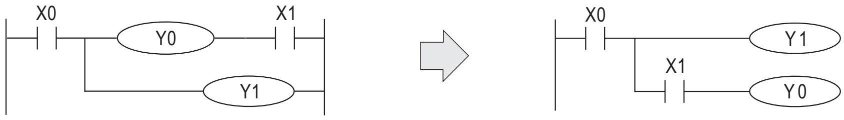
① Using SRAM (Static Random Access Memory) plus a Lithium battery as the data storage for protect the program and data in case of power failures. Due to Limited battery life span, the battery should be changed regularly. Otherwise, the program and data will disappear after the battery out. If there is short of professionally maintenance, using SRAM is not a solution.

② Using Non-Volatile Memory as the data storage for protect the program and data in case of power failures. EEPROM (Electrically Erasable Programmable Read-Only Memory) and Flash ROM are two popular types of the Non-Volatile Memory. This technique using some peculiar components to keep and preserve the data more than 10 years, which is the most stable solution for a machine is lack of care.

1-1-5 The Cases a PLC Unable To Replace a Circuit Directly

Some conventional relay switchboard circuits cannot replace by PLC Ladder Circuits directly. At the left side diagrams below are Ladder Circuits for switchboard and at the right side are alternatives for PLC.





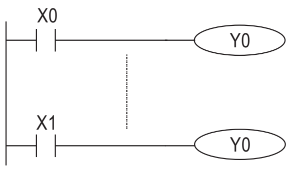
1-1-6 Double Coil Out

Please pay attention to the characteristics of PLC:

① PLC executes the program by orderly scanning (from top to bottom, from left to right).

② During the program execution, data running and changing only in the memory. The output of the operation result is only performed at the end of all execution.

The diagram shows below, the coil “Y0” has been set as OUTPUT twice, which is called “Double Coil”. In the PLC program, “X1” is the only useful (“X0” is useless) input status for control the output coil “Y0”.



Recommendations for solving Double Coil:

- Put output commands after execution and parallel all relative status
- Using SET, RST instructions
- Using CJ instruction
- Using SFC (Signal Function Chart).

MEMO

1-2 Product Profile of M Series PLC

1-2-1 Primary Features of M Series PLC

◆ Efficient Wiring, Saves Labor And Cost

- M Series PLC provides the convenient connector I/O method, which will save labor and avoid errors to reduce expense.

◆ Easy Maintenance Modular Structure

- Modular structural of M series with the wired I/O connector and conveniently dissociable base, easy for machine maintenance.

◆ Flexible modular structure, available extends to 1024 points

- Flexible I/O modular combination easily suited to even the most complicated applications. The M series is the most competitive ability product in this class.

◆ Complete System Function

- Built-in Flash ROM program memory (8K Steps), no back-up battery required.
- Main programs, component annotations and program annotations can be completely loaded to the PLC, which is a very useful tool for system maintenance.
- The password protection function can be used. It protects the copyright of the program and limited people to change the program.
- Available install a Real Time Clock unit for time dependent applications.

◆ Fully Communication Function

- When the main unit (CPU) using the RS-232 communication port, data can be transferred between the PLC and computer, HMI or SCADA, also available through a MODEM to remote control, edit program or data observe.
- Multiplex communication cards and expansion modules provide RS-232 and RS-485 interfaces.
- Support Computer Link, CPU Link, Parallel Link, Easy Link, MODBUS, MODEM and Non Protocol commutations, to satisfy diversified commutation demands.

◆ Plentiful Instructions

- The applied instructions include: program flow, compare, move, arithmetic operation, logical operation, shift, rotation, high-speed processing and handy instructions, etc.
- Extensive instruction set provides 16 Index Registers, which features more flexible program editing.

◆ Data Bank Provide Large Data Storage

◆ High-Speed Pulse Output

- The CPU unit equipped two of maximum 20 kHz high-speed pulse outputs, could drive stepping motor or servomotor.

◆ Interrupt Input and High-Speed Counter

- The CPU unit equipped 6 high-speed input points (X0 ~ X5), could be use for the interrupt inputs or high-speed counters.

◆ Flexible Modular Structure With Multitudinous Models and Modules.
Compact and Ingenious Design, Saves Assembling Space

◆ Advanced Windows® Based Software: Ladder Master

- User-friendly interface, and multi-lingual support (English, Traditional Chinese and Simple Chinese.) Function complete, easy to learn, easy to use.

◆ Advanced PDA Palm® OS Based Screen Creation Software : NeoTouch
Inaugurate a New Fashion.

1-2-2 Specifications of M Series PLC

Item			Specifications
Operation Control Method			Cyclic Operation by Stored Program
Programming Language			Electric Ladder Diagram + SFC
I/O Control Method			Batch Processing
Operation Processing Time	Basic Instruction		0.125 ~ 3.25 μs
	Applied Instruction		Several μs ~ Several 100 μs
Number of Instructions	Basic Instructions		27 (including: LDP, LDF, ANDP, ANDF, ORP, ORF, INV)
	Stepladder Instructions		2
	Applied Instructions		98
Program Capacity			8 K Steps (Flash ROM built into the unit)
Comment Capacity			2730 comments (16 characters or 8 Chinese characters for each comment)
Max. Input / Output Points			1024 points: X0 ~ X777, Y0 ~ Y777
Internal Relay	Auxiliary coil (M)	General	2000 points: M0 ~ M1999
		Latched	3120 points: M2000 ~ M5119
		Special	256 points: M9000 ~ M9255
	State coil (S)	Initial	10 points: S0 ~ S9
		General	490 points: S10 ~ S499
		Latched	400 points: S500 ~ S899
		Annunciaor	100 points: S900 ~ S999 (Latched)
Timer (T)		100 ms	200 points: T0 ~ T199 (Timer range: 0.1 ~ 3276.7 sec.)
		10 ms	46 points: T200 ~ T245 (Timer range: 0.01 ~ 327.67 sec.)
		1 ms (Retentive)	4 points: T246 ~ T249 (Timer range: 0.001 ~ 32.767 sec.)
		100 ms (Retentive)	6 points: T250 ~ T255 (Timer range: 0.1 ~ 3276.7 sec.)
Counter (C)	16-bit Up	General	100 points: C0 ~ C99
		Latched	100 points: C100 ~ C199
	32-bit Up/Down	General	20 points: C200 ~ C219
		Latched	15 points: C220 ~ C234
High Speed Counter (C)	32-bit Up/Down, Latched	1-phase Counter	11 points: C235 ~ C245
		2-phase Counter	5 points: C246 ~ C250
		A/B Phase Counter	5 points: C251 ~ C255
Data Register (D)		General	7000 points: D0 ~ D6999
		Latched	1192 points: D7000 ~ D8191
		File Register	7000 points: D1000 ~ D7999
		Special	256 points: D9000 ~ D9255
		Index	16 points: V0 ~ V7, Z0 ~ Z7
Level		Branch Level (P)	256 points: P0 ~ P255
		Interrupt Level (I)	15 points: 6 points for external interrupt, 3 points for timer interrupt, and 6 points for counter interrupt
		Nest Level (N)	8 points: N0 ~ N7
Constants	Decimal (K)	16 Bits	-32768 ~ 32767
		32 Bits	-2147483648 ~ 2147483647
	Hexadecimal (H)	16 Bits	0H ~ FFFFH
		32 Bits	0H ~ FFFFFFFFH
Pulse Output			2 points; Max. 20 kHz
Programming Device Link Interface			RS-232C
Communication Link Interface (Optional)			RS-232C or RS-422 / RS-485
Real Time Clock (Optional)			To indicates year, month, day, hour, min., sec. and week
Analog Potentiometer			2 Analog Potentiometers, each one can be seating as 0 ~ 255
Input Specifications (X0 ~ X7)	Power Source Require		DC24V ±10%, 7mA/DC24V for each point
	Input Response Time		0 ~ 60 ms, variable (Approx. 10 ms, general)
	Input Signal Type		NO/NC dry contact or NPN Open Collected Transistor
Output Specifications (Y0, Y1)	Loading Specification		DC5V ~ 30V 0.1A
	Response Time		OFF → ON: 15 μs ON → OFF: 30 μs
	Output Type		NPN Transistor Output

1-2-3 Models of M Series PLC

Item	Model No.	Specifications
CPU Module	M1-CPU1	Program capacity: 8K Steps Flash ROM Build-in; 8 points DC24V input and 2 points 0.1A transistor output
Power Module	M-PSA1	AC input power supply module. Input: Ac100 ~ 240V; Output DC24V 500mA
	M-PSD1	DC24V input power supply module
Base Unit	M-3BS	3 module units mounted base board
	M-5BS	5 module units mounted base board
	M-8BS	8 module units mounted base board
I/O Module	M-8X1	8 points DC24V input (use ATX connector, cables included)
	M-16X1	16 points DC24V input (use ATX connector, cables included)
	M-32X1	32 points DC24V input (use D-SUB connector)
	M-8YR	8 points relay output (use ATX connector, cables included)
	M-16YR	16 points relay output (use ATX connector, cables included)
	M-8YT	8 points 500mA NPN transistor output (use ATX connector, cables included)
	M-16YT	16 points 500mA NPN transistor output (use ATX connector, cables included)
	M-32YT	32 points 100mA NPN transistor output (use D-SUB connector)
	M-16XY	8 points DC24V input, 8 points relay output (use ATX connector, cables included)
Special Module	M-4AD	Analog input module, 4 points voltage or current input, 14 bits resolution
	M-2DA	Analog output module, 2 points voltage or current output, 12 bits resolution
	M-1PG	Purse output module, one axis positioning control, output pulse: 10 ~ 100Kpps
Commutation Port Expansion Card	M-RTC	RTC (Real Time Clock) expansion cord, indication of year, month, day, hour, min., sec. and week.
	M-232R	RS-232 communication expansion card (including RTC function)
	M-485R	RS-485 communication expansion card (including RTC function)
Extended Memory Card	M-MP1	8K steps Flash ROM memory card
	M-DB1	64K words Flash ROM data storage extended card
Peripheral	M-32TB1	Screw-Clamp style terminal block adapter for 32 points I/O module
	M-32TB2	Barrier style terminal block adapter for 32 points I/O module
	M-DUM	Dust cover null module
Connective Cable	VBUSB-200	200cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (USB A-type female connector)
	MWPC-200	200cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (9 pin female connector)
	MWMD-200	200cm (6.56 ft.) length connection cable from PLC Program Writer Port to a MODEM (9 pin male connector)
	MWPC25-200	200cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (25 pin female connector)
	MWBC-030	30cm (0.98 ft.) length connection cable to connect with two Base Units
	MWBC-080	80cm (2.62 ft.) length connection cable to connect with two Base Units
	MWBC-120	120cm (3.94 ft.) length connection cable to connect with two Base Units
	MWD37-050	50cm (164.04 ft.) length connection adapter cable for 32 points I/O module
	MWD37-100	100cm (328.08 ft.) length connection adapter cable for 32 points I/O module
	MWD37-200	200cm (656.16 ft.) length connection adapter cable for 32 points I/O module
	MWD37-300	300cm (984.25 ft.) length connection adapter cable for 32 points I/O module

MEMO

1-3 Product Profile of VB Series PLC
1-3-1 Primary Features of VB Series PLC

- ◆ The Innocative Multi-Functional Display, Promotion Additional Value
- ◆ Complete System Function
  - Built-in 16K (VB2, VB1) / 8K (VB0) Steps Flash ROM memory, no back-up battery required.
  - The user program, component annotations and program annotations can be completely loaded to the PLC, which is a very useful tool for system maintenance.
  - Provide password setting and prohibited upload functions, protect the copyright of PLC program.
  - Available to install a Real Time Clock unit for time dependent applications.
  - The Main Unit build-in a Multi-Function Display, display information and easy to recognize.
  - Plenty of instructions, including: floating point calculations, PID and comparison instructions, etc.
- ◆ Plenty Communication Function
  - When the Main Unit using the RS-232 communication port, data can be transferred between the PLC and a computer, HMI, or SCADA, also available through a MODEM to remote control, edit program and data observe.
  - Various RS-232 / 485 / 422 communication cards/modules, a system could expand to 19 ports.
  - The VB series PLC provides communication and link functions, ex: the Computer Link, CPU Link, Parallel Link, Easy Link, MODBUS (Master/Slave), MODEM and Non Protocol communications. The VB satisfy the most demanding communication applications.
  - Provides the MODBUS (Master/Slave) communication mode, which promotes its communication capability to other peripherals (ex. Inverters or Temperature controllers).
  - The Ethernet communication module provides to connect PLC through the network.
  - Via the Bluetooth wireless adapter, connecting to a PC could get rid of the constraint of cable.
- ◆ Numerous Models, to Satisfy Vast Control Demand
  - For a different demand the Main Unit has many kind of models 14 ~ 32 I/O points can select, satisfy diversified demand.
  - The Expansion Module has 4X/4Y ~ 16X/16Y points to choose, satisfy diversified expansion.
  - Provide several I/O connecting types (barrier terminal, IDC or ATX connector)
  - Power input provide AC85 ~ 264V or DC24V power supply.
  - Provide relay, NPN transistor or PNP transistor output.
  - The input type use Sinking/Sourcing selectable mode design.
- ◆ Diversified Special Modules, Supported Diversify Application
  - The VB series provide special modules include: analog input modules, analog output modules, temperature sensor input modules, purse output positioning controller modules, high-speed counter module, valve control module and communications module etc.
- ◆ Data Bank Provide Large Data Storage
- ◆ High-Speed Pulse Output
  - The VB1 Main Unit equipped four point high-speed pulse outputs (Y0,Y1 up to 20 KHz; Y2,Y3 up to 200KHz), could drive stepping motors or servomotors. Also, the VB1 provides some particular procession control instructions, which can procure a smooth position control easily.
  - The VB0/VB2 Main Unit equipped two of maximum 7 KHz high-speed pulse outputs, could drive stepping motors or servomotors.
- ◆ Interrupt Input and High-Speed Counter
  - The VB1 series Main Unit equipped 2 hardware high-speed counters, each one can catch a 200KHz signal (1, 2 or A/B phase).
  - The VB series Main Unit equipped 6 high-speed input points (X0 ~ X5), could be use for the interrupt inputs or high-speed counters.
- ◆ Advanced Windows® Based Software: Ladder Master
  - User-friendly interface, function complete, easy to learn, easy to use.
- ◆ The World's Forerunner of Mobile Editor : PLCmate
  - The PLCmate mobile editor could install in an intelligent cellular phone to edit the PLC program.
  - Could up/down load and edit PLC program; also the PLC system setting and monitor are available.
  - By the Bluetooth or wireless networking to connect with the PLC, escape the limit of cables.

1-3-2 Specifications of VB Series PLC

Item			VB0 Specifications	VB1 Specifications	VB2 Specifications
Operation Control Method			Cyclic Operation by Stored Program		
Programming Language			Electric Ladder Diagram + SFC		
I/O Control Method			Batch Processing		
Operation Processing Time	Basic Instruction		0.375 ~ 12.56 μs		
	Applied Instruction		Several μs ~ Several 100 μs		
Number of Instructions	Basic Instructions		27 (including: LDP,LDF, ANDP, ANDF, ORP, ORF, INV)		
	Stepladder Instructions		2		
	Applied Instructions		136	144	136
Memory Capacity (Flash ROM)	Program Capacity		8 K Steps	16 K Steps	16 K Steps
	Comment Capacity		2730 comments (16 characters for each comment)		
	Program Comment Capacity		20,000 characters		
Max. Input / Output Points			128 points	256 points	512 points
Internal Relay	Auxiliary coil (M)	General	3120 points: M0 ~ M1999, M4000 ~ M5119		
		Latched	2000 points: M2000 ~ M3999		
		Special	256 points: M9000 ~ M9255		
	State coil (S)	Initial	10 points: S0 ~ S9		
		General	490 points: S10 ~ S499		
		Latched	400 points: S500 ~ S899		
		Annunciaor	100 points: S900 ~ S999 (Latched)		
Timer (T)		100mS	200 points: T0 ~ T199 (Timer range: 0.1 ~ 3276.7 sec.)		
		10mS	46 points: T200 ~ T245 (Timer range: 0.01 ~ 327.67 sec.)		
		1 ms (Retentive)	4 points: T246 ~ T249 (Timer range: 0.001 ~ 32.767 sec.)		
		100 ms (Retentive)	6 points: T250 ~ T255 (Timer range: 0.1 ~ 3276.7 sec.)		
Counter (C)	16-bit Up	General	100 points: C0 ~ C99		
		Latched	100 points: C100 ~ C199		
	32-bit Up/Down	General	20 points: C200 ~ C219		
		Latched	15 points: C220 ~ C234		
High Speed Counter (C)	32-bit Up/Down, Latched	1-phase Counter	11 points: C235 ~ C245 (Signal Frequency: 10 KHz Max.)		
		2-phase Counter	5 points: C246 ~ C250 (Signal Frequency: 10 KHz Max.)		
		A/B Phase Counter	5 points: C251 ~ C255 (Signal Frequency: 5 KHz Max.)		
Data Register (D)		General	7680 points: D0 ~ D6999, D7512 ~ D8191		
		Latched	512 points: D7000 ~ D7511		
		File Register	7000 points: D1000 ~ D7999		
		Special	256 points: D9000 ~ D9255		
		Index	16 points: V0 ~ V7, Z0 ~ Z7		
Level		Branch Level (P)	256 points: P0 ~ P255		
		Interrupt Level (I)	15 points: 6 points for external interrupt, 3 points for timer interrupt, and 6 points for counter interrupt		
		Nest Level (N)	8 points: N0 ~ N7		
Constants	Decimal (K)	16 Bits	-32768 ~ 32767		
		32 Bits	-2147483648 ~ 2147483647		
	Hexadecimal (H)	16 Bits	0H ~ FFFFH		
		32 Bits	0H ~ FFFFFFFFH		
Hardware 32-bit High Speed Counter			–	2 points 200 KHz	–
Pulse Output			2 points, 7 KHz Max.	2 points 20 KHz; 2 points 200 KHz	2 points, 7 KHz Max.
Programming Device Link Interface CP1			RS-232C for directly connect to a PC, HMI or MODEM; with the BT-232 via Bluetooth wireless to connect to a PC or cellular phone		
Communication Link Interface CP2 (Optional)			RS-232C , RS-422 / RS-485 or Enthernet		
Real Time Clock (Optional)			To indicates year, month, day, hour, min., sec. and week		
The Number of Special Modules Limited			4	8	16
Multi-Functional Display			128 points (16×8 LED) display for I/O status and information		
Analog Potentiometers			2 Analog Potentiometers, each one can be seating as 0~255		

1-3-3 Models of VB Series PLC

<table>
<tr><th>Item</th><th>Model No.</th><th colspan="2">Specifications</th></tr>
<tr><td rowspan="6">VB0 Series Main Unit</td><td>VB0-14M★–◆</td><td colspan="2">8 points DC24V input, 6 points output, One set DC24V 420mA output, Barrier terminal I/O</td></tr>
<tr><td>VB0-20M★–◆</td><td colspan="2">12 points DC24V input, 8 points output, One set DC24V 420mA output, Barrier terminal I/O</td></tr>
<tr><td>VB0-28M★–◆</td><td colspan="2">16 points DC24V input, 12 points output, One set DC24V 420mA output, Barrier terminal I/O</td></tr>
<tr><td>VB0-32M★–◆</td><td colspan="2">16 points DC24V input, 16 points output, One set DC24V 420mA output, Barrier terminal I/O</td></tr>
<tr><td>VB0-32M★–◆C</td><td colspan="2">16 points DC24V input, 16 points output, One set DC24V 420mA output, ATX connector I/O (W/cables)</td></tr>
<tr><td>VB0-32MT-DI</td><td colspan="2">16 points DC24V input, 16 points 0.1A NPN transistor output, IDC connector I/O</td></tr>
<tr><td rowspan="4">VB1 Series Main Unit</td><td>VB1-14MT-D</td><td colspan="2">DC 24V power input, 8 points DC 24V input, 6 points NPN transistor output, Barrier terminal I/O</td></tr>
<tr><td>VB1-24MT-D</td><td colspan="2">DC 24V power input, 14 points DC 24V input, 10 points NPN transistor output, Barrier terminal I/O</td></tr>
<tr><td>VB1-32MT-D</td><td colspan="2">DC 24V power input, 16 points DC 24V input, 16 points NPN transistor output, Barrier terminal I/O</td></tr>
<tr><td>VB1-28ML-D</td><td colspan="2">DC24V power input, 12 points DC24V input, 4 points line-driver high speed counter, 8 points NPN transistor output, 4 points line-driver pulse output</td></tr>
<tr><td rowspan="4">VB2 Series Main Unit</td><td>VB2-16M★–◆</td><td colspan="2">8 points DC24V input, 8 points output, One set DC24V 420mA output,Barrier terminal I/O</td></tr>
<tr><td>VB2-32M★–◆</td><td colspan="2">16 points DC24V input, 16 points output, One set DC24V 420mA output, Barrier terminal I/O</td></tr>
<tr><td>VB2-32M★–◆C</td><td colspan="2">16 points DC24V input, 16 points output, One set DC24V 420mA output, ATX connector I/O (W/cables)</td></tr>
<tr><td>VB2-32MT-DI</td><td colspan="2">16 points DC24V input, 16 points 0.1A NPN transistor output, IDC connector I/O</td></tr>
<tr><td rowspan="2">Expansion Unit</td><td>VB-32E★–◆</td><td colspan="2">16 points DC24V input, 16 points output, One set DC24V 420mA output, Barrier terminal I/O</td></tr>
<tr><td>VB-32E★–◆C</td><td colspan="2">16 points DC24V input, 16 points output, One set DC24V 420mA output,ATX connector I/O (W/cables)</td></tr>
<tr><td rowspan="16">Expansion Module</td><td>VB-32XY★</td><td colspan="2">16 points DC 24V input, 16 points output, Barrier terminal I/O</td></tr>
<tr><td>VB-16XY★</td><td colspan="2">8 points DC24V input, 8 points output, Barrier terminal I/O</td></tr>
<tr><td>VB-16X</td><td colspan="2">16 points DC24V input, Barrier terminal input</td></tr>
<tr><td>VB-16Y★</td><td colspan="2">16 points output, Barrier terminal I/O</td></tr>
<tr><td>VB-8XY★</td><td colspan="2">4 points DC24V input, 4 points output, Barrier terminal I/O</td></tr>
<tr><td>VB-8X</td><td colspan="2">8 points DC24V input, Barrier terminal input</td></tr>
<tr><td>VB-8Y★</td><td colspan="2">8 points output, Barrier terminal Output</td></tr>
<tr><td>VB-32XY★ – C</td><td colspan="2">16 points DC 24V input, 16 points output, ATX connector I/O (with cables)</td></tr>
<tr><td>VB-16XY★ – C</td><td colspan="2">8 points DC24V input, 8 points output,ATX connector I/O (with cables)</td></tr>
<tr><td>VB-16X-C</td><td colspan="2">16 points DC24V input, ATX connector input (with cables)</td></tr>
<tr><td>VB-8X-C</td><td colspan="2">8 points DC 24V input, ATX connector input (with cables)</td></tr>
<tr><td>VB-8Y★ – C</td><td colspan="2">8 points output, ATX connector output (with cables)</td></tr>
<tr><td>VB-32XYT-I</td><td colspan="2">16 points DC24V input, 16 points 0.1A NPN transistor output, IDC connector I/O</td></tr>
<tr><td>VB-16XYT-I</td><td colspan="2">8 points DC24V input, 8 points 0.1A NPN transistor output, IDC connector I/O</td></tr>
<tr><td>VB-16X-I</td><td colspan="2">16 points DC24V input, IDC connector I/O</td></tr>
<tr><td>VB-16YT-I</td><td colspan="2">16 points 0.1A NPN transistor output, IDC connector I/O</td></tr>
<tr><td rowspan="13">Special Module</td><td>VB-4AD</td><td colspan="2">Analog input module, 4 points voltage or current input, 12 bits resolution</td></tr>
<tr><td>VB-2DA</td><td colspan="2">Analog output module, 2 points voltage or current output, 12 bits resolution</td></tr>
<tr><td>VB-4DA</td><td colspan="2">Analog output module, 4 points voltage or current output, 8 bits resolution</td></tr>
<tr><td>VB-3A</td><td colspan="2">Analog I/O module, 2 points voltage or current 12 bits resolution input, 1 points voltage or current 12 bits resolution output</td></tr>
<tr><td>VB-6A</td><td colspan="2">Analog I/O module, 4 points voltage or current 12 bits resolution input, 2 points voltage or current 12 bits resolution output</td></tr>
<tr><td>VB-4T</td><td>4 channels temperature input module</td><td rowspan="2">K/J type thermocouple inputs, 0.1℃ (0.18℉) resolution, Equipped with the cold junction compensation, open circuit detection and digital filter</td></tr>
<tr><td>VB-8T</td><td>8 channels temperature input module</td></tr>
<tr><td>VB-2PT</td><td>2 channels temperature input module</td><td rowspan="2">3-wire PT-100 3850PPM/℃, 0.1℃(0.18℉) resolution, Equipped with open circuit detection and digital filter</td></tr>
<tr><td>VB-4PT</td><td>4 channels temperature input module</td></tr>
<tr><td>VB-1LC</td><td>1 channel temperature control module</td><td rowspan="2">K/J type thermocouple or 3-wire PT-100 3850PPM/℃ inputs, 0.1℃ (0.18℉) resolution, Support CT input for observe current, Open-collector output to perform PID control, Auto Tuning and provide 14 alarm modes</td></tr>
<tr><td>VB-2LC</td><td>2 channels temperature control module</td></tr>
<tr><td>VB-1PG</td><td colspan="2">1 axis pulse output positioning control module, Output pulse frequency: 10 pps ~ 100 Kpps</td></tr>
<tr><td>VB-1HC</td><td colspan="2">1 point High-Speed Counter module, Counts pulses up to 150 KHz, 2 channels hardware compare output</td></tr>
</table>

★ -- Output type R: relay output T: NPN transistor output P: PNP transistor output

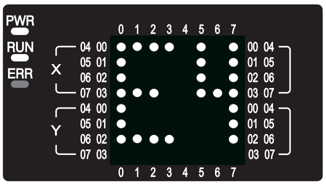
◆ -- Power type A: AC 100V ~ 240V (-15% / +10%) D: DC24V (-15% / +20%)

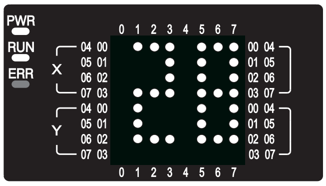
Item	Model No.	Specifications
Special Module	VB-2VC	2 channels proportional Valve Control Module; 12 bit DAC, up to 1.05A/CH, available to set the Min./Max. currents and adjust rising/falling slopes
	VB-1COM	Serial-line communication expansion module, Photo-coupler isolating, Transmission distance: 1,000 m (3280 ft.) Max.(RS-485)
	VB-PWR	Power expansion module, Input: AC 100V ~ 240V, Output: DC5V 0.4A / DC12V 0.8A / DC24V 0.5A(for sensors)
Communication Expansion Module	VB-485A	RS 485 communication expansion module, Photo-coupler isolating, Transmission distance: 1,000 m (3280 ft.) Max.
	VB-CADP	Dual communication ports expansion module, Includes an isolated RS-232/485 port and an isolated RS-485 port, Transmission distance: 1,000 m (3280 ft.) Max. (RS-485)
	VB-ENET	Ethernet communication expansion module, 10 Base T/100 Base TX by RJ-45, one isolated RS-485 port
	VB-BT232	Bluetooth communication adapter for CP1 (RS-232), distance: 100m (328 ft.) Max.
Communication Expansion Card	VB-232	RS-232 communication expansion card
	VB-485	RS-422/RS-485 communication expansion card, No isolation, Transmission distance: 50 m (162 ft.) Max.
Memory and RTC Expansion Card	VB-MP1R	16K Steps Flash ROM memory expansion card, Including RTC function
	VB-RTC	RTC (Real Time Clock) expansion card, Indicates of year, month, day, hour, min., sec. and week
	VB-DB1R	128 words data storage expansion card, Including RTC function
Connective Cable	VBUSB-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (USB A-type female connector)
	MWPC-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (9 pin female connector)
	MWMD-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a MODEM (9 pin male connector)
	MWPC25-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (25 pin female connector)
	VBMD09-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Auxiliary Port to a MODEM (9 pin male connector)
	VBPC25-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Auxiliary Port to a computer (25 pin female connector)
	VBFDHMI-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Auxiliary Port to a Fuji or Digital HMI (25 pin male connector)
	VBEC-050	50 cm (1.64 ft.) length of VB series PLC expansion cable
	VBEC-100	100 cm (3.28 ft.) length of VB series PLC expansion cable
Accessories for the IDC connector I/O model	VB-T8R	8 replaceable relays output module, 16A Max./CH, w/ surge absorbers
	VB-T8RS	8 relays output module, 5A Max./CH, w/ separable Screw-Cage Clamp terminals
	VB-T8M	8 MOSFETs output module, sourcing, 2A Max.
	VB-T16M	16 MOSFETs output module, sourcing, 2A Max.
	VB-T16TB	16 channels IDC to Screw-Cage Clamp terminal convert module
	VBIDC-050	IDC connecting cable, IDC connector at both ends, 50cm (1.64ft.) flat cable
	VBIDC-100	IDC connecting cable, IDC connector at both ends, 100cm (3.28ft.) flat cable
	VBIDC-150	IDC connecting cable, IDC connector at both ends, 150cm (4.92ft.) flat cable
	VBIDC-200	IDC connecting cable, IDC connector at both ends, 200cm (6.56ft.) flat cable
	VBIDC-250	IDC connecting cable, IDC connector at both ends, 250cm (8.2ft.) flat cable
	VBIDC-300	IDC connecting cable, IDC connector at both ends, 300cm (9.84ft.) flat cable
	VBIW-050	IDC connecting cable, IDC connector at one end, 50cm (1.64ft.) 22AWG unfasten 10 color wires
	VBIW-100	IDC connecting cable, IDC connector at one end, 100cm (3.28ft.) 22AWG unfasten 10 color wires
	VBIW-200	IDC connecting cable, IDC connector at one end, 200cm (6.56ft.) 22AWG unfasten 10 color wires
	VBIW-300	IDC connecting cable, IDC connector at one end, 300cm (9.84ft.) 22AWG unfasten 10 color wires
	VBIDC-FC100	30.48m (100ft.) 10-pin flat cable, 28AWG, for combine with IDC socket
	VBIDC-FC250	76.22m (250ft.) 10-pin flat cable, 28AWG, for combine with IDC socket
	VBIDC-HD20	20 pcs. 10-pin IDC socket w/ strain relief
	VBIDC-HD100	100 pcs. 10-pin IDC socket w/ strain relief
	VB-HT214	IDC crimping pliers tool
Power Supply	VB-30PS	30W power supply, Input: AC 100V ~ 240V, Output: DC 5V 0.2A / DC 24V 1.2A
Setting Board	DAP-100	4 keys data setting board, to collocating with Multi-Functional Display for seating arguments.

1-4 Product Profile of VH Series PLC
1-4-1 Primary Features of VH Series PLC

◆ Error Code Display Function *

- The LCD display screen on the Main Unit, which is not only for displaying the I/O status, but also hasa 109 error code (01 ~ 99 or E0 ~ E9) display function. This very useful function will promote the machine system maintenance effecting.





◆ Interrupt Input and High Speed Counter Function

- The Main Unit contains 6 rapid points (X0 ~ X5) can be used as the external interrupt input terminal and high speed counter input terminal. It can be connected up to 6 single-phase high-speed counter input signals or 2 AB-phase rotation encoders.

◆ Complete System Function

- Built-in 4K Steps Flash ROM memory, no back-up battery required.
- The user program, component annotations and program annotations can be completely loaded to the PLC, which is a very useful tool for system maintenance.
- Plenty of instructions, including (rise/fall) pulse and in-line comparison instructions made smoothly program editing.
- Provide password setting and prohibited upload functions, protect the copyright of PLC program.
- The password protection function can be used. It protects the copyright of the program and limited people to change the program.
- AC unit has a wide range switching power supply, its operational voltage is from 85 to 264V.
- Two Analog Rotary Potentiometers provide number values (0 ~ 255) which can be used for data input (i.e. changing timer settings).

◆ Flexible Modular Structure With Multitudinous Models and Modules

- The Main Unit provided 10 ~ 60 I/O points for various needs.
- The I/O expansion modules provided from 4X/4Y to 16X/16Y, fully support expansion feature needs. *
- Provide two I/O connecting types (barrier terminal or IDC connector).

◆ Plenty Communication Function

- When the Main Unit using RS-232 communication port (CP1), data can be transferred between the PLC and the computer, HMI, or SCADA, also available through a MODEM to remote control, edit program and data observe.
- Various RS-232 / RS-485 / RS-422 communication cards and modules, a system could have 3 communication ports (CP1 ~ CP3).
- The VH series PLC through the Computer Link (protocol for VH, VB and the M series) or MODBUS slave communication protocol to connect with a computer, HMI or SCADA become a local area network monitor.
- The VH series PLC has the MODBUS (Master/Slave) communication function, which can be used for connect with any MODBUS peripherals to access data.
- The Ethernet communication module provides to connect PLC through the network.
- Via the Bluetooth wireless adapter, connecting to a PC could get rid of the constraint of cable.

◆ Advanced Windows® Based Software: Ladder Master

- User-friendly interface, function complete, easy to learn, easy to use.

◆ The World's Forerunner of Mobile Editor : PLCmate

- The PLCmate mobile editor could install in an intelligent cellular phone to edit the PLC program.
- Could up/down load and edit PLC program; also the PLC system setting and monitorare available.
- By the Bluetooth or wireless networking to connect with the PLC, escape the limit of cables.

* Not available for VH-10MR , VH-14MR and VH-16MT-DI

1-4-2 Specifications of VH Series PLC

Item			Specifications
Operation Control Method			Cyclic Operation by Stored Program
Programming Language			Electric Ladder Diagram + SFC
I/O Control Method			Batch Processing
Operation Processing Time	Basic Instruction		0.375 ~ 12.56 μs
	Applied Instruction		Several μs ~ Several 100 μs
Number of Instructions	Basic Instructions		27 (including: LDP,LDF, ANDP, ANDF, ORP, ORF, INV)
	Stepladder Instructions		2
	Applied Instructions		81
Memory Capacity (Flash ROM)	Program Capacity		Built-in 4 K Steps Flash ROM
	Comment Capacity		2730 comments (16 characters for each comment)
	Program Comment Capacity		20,000 characters
Max. Input / Output Points			128 points: X0 ~ X77, Y0 ~ Y77
Internal Relay	Auxiliary Coil (M)	General	384 points: M0 ~ M383
		Latched	128 points: M384 ~ M511
		Special	256 points: M9000 ~ M9255
	State Coil (S)	Initial	10 points: S0 ~ S9 (Latched)
		Latched	118 points: S10 ~ S127
Timer (T)		100mS	63 points: T0 ~ T62 (Timer range: 0.1 ~ 3276.7 sec.)
		10mS	31 points: T32 ~ T62 (Timer range: 0.01 ~ 327.67 sec.), when the coil M9028= "ON"
		1 ms	1 points: T63 (Timer range: 0.001 ~ 32.767 sec.)
Counter (C)	16-bit Up	General	16 points: C0 ~ C15
		Latched	16 points: C16 ~ C31
High Speed Counter (C)	32-bit Up/Down, Latched	1-phase Counter	11 points: C235 ~ C245 (Signal Frequency: 10 KHz Max.)
		2-phase Counter	5 points: C246 ~ C250 (Signal Frequency: 10 KHz Max.)
		A/B Phase Counter	4 points: C251 ~ C254 (Signal Frequency: 5 KHz Max.)
Data Register (D)		General	128 points: D0 ~ D127
		Latched	128 points: D128 ~ D255
		Special	256 points: D9000 ~ D9255
		Index	16 points: V0 ~ V7, Z0 ~ Z7
Level		Branch Level (P)	64 points: P0 ~ P63
		Interrupt Level (I)	15 points: 6 points for external interrupt, 3 points for timer interrupt, and 6 points for counter interrupt
		Nest Level (N)	8 points: N0 ~ N7
Constants	Decimal (K)	16 Bits	-32768 ~ 32767
		32 Bits	-2147483648 ~ 2147483647
	Hexadecimal (H)	16 Bits	0H ~ FFFFH
		32 Bits	0H ~ FFFFFFFFH
Pulse Output			1 point; Max. 7 KHz
Programming Device Link Interface CP1			RS-232C for directly connect to a PC, HMI or MODEM; with the BT-232 via Bluetooth wireless to connect to a PC or cellular phone
Communication Link Interface CP2 (Optional)			RS-232C, RS-422/485 or Ethernet
Communication Link Interface CP3 (Optional)			RS-485, available direct connect to a computer, HMI
Real Time Clock (Optional)			To indicates year, month, day, hour, min., sec. and week
Error Code Display Function			Displays 109 error code (01~99 or E0~E9)
Analog Potentiometer			2 Analog Potentiometers, each one can be seating as 0~255

1-4-3 Models of VH Series PLC

Item	Model No.	Specifications
Main Unit	VH-10MR	6 points DC24V input, 4 points output, Power source: DC24V
	VH-14MR	8 points DC24V input, 6 points output, Power source: DC24V
	VH-20MR	12 points DC24V input, 8 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
	VH-24MR	14 points DC24V input, 10 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
	VH-28MR	16 points DC24V input, 12 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
	VH-32MR	16 points DC24V input, 16 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
	VH-40MR	24 points DC24V input, 16 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
	VH-60MR	36 points DC24V input, 24 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
	VH-20AR	8 points DC 24V digital input, 8 points relay outputs, Power source: DC 24V; 4 CH 12-bit analog inputs (+-10V / 4~20mA / +-20mA); 2 CH 12-bit analog outputs (+-10V / 4~20mA / +-20mA)
	VH-16MT-DI	8 points DC24V input, 8 points 0.1A NPN transistor output, IDC connector I/O, Power source: DC 24V
	VH-32MT-DI	16 points DC24V input, 16 points 0.1A NPN transistor output, IDC connector I/O, Power source: DC 24V
Expansion Unit	VH-32ER	16 points DC24V input, 16 points output, Power source: AC100 ~ 240V, One set DC24V 420mA output
Expansion Module	VH-28XYR	20 points DC24V input, 8 points relay output
	VH-16XYR	8 points DC24V input, 8 points relay output
	VH-16X	16 points DC24V input
	VH-8XYR	4 points DC24V input, 4 points relay output
	VH-8X	8 points DC24V input
	VH-8YR	8 points relay output
	VH-16XYT-I	8 points DC24V input, 8 points 0.1A NPN transistor output, IDC connector I/O
Communication Expansion Module	VB-485A	RS 485 communication expansion module, Photocoupler isolating, Transmission distance: 1,000 m (3280 ft.) Max.
	VB-CADP	Dual communication ports expansion module, Includes an isolated RS-232/485 port and an isolated RS-485 port, Transmission distance: 1,000 m (3280 ft.) Max. (RS-485)
	VB-ENET	Ethernet communication expansion module, 10 Base T/100 Base TX by RJ-45, one isolated RS-485 port
	VB-BT232	Bluetooth communication adapter for Cp1 (RS-232) , distance: 100m (328 ft.) Max.
Communication Expansion Card	VB-232	RS-232 communication expansion card
	VB-485	RS-422/RS-485 communication expansion card, No isolation, Transmission distance: 50 m (162 ft.) Max.
Expansion Card	VB-MP1R	Flash ROM memory cartridge (Only 4 K steps programs stored for VH Series), Including RTC function
	VB-RTC	RTC (Real Time Clock) expansion card, Indicates of year, month, day, hour, min., sec. and week
Connective Cable	VBUSB-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (USB A-type female connector)
	MWPC-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (9 pin female connector)
	MWMD-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a MODEM (9 pin male connector)
	MWPC25-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Port to a computer (25 pin female connector)
	VBMD09-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Auxiliary Port to a MODEM (9 pin male connector)
	VBPC25-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Auxiliary Port to a computer (25 pin female connector)
	VBFDHMI-200	200 cm (6.56 ft.) length connection cable from PLC Program Writer Auxiliary Port to a Fuji or Digital HMI (25 pin male connector)
	VHEC-050	50 cm (6.56 ft.) length of VH series PLC expansion cable
Power Supply	VB-30PS	30W power supply, Input: AC 100V ~ 240V, Output: DC 5V 0.2A / DC 24V 1.2A

Item	Model No.	Specifications
Accessories for the IDC connector I/O model	VB-T8R	8 replaceable relays output module, 16A Max./CH, w/ surge absorbers
	VB-T8RS	8 relays output module, 5A Max./CH, w/ separable Screw-Cage Clamp terminals
	VB-T8M	8 MOSFETs output module, sourcing, 2A Max.
	VB-T16M	16 MOSFETs output module, sourcing, 2A Max.
	VB-T16TB	16 channels IDC to Screw-Cage Clamp terminal convert module
	VBIDC-050	IDC connecting cable, IDC connector at both ends, 50cm (1.64ft.) flat cable
	VBIDC-100	IDC connecting cable, IDC connector at both ends, 100cm (3.28ft.) flat cable
	VBIDC-150	IDC connecting cable, IDC connector at both ends, 150cm (4.92ft.) flat cable
	VBIDC-200	IDC connecting cable, IDC connector at both ends, 200cm (6.56ft.) flat cable
	VBIDC-250	IDC connecting cable, IDC connector at both ends, 250cm (8.2ft.) flat cable
	VBIDC-300	IDC connecting cable, IDC connector at both ends, 300cm (9.84ft.) flat cable
	VBIW-050	IDC connecting cable, IDC connector at one end, 50cm (1.64ft.) 22AWG unfasten 10 color wires
	VBIW-100	IDC connecting cable, IDC connector at one end, 100cm (3.28ft.) 22AWG unfasten 10 color wires
	VBIW-200	IDC connecting cable, IDC connector at one end, 200cm (6.56ft.) 22AWG unfasten 10 color wires
	VBIW-300	IDC connecting cable, IDC connector at one end, 300cm (9.84ft.) 22AWG unfasten 10 color wires
	VBIDC-FC100	30.48m (100ft.) 10-pin flat cable, 28AWG, for combine with IDC socket
	VBIDC-FC250	76.22m (250ft.) 10-pin flat cable, 28AWG, for combine with IDC socket
	VBIDC-HD20	20 pcs. 10-pin IDC socket w/ strain relief
	VBIDC-HD100	100 pcs. 10-pin IDC socket w/ strain relief
	VB-HT214	IDC crimping pliers tool

1-5 Instruction Table of M, VB, VH Series PLC

1-5-1 Basic Instruction Table

Instruction Title	Function	Devices	Applicable PLC Type			Ref. Page
			M	VB	VH	
LD	Initial logical operation contact type NO (normally open)	X, Y, M, S, T, C	○	○	○	75
LDI	Initial logical operation contact type NC (normally closed)	X, Y, M, S, T, C	○	○	○	75
AND	Serial link of NO (normally open) contacts	X, Y, M, S, T, C	○	○	○	75
ANI	Serial link of NC (normally closed) contacts	X, Y, M, S, T, C	○	○	○	75
OR	Parallel link of NO (normally open) contacts	X, Y, M, S, T, C	○	○	○	75
ORI	Parallel link of NC (normally closed) contacts	X, Y, M, S, T, C	○	○	○	75
ANB	Serial link of multiple parallel circuits	—	○	○	○	76
ORB	Parallel link of multiple contact circuits	—	○	○	○	76
OUT	Final logical operation type coil drive	Y, M, S, T, C	○	○	○	75
SET	Sets component permanently ON	Y, M, S	○	○	○	79
RST	Resets component permanently OFF	Y, M, S, T, C, D	○	○	○	79
PLS	Rising edge pulse	Y, M	○	○	○	79
PLF	Falling/trailing edge pulse	Y, M	○	○	○	79
MC	Denotes the start of a master control block	N0 ~ N7	○	○	○	78
MCR	Denotes the end of a master control block	N0 ~ N7	○	○	○	78
MPS	Stores the current result of the internal PLC operations	—	○	○	○	77
MRD	Reads the current result of the internal PLC operations	—	○	○	○	77
MPP	Pops (recalls and removes) the currently stored result	—	○	○	○	77
NOP	No operation or null step	—	○	○	○	—
END	Force the current program scan to end	—	○	○	○	75
LDP	Initial logical operation Rising edge pulse	X, Y, M, S, T, C	○	○	○	81
LDF	Initial logical operation Falling/trailing edge pulse	X, Y, M, S, T, C	○	○	○	81
ANDP	Serial link of Rising edge pulse	X, Y, M, S, T, C	○	○	○	81
ANDF	Serial link of Falling/trailing edge pulse	X, Y, M, S, T, C	○	○	○	81
ORP	Parallel link of Rising edge pulse	X, Y, M, S, T, C	○	○	○	81
ORF	Parallel link of Falling/trailing edge pulse	X, Y, M, S, T, C	○	○	○	81
INV	Invert the current result of the internal PLC operations	—	○	○	○	81

Step Ladder Instruction Table

Instruction Title	Function	Devices	Applicable PLC Type			Ref. Page
			M	VB	VH	
STL	Initiation of Stepladder	S	○	○	○	87
REL	End of Stepladder	—	○	○	○	87

1-5-2 Applied Instruction Table

Type	FNC No.	Instruction Title (D)	Instruction Title	Instruction Title (P)	Function	Applicable PLC Type M	Applicable PLC Type VB	Applicable PLC Type VH	Ref. Page
Program Flow	00		CJ	P	Conditional jump	○	○	○	110
	01		CALL	P	Call subroutine	○	○	○	111
	02		SRET		Subroutine return	○	○	○	111
	03		IRET		Interrupt return	○	○	○	112
	04		EI		Enable interrupt	○	○	○	112
	05		DI		Disable interrupt	○	○	○	112
	06		FEND		First end	○	○	○	113
	07		WDT	P	Watch Dog Timer refresh	○	○	○	114
	08		FOR		Start of a FOR-NEXT loop	○	○	○	115
	09		NEXT		End of a FOR-NEXT loop	○	○	○	115
Compare and Move	10	D	CMP	P	Compare	○	○	○	118
	11	D	ZCP	P	Zone compare	○	○	○	119
	12	D	MOV	P	Move	○	○	○	120
	13		SMOV	P	Shift move	○	○	○	121
	14	D	CML	P	Compliment	○	○	○	122
	15		BMOV	P	Block move	○	○	○	123
	16	D	FMOV	P	Fill move	○	○	○	124
	17	D	XCH	P	Exchange	○	○	○	125
	18	D	BCD	P	Converts BIN → BCD	○	○	○	126
	19	D	BIN	P	Converts BCD → BIN	○	○	○	126
Arithmetic and Logical Operations	20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○	128
	21	D	SUB	P	Subtraction (S1) − (S2) → (D)	○	○	○	129
	22	D	MUL	P	Multiplication (S1)×(S2) → (D+1,D)	○	○	○	130
	23	D	DIV	P	Division (S1)÷(S2) → (D), (D+1)	○	○	○	131
	24	D	INC	P	Increment (D)+1 → (D)	○	○	○	132
	25	D	DEC	P	Decrement (D)−1 → (D)	○	○	○	132
	26	D	WAND	P	Logic word AND (S1) ∧ (S2) → (D)	○	○	○	133
	27	D	WOR	P	Logic word OR (S1) ∨ (S2) → (D)	○	○	○	133
	28	D	WXOR	P	Logic word exclusive OR (S1) $\veebar$ (S2) → (D)	○	○	○	133
	29	D	NEG	P	Negation ($\overline{D}$)+1 → (D)	○	○		134
Rotary and Shift	30	D	ROR	P	Rotation Right	○	○	○	136
	31	D	ROL	P	Rotation Left	○	○	○	136
	32	D	RCR	P	Rotation Right with carry	○	○	○	137
	33	D	RCL	P	Rotation Left with carry	○	○	○	137
	34		SFTR	P	Bit shift Right	○	○	○	138
	35		SFTL	P	Bit shift Left	○	○	○	138
	36		WSFR	P	Word shift Right	○	○		139
	37		WSFL	P	Word shift Left	○	○		140
	38		SFWR	P	Shift register write (FIFO Write)	○	○	○	141
	39		SFRD	P	Shift register read (FIFO Read)	○	○	○	142
Data Operation	40		ZRST	P	Zone reset	○	○	○	144
	41		DECO	P	Decode	○	○	○	145
	42		ENCO	P	Encode	○	○	○	146
	43	D	SUM	P	The sum of active bits	○	○		147
	44	D	BON	P	Check specified bit status	○	○		148
	45	D	MEAN	P	Mean	○	○		149
	46		ANS		Timed annunciator set	○	○		150
	47		ANR	P	Annunciator reset	○	○		150
	48	D	SQR	P	Square root	○	○		152
	49	D	FLT	P	BIN integer → Binary floating point format	○	○		153

* D ~ A 32 bit mode instruction option.

* P ~ Pulse (signal) operation option.

* ○ ~ The applicable PLC type

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
High-speed Processing	50		REF	P	I/O refresh	○	○	○	156
	51		REFF	P	I/O refresh and filter adjust	○	○		157
	52		MTR		Input matrix	○	○		158
	53	D	HSCS		High Speed Counter set	○	○	○	159
	54	D	HSCR		High Speed Counter reset	○	○	○	161
	55	D	HSZ		High Speed Counter zone compare	○	○		162
	56		SPD		Speed detection	○	○	○	167
	57	D	PLSY		Pulse Y output	○	○	○	168
	58		PWM		Pulse width modulation	○	○	○	169
	59	D	PLSR		Variable speed of Pulse output		○	○	170
Handy Instruction									
	61	D	SER	P	Search	○	○		174
	62	D	ABSD		Absolute Drum sequencer	○	○	○	175
	63		INCD		Incremental Drum sequencer	○	○	○	177
	64		TTMR		Teaching Timer	○	○		178
	65		STMR		Special Timer	○	○		179
	66		ALT	P	Alternate state	○	○	○	180
	67		RAMP		Ramp variable value	○	○	○	181
	69		SORT		Sort data	○	○		183
External Setting and Display	70	D	TKY		Ten Key input	○	○		186
	71	D	HKY		Hexadecimal Key input	○	○		187
	72		DSW		Digital Switch (Thumbwheel input)	○	○		189
	73		SEGD	P	Seven Segment Decoder	○	○	○	190
	74		SEGL		Seven Segment with Latch	○	○		191
	76		ASC		ASCII code Convert	○	○		193
	77		PR		Print	○	○		194
	78	D	FROM	P	Read from a special function block	○	○		195
	79	D	TO	P	Write to a special function block	○	○		195
External Serial Comm-unications	80		RS		Serial communication instruction	○	○	○	198
	81	D	PRUN	P	Parallel Run	○	○		202
	82		ASCI	P	Converts HEX → ASCII	○	○	○	203
	83		HEX	P	Converts ASCII → HEX	○	○	○	204
	84		CCD	P	Check Code	○	○	○	205
	85		VRRD	P	VR volume read	○	○	○	206
	86		VRSC	P	VR volume scale	○	○	○	207
	88		PID		PID control loop		○		352
	89		LINK		Easy Link communication	○	○		208
	149		MBUS		MODBUS communication		○	○	370
Floating Point	110	D	ECMP	P	Compares two BIN floating point values		○		214
	111	D	EZCP	P	Compares a BIN float range with a BIN float value		○		215
	118	D	EBCD	P	Converts BIN floating point format to DEC format		○		216
	119	D	EBIN	P	Converts DEC format to BIN floating point format		○		216
	120	D	EADD	P	Adds up two BIN floating point numbers		○		217
	121	D	ESUB	P	Subtracts one BIN floating point number from another		○		218
	122	D	EMUL	P	Multiplies two BIN floating point numbers		○		219
	123	D	EDIV	P	Divides one BIN floating point number from another		○		220
	127	D	ESQR	P	Square root of a BIN floating point value		○		221
	129	D	INT	P	BIN floating point → BIN integer format		○		222
	130	D	SIN	P	Calculates the sine of a BIN floating point value		○		223
	131	D	COS	P	Calculates the cosine of a BIN floating point value		○		224
	132	D	TAN	P	Calculates the tangent of a BIN floating point value		○		225

Type	FNC No.	Instruction Title D		P	Function	Applicable PLC Type M	VB	VH	Ref. Page
Others	90		DBRD	P	Reads data from the data bank	○	○		228
	91		DBWR	P	Writes data into the data bank	○	○		229
	147	D	SWAP	P	Swaps high/low byte	○	○		230
	169	D	HOUR		Operational Hour meter		○		376
	176		TFT		Timer (10 ms)	○	○	○	231
	177		TFH		Timer (100 ms)	○	○	○	232
	178		TFK		Timer (1 sec.)	○	○	○	233
Position Control	155	D	ABS		Absolute current value read		VB1		253
	156	D	ZRN		Zero position return		VB1		254
	157	D	PLSV		Pulse variable output		VB1		255
	158	D	DRVI		Drive to increment		VB1		256
	159	D	DRVA		Drive to absolute		VB1		257
Time & Convert	160		TCMP	P	Compare two times	○	○		236
	161		TZCP	P	Compare a time to a specified time range	○	○		237
	162		TADD	P	Adds ups two time values to get a new time	○	○		238
	163		TSUB	P	Subtracts one time value from another to get a new time	○	○		239
	166		TRD	P	Reads the RTC current value to a group of registers	○	○		240
	167		TWR	P	Sets the RTC to the value stored in a group of registers	○	○	○	241
	170	D	GRY	P	Converts BIN → Gray code	○	○		242
	171	D	GBIN	P	Converts Gray code → BIN	○	○		243
In-line Comparisons	224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○	246
	225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○	246
	226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○	246
	228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○	246
	229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○	246
	230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○	246
	232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○	246
	233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○	246
	234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○	246
	236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○	246
	237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○	246
	238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○	246
	240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○	246
	241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○	246
	242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○	246
	244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○	246
	245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○	246
	246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○	246
Newly Added Instructions	92		TPID		Temperature PID Control		V1.70		363
	250	D	SCL	P	Scaling (Translated by Coordinate)		V1.70		377
	251	D	SCL2	P	Scaling II (Translated by Coordinate)		V1.70		377
	151	D	DVIT		One-speed Interrupt Constant Quantity Feed		VB1		379
	153	D	LIR		Relatively Linear Interpolation		VB1		381
	154	D	LIA		Absolutely Linear Interpolation		VB1		384
	188		CRC	P	Cyclic Redundancy Check - 16		V1.72		387

Type	FNC No.	D	Instruction Title	P	Function	Applicable PLC Type M	VB	VH	Ref. Page
A	20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○	128
	46		ANS		Timed annunciator set	○	○		150
	47		ANR	P	Annunciator reset	○	○		150
	62	D	ABSD		Absolute Drum sequencer	○	○	○	175
	66		ALT	P	Alternate state	○	○	○	180
	76		ASC		ASCII code Convert	○	○		193
	82		ASCI	P	Converts HEX → ASCII	○	○	○	203
	155	D	ABS		Absolute current value read		VB1		253
	232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○	246
	233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○	246
	234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○	246
	236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○	246
	237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○	246
	238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○	246
B	15		BMOV	P	Block move	○	○	○	123
	18	D	BCD	P	Converts BIN → BCD	○	○	○	126
	19	D	BIN	P	Converts BCD → BIN	○	○	○	126
	44	D	BON	P	Check specified bit status	○	○		148
C	00		CJ	P	Conditional jump	○	○	○	110
	01		CALL	P	Call subroutine	○	○	○	111
	10	D	CMP	P	Compare	○	○	○	118
	14	D	CML	P	Compliment	○	○	○	122
	84		CCD	P	Check Code	○	○	○	205
	131	D	COS	P	Calculates the cosine of a BIN floating point value		○		224
	188		CRC	P	Cyclic Redundancy Check - 16		V1.72		387
D	05		DI		Disable interrupt	○	○	○	112
	23	D	DIV	P	Division (S1)÷(S2) → (D), (D+1)	○	○	○	131
	25	D	DEC	P	Decrement (D)–1 → (D)	○	○	○	132
	41		DECO	P	Decode	○	○	○	145
	72		DSW		Digital Switch (Thumbwheel input)	○	○		189
	90		DBRD	P	Reads data from the data bank	○	○		228
	91		DBWR	P	Writes data into the data bank	○	○		229
	151	D	DVIT		One-speed Interrupt Constant Quantity Feed		VB1		379
	158	D	DRVI		Drive to increment		VB1		256
	159	D	DRVA		Drive to absolute		VB1		257
E	04		EI		Enable interrupt	○	○	○	112
	42		ENCO	P	Encode	○	○	○	146
	110	D	ECMP	P	Compares two BIN floating point values		○		214
	111	D	EZCP	P	Compares a BIN float range with a BIN float value		○		215
	118	D	EBCD	P	Converts BIN floating point format to DEC format		○		216
	119	D	EBIN	P	Converts DEC format to BIN floating point format		○		216
	120	D	EADD	P	Adds up two BIN floating point numbers		○		217
	121	D	ESUB	P	Subtracts one BIN floating point number from another		○		218
	122	D	EMUL	P	Multiplies two BIN floating point numbers		○		219
	123	D	EDIV	P	Divides one BIN floating point number from another		○		220
	127	D	ESQR	P	Square root of a BIN floating point value		○		221
F	06		FEND		First end	○	○	○	113
	08		FOR		Start of a FOR-NEXT loop	○	○	○	115
	16	D	FMOV	P	Fill move	○	○	○	124
	49	D	FLT	P	BIN integer → Binary floating point format	○	○		153
	78	D	FROM	P	Read from a special function block	○	○		195
G	170	D	GRY	P	Converts BIN → Gray code	○	○		242
	171	D	GBIN	P	Converts Gray code → BIN	○	○		243
H	53	D	HSCS		High Speed Counter set	○	○	○	159
	54	D	HSCR		High Speed Counter reset	○	○	○	161
	55	D	HSZ		High Speed Counter zone compare	○	○		162
	71	D	HKY		Hexadecimal Key input	○	○		187
	83		HEX	P	Converts ASCII → HEX	○	○	○	204
	169	D	HOUR		Operational Hour meter		○		376

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
I	03		IRET		Interrupt return	○	○	○	112
	24	D	INC	P	Increment (D)+1 → (D)	○	○	○	132
	63		INCD		Incremental Drum sequencer	○	○	○	177
	129	D	INT	P	BIN floating point → BIN integer format		○		222
L	89		LINK		Easy Link communication	○	○		208
	153	D	LIR		Relatively Linear Interpolation		VB1		381
	154	D	LIA		Absolutely Linear Interpolation		VB1		384
	224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○	246
	225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○	246
	226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○	246
	228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○	246
	229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○	246
	230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○	246
M	12	D	MOV	P	Move	○	○	○	120
	22	D	MUL	P	Multiplication (S1)×(S2) → (D+1.D)	○	○	○	130
	45	D	MEAN	P	Mean	○	○		149
	52		MTR		Input matrix	○	○		158
	149		MBUS		MODBUS communication		○	○	370
N	09		NEXT		End of a FOR-NEXT loop	○	○	○	115
	29	D	NEG	P	Negation $(\overline{D})+1 \rightarrow (D)$	○	○		134
O	240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○	246
	241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○	246
	242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○	246
	244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○	246
	245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○	246
	246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○	246
P	57	D	PLSY		Pulse Y output	○	○	○	168
	58		PWM		Pulse width modulation	○	○	○	169
	59	D	PLSR		Variable speed of Pulse output		○	○	170
	77		PR		Print	○	○		194
	81	D	PRUN	P	Parallel Run	○	○		202
	88		PID		PID control loop		○		352
	157	D	PLSV		Pulse variable output		VB1		255
R	30	D	ROR	P	Rotation Right	○	○	○	136
	31	D	ROL	P	Rotation Left	○	○	○	136
	32	D	RCR	P	Rotation Right with carry	○	○	○	137
	33	D	RCL	P	Rotation Left with carry	○	○	○	137
	50		REF	P	I/O refresh	○	○	○	156
	51		REFF	P	I/O refresh and filter adjust	○	○		157
	67		RAMP		Ramp variable value	○	○	○	181
	80		RS		Serial communication instruction	○	○	○	198
S	02		SRET		Subroutine return	○	○	○	111
	13		SMOV	P	Shift move	○	○	○	121
	21	D	SUB	P	Subtraction (S1) – (S2) → (D)	○	○	○	129
	34		SFTR	P	Bit shift Right	○	○	○	138
	35		SFTL	P	Bit shift Left	○	○	○	138
	38		SFWR	P	Shift register write (FIFO Write)	○	○	○	141
	39		SFRD	P	Shift register read (FIFO Read)	○	○	○	142
	43	D	SUM	P	The sum of active bits	○	○		147
	48	D	SQR	P	Square root	○	○		152
	56		SPD		Speed detection	○	○	○	167
	61	D	SER	P	Search	○	○		174
	65		STMR		Special Timer	○	○		179
	69		SORT		Sort data	○	○		183
	73		SEGD	P	Seven Segment Decoder	○	○	○	190
	74		SEGL		Seven Segment with Latch	○	○		191
	130	D	SIN	P	Calculates the sine of a BIN floating point value		○		223
	147	D	SWAP	P	Swaps high/low byte	○	○		230
	250	D	SCL	P	Scaling (Translated by Coordinate)		V1.70		377
	251	D	SCL2	P	Scaling II (Translated by Coordinate)		V1.70		377

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
T	64		TTMR		Teaching Timer	○	○		178
	70	D	TKY		Ten Key input	○	○		186
	79	D	TO	P	Write to a special function block	○	○		195
	92		TPID		Temperature PID Control		○		363
	132	D	TAN	P	Calculates the tangent of a BIN floating point value		○		225
	160		TCMP	P	Compare two times	○	○		236
	161		TZCP	P	Compare a time to a specified time range	○	○		237
	162		TADD	P	Adds ups two time values to get a new time	○	○		238
	163		TSUB	P	Subtracts one time value from another to get a new time	○	○		239
	166		TRD	P	Reads the RTC current value to a group of registers	○	○		240
	167		TWR	P	Sets the RTC to the value stored in a group of registers	○	○	○	241
	176		TFT		Timer (10 ms)	○	○	○	231
	177		TFH		Timer (100 ms)	○	○	○	232
	178		TFK		Timer (1 sec.)	○	○	○	233
V	85		VRRD	P	VR volume read	○	○	○	206
	86		VRSC	P	VR volume scale	○	○	○	207
W	07		WDT	P	Watch Dog Timer refresh	○	○	○	114
	26	D	WAND	P	Logic word AND (S1) $\wedge$ (S2) → (D)	○	○	○	133
	27	D	WOR	P	Logic word OR (S1) $\vee$ (S2) → (D)	○	○	○	133
	28	D	WXOR	P	Logic word exclusive OR (S1) $\veebar$ (S2) → (D)	○	○	○	133
	36		WSFR	P	Word shift Right	○	○		139
	37		WSFL	P	Word shift Left	○	○		140
X	17	D	XCH	P	Exchange	○	○	○	125
Z	11	D	ZCP	P	Zone compare	○	○	○	119
	40		ZRST	P	Zone reset	○	○	○	144
	156	D	ZRN		Zero position return		VB1		254

MEMO

2. Function Description of Component

2-1 Component Tables

2-1-1 M Series PLC Component Table

Item	Description
Input at X	X0 ~ X777, 512 points, Numbered by octal.
Output at Y	Y0 ~ Y777, 512 points, Numbered by octal.

Auxiliary Relay (M)	General	M0 ~ M1999, 2000 points	
	Latched	M2000 ~ M5119, 3120 points	
	Special	M9000 ~ M9255, 256 points	
Step Relay (S)	Initial	S0 ~ S9, 10 points	
	General	S10 ~ S499, 490 points	
	Latched	S500 ~ S899, 400 points	
	For Annunciating	S900 ~ S999, 100 points, Latched	
Timer (T)	100 ms	T0 ~ T199, 200 points, for Subroutine T192 ~ T199	
	10 ms	T200 ~ T245, 46 points	
	1 ms (Retentive)	T246 ~ T249, 4 points, Latched	
	100 ms (Retentive)	T250 ~ T255, 6 points, Latched	
Counter (C)	16-bit Up	C0 ~ C99, 100 points	
		C100 ~ C199, 100 points, Latched	
	32-bit Up/Down	C200 ~ C219, 20 points	
		C220 ~ C234, 15 points, Latched	
High Speed Counter (C)	32-bit Up/Down, Latched	C235 ~ C245, 11 points, 1-Phase Counter	Total: 6 points Max.
		C246 ~ C250, 5 points, 2-Phase Counter	
		C251 ~ C255, 5 points, A/B Phase Counter	
Data Register (D)	General	D0 ~ D6999, 7000 points	
	Latched	D7000 ~ D8191, 1192 points	
	File Register	D1000 ~ D7999, 500 points for each unit, 7000 points Max.	
	Special	D9000 ~ D9255, 256 points	
Index Registers (V), (Z)		V0 ~ V7, Z0 ~ Z7, 16 points	
Branch Level (P)		P0 ~ P255, 256 points, for CJ, CALL use	
Interrupt Level (I)	External Interrupt	I00□ ~ I50□ , 6 points	
	Timer Interrupt	I6□ □ ~ I8□ □ , 3 points	
	Counter Interrupt	I010 ~ I060, 6 points	
Nest Level (N)		N0 ~ N7, 8 points, for MC and MCR	

Decimal Constants (K)	16 bits	-32,768 ~ 32,767
	32 bits	-2,147,483,648 ~ 2,147,483,647
Hexadecimal Constants (H)	16 bits	0H ~ FFFFH
	32 bits	0H ~ FFFFFFFFH

2-1-2 VB Series PLC Component Table

Item	Description		
Input at X	VB0 Series	X0 ~ X77, 64 points, ASCII	
	VB1 Series	X0 ~ X177, 128 points, ASCII	
	VB2 Series	X0 ~ X377, 256 points, ASCII	
Output at Y	VB0 Series	Y0 ~ Y77, 64 points, ASCII	
	VB1 Series	Y0 ~ Y177, 128 points, ASCII	
	VB2 Series	Y0 ~ Y377, 256 points, ASCII	
Auxiliary Relay (M)	General	M0 ~ M1999 and M4000 ~ M5119, Total 3120 points	
	Latched	M2000 ~ M3999, 2000 points	
	Special	M9000 ~ M9255, 256 points	
Step Relay (S)	Initial	S0 ~ S9, 10 points	
	General	S10 ~ S499, 490 points	
	Latched	S500 ~ S899, 400 points	
	For Annunciating	S900 ~ S999, 100 points, Latched	
Timer (T)	100mS	T0 ~ T199, 200 points, for Subroutine T192 ~ T199	
	10mS	T200 ~ T245, 46 points	
	1 ms (Retentive)	T246 ~ T249, 4 points, Latched	
	100 ms (Retentive)	T250 ~ T255, 6 points, Latched	
Counter (C)	16-bit Up	C0 ~ C99, 100 points	
		C100 ~ C199, 100 points, Latched	
	32-bit Up/Down	C200 ~ C219, 20 points	
		C220 ~ C234, 15 points, Latched	
High Speed Counter (C)	32-bit Up/Down, Latched	C235 ~ C245, 11 points, 1-Phase Counter	Total: 6 points Max.
		C246 ~ C250, 5 points, 2-Phase Counter	
		C251 ~ C255, 5 points, A/B Phase Counter	
Data Register (D)	General	D0 ~ D6999 and D7512 ~ D8191, Total 7680 points	
	Latched	D7000 ~ D7511, 512 points	
	File Register	D1000 ~ D7999, 500 points for each unit, 7000 points Max.	
	Special	D9000 ~ D9255, 256 points	
Index Registers (V), (Z)		V0 ~ V7, Z0 ~ Z7, 16 points	
Branch Level (P)		P0 ~ P255, 256 points, for CJ, CALL use	
Interrupt Level (I)	External Interrupt	I00□ ~ I50□ , 6 points	
	Timer Interrupt	I6□ □ ~ I8□ □ , 3 points	
	Counter Interrupt	I010 ~ I060, 6 points	
Nest Level (N)		N0 ~ N7, 8 points, for MC and MCR	
Decimal Constants (K)	16 bits	-32,768 ~ 32,767	
	32 bits	-2,147,483,648 ~ 2,147,483,647	
Hexadecimal Constants (H)	16 bits	0H ~ FFFFH	
	32 bits	0H ~ FFFFFFFFH	

2-1-3 VH Series PLC Component Table

Item	Description
Input at X	X0 ~ X77, 64 points, Numbered by octal.
Output at Y	Y0 ~ Y77, 64 points, Numbered by octal.

<table>
<tr><td rowspan="3">Auxiliary Relay (M)</td><td>General</td><td colspan="2">M0 ~ M383, 384 points</td></tr>
<tr><td>Latched</td><td colspan="2">M384 ~ M511, 128 points</td></tr>
<tr><td>Special</td><td colspan="2">M9000 ~ M9255, 256 points</td></tr>
<tr><td rowspan="2">Step Relay (S)</td><td>Initial</td><td colspan="2">S0 ~ S9, 10 points, Latched</td></tr>
<tr><td>Latched</td><td colspan="2">S10 ~ S127, 118 points</td></tr>
<tr><td rowspan="3">Timer (T)</td><td>100 ms</td><td colspan="2">T0 ~ T62, 63 points</td></tr>
<tr><td>10 ms</td><td colspan="2">T32 ~ T62, 31 points (When M9028＝ON)</td></tr>
<tr><td>1 ms</td><td colspan="2">T63, 1 point</td></tr>
<tr><td rowspan="2">Counter (C)</td><td rowspan="2">16-bit Up</td><td colspan="2">C0 ~ C15, 16 points</td></tr>
<tr><td colspan="2">C16 ~ C31, 16 points, Latched</td></tr>
<tr><td rowspan="3">High Speed Counter (C)</td><td rowspan="3">32-bit Up/Down, Latched</td><td>C235 ~ C245, 11 points, 1-Phase Counter</td><td rowspan="3">Total: 6 points Max.</td></tr>
<tr><td>C246 ~ C250, 5 points, 2-Phase Counter</td></tr>
<tr><td>C251 ~ C254, 4 points, A/B Phase Counter</td></tr>
<tr><td rowspan="3">Data Register (D)</td><td>General</td><td colspan="2">D0 ~ D127, 128 points</td></tr>
<tr><td>Latched</td><td colspan="2">D128 ~ D255, 128 points</td></tr>
<tr><td>Special</td><td colspan="2">D9000 ~ D9255, 256 points</td></tr>
<tr><td colspan="2">Index Registers (V), (Z)</td><td colspan="2">V0 ~ V7, Z0 ~ Z7, 16 points</td></tr>
<tr><td colspan="2">Branch Level (P)</td><td colspan="2">P0 ~ P63, 64 points, for CJ, CALL use</td></tr>
<tr><td rowspan="3">Interrupt Level (I)</td><td>External Interrupt</td><td colspan="2">I00□ ~ I50□ , 6 points</td></tr>
<tr><td>Timer Interrupt</td><td colspan="2">I6□ □ ~ I8□ □ , 3 points</td></tr>
<tr><td>Counter Interrupt</td><td colspan="2">I010 ~ I060, 6 points</td></tr>
<tr><td colspan="2">Nest Level (N)</td><td colspan="2">N0 ~ N7, 8 points, for MC and MCR</td></tr>
</table>

<table>
<tr><td rowspan="2">Decimal Constants (K)</td><td>16 bits</td><td>-32,768 ~ 32,767</td></tr>
<tr><td>32 bits</td><td>-2,147,483,648 ~ 2,147,483,647</td></tr>
<tr><td rowspan="2">Hexadecimal Constants (H)</td><td>16 bits</td><td>0H ~ FFFFH</td></tr>
<tr><td>32 bits</td><td>0H ~ FFFFFFFFH</td></tr>
</table>

2-2 Input Point X and Output Point Y

2-2-1 Input Point (X devices)

A PLC via Input Points to read the external status (switches or detectors ON/OFF signals) for the PLC operation.

2-2-2 Output Point (Y devices)

The coil of Output Points may direct drives external appliance. Via Output Relays or Transistors transmit the PLC operation result to the external devices. These contacts of coils are available set as either "normally open"(NO) or "normally closed"(NC) configuration, which handle various loads (Ex: motors, electromagnetic valves, and electromagnetic contactor .... etc.) to execute the control actions.

2-2-3 The Assigned I/O Point Identify Numbers of M Series

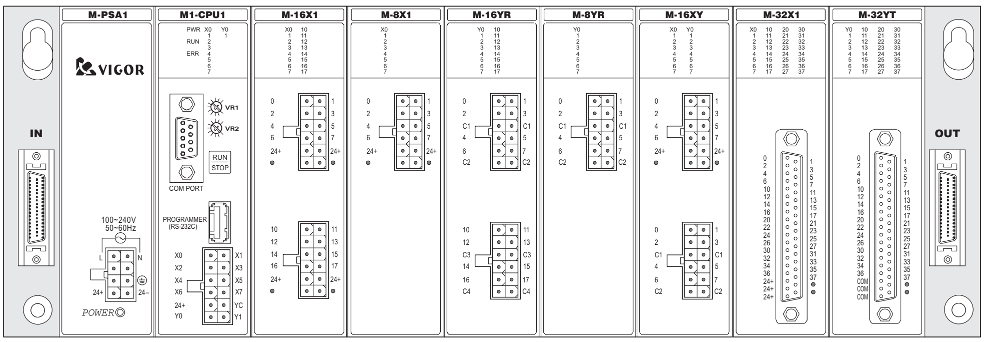
- The assigned identify numbers of Input Points use the ASCII codes, there will be 512 points available maximally. The ranges are: X0 ~ X7, X10 ~ X17,……., X770 ~ X777
- The assigned identify numbers of Output Points use the ASCII codes, there will be 512 points available maximally. The ranges are: Y0 ~ Y7, Y10 ~ Y17,……., Y770 ~ Y777
- The CPU module (M1-CPU1) will takes 16 input points and 16 output points; the X/Y assigned identify numbers are described as below:

Input (X)	Real accessible input points	X0 ~ X7
	Reserved for the system	X10 ~ X17
Output (Y)	Real accessible output points	Y0 and Y1
	Reserved for the system	Y2 ~ Y7,Y10 ~ Y17

- The X/Y assigned identify numbers of I/O module are arrange in order from left to right, start by the nearest CPU module. Here are the example diagrams below:

Ex1:

The CPU module and other I/O module installed in the M-8BS base

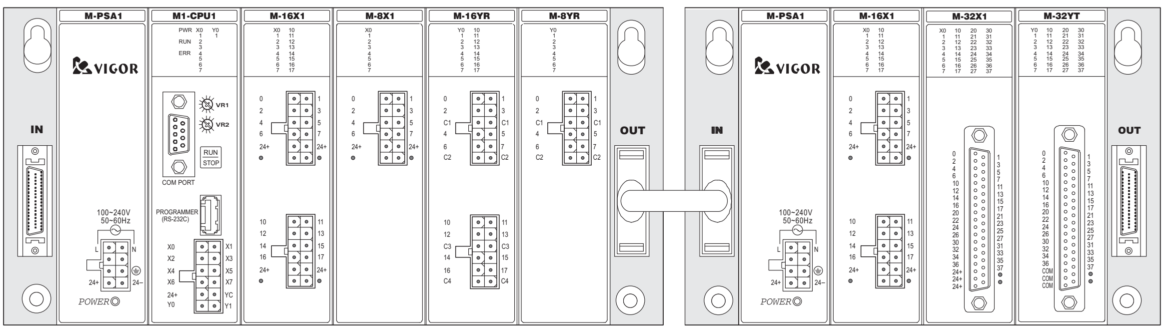


Power Module | CPU Module X0 ~ X7 Y0 ~ Y1 | 16points input X20 ~ X37 | 8points input X40 ~ X47 | 16points output Y20 ~ Y37 | 8points output Y40 ~ Y47 | 8points input 8points output X50 ~ X57 Y50 ~ Y57 | 32points input X60 ~ X117 | 32points output X60 ~ X117

Ex2:

The CPU module and other I/O module installed in a M-5BS base

The expanding I/O module in a M-3BS base



Power Module | CPU Module X0 ~ X7 Y0 ~ Y1 | 16points input X20 ~ X37 | 8points input X40 ~ X47 | 16points output Y20 ~ Y37 | 8points output Y40 ~ Y47

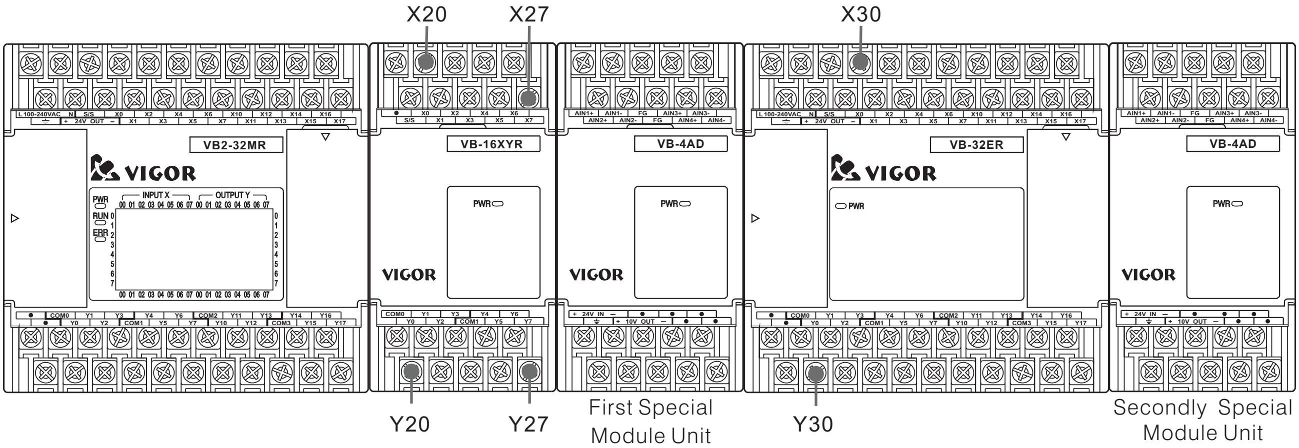
Power Module | 8points input 8points output X50 ~ X57 Y50 ~ Y57 | 32points input X60 ~ X117 | 32points output X60 ~ X117

2-2-4 The Assigned I/O Point Identify Numbers of VB Series

- The assigned identify numbers of Input/Output Points use the octal number code.
- The X/Y assigned identify numbers for the VB series Main Unit are described as below:

Models	VB0-14M	VB0-20M	VB0-28M	VB0-32M	VB1-14M	VB1-24M	VB1-32M	VB2-16M	VB2-32M
Input (X)	X0 ~ X7 (8 points)	X0 ~ X13 (12 points)	X0 ~ X17 (16 points)	X0 ~ X17 (16 points)	X0 ~ X7 (8 points)	X0 ~ X15 (14 points)	X0 ~ X17 (16 points)	X0 ~ X7 (8 points)	X0 ~ X17 (16 points)
Output (Y)	Y0 ~ Y5 (6 points)	Y0 ~ Y7 (8 points)	Y0 ~ Y13 (12 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y5 (6 points)	Y0 ~ Y11 (10 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y7 (8 points)	Y0 ~ Y17 (16 points)

- The X/Y assigned identify numbers diagram and descriptions for VB series Expansion Units:



- The X/Y assigned identify numbers for the VB series Main Unit are X0 ~ X17/Y0 ~ Y17 without exception. So, the first Expansion Module assigned I/O identify numbers will start at X20/Y20.
- The X/Y assigned identify numbers for the VB series Special Modules are K1 ~ K16, and they would not occupy any I/O port.
- The modules using BFM (Buffer Memory see P.196) to communicate with the Main Unit, which defined as Special Modules. The VB-PWR is a power extend module, it would not occupy the Special Module assigned identify numbers.
- The VB-8XY Expansion Module would occupy 8 input points and 8 output points.
- The maximum Input/Output points: VB0 series 128 points X0 ~ X77, Y0 ~ Y77
  VB1 series 256 points X0 ~ X177, Y0 ~ Y177
  VB2 series 512 points X0 ~ X377, Y0 ~ Y377
- The maximum available Special Modules: VB0 series 4 Special Modules Max.
  VB1 series 8 Special Modules Max.
  VB2 series 16 Special Modules Max.
- A Main Unit to use its I/O Expansion Slot connected with Expansion Units, Expansion Modules and Special Modules is available up to 31 units. (The VB1-14MT-D has no I/O Expansion Slot)
- The statement about expand:

  The VB series PLC Main Unit and Expansion Unit included a power supply unit, but the Expansion Module and Special Module does not have a power unit, those module needs a power source to get power (for example from a Main Unit, Expansion Unit or VB-PWR Power Expansion Unit).

  The statement of available modules amount with a Main Unit, Expansion Unit or VB-PWR Power Expansion Unit:

  Two important connecting limits from a Main Unit to Expansion Modules:

  (1) [(The amount of Expansion Modules)+(The amount of Special Modules)×2]≦4

  (2) All equipments using power form the Main Unit (including itself & Modules), the output points [(The amount of “ON” status relays×6)+(The amount of “ON” status transistors)]≦192

  Two important connecting limits from an Expansion Unit to Expansion Modules:

  (1) [(The amount of Expansion Modules)+(The amount of Special Modules)×2]≦12

  (2) All equipments using power form the Unit (including itself & Modules), the output points [(The amount of ”ON” status relays×6)+(The amount of “ON” status transistors)]≦192

  Two important connecting limits from a VB-PWR Power Expansion Unit to Expansion Modules:

  (1) [(The amount of Expansion Modules)+(The amount of Special Modules)×2]≦12

  (2) All equipments using power form the VB-PWR Power Expansion Unit, the output points [(The amount of “ON” status relays×6)+ (The amount of “ON” status transistors)]≦288

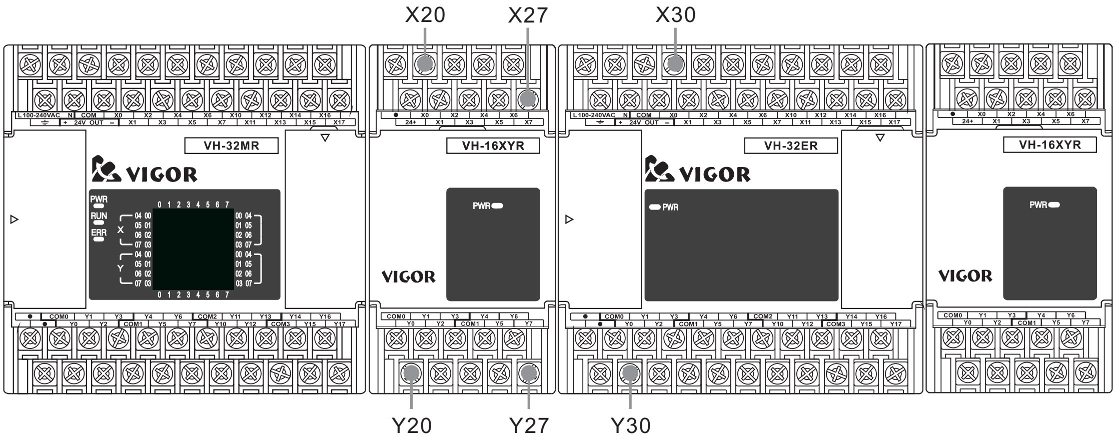
2-2-5 The Assigned I/O Point Identify Number of VH Series

- The assigned identify numbers of Input/Output Points use the octal number code.
- The X/Y assigned identify numbers for the VH series Main Unit are described as below:

Models	VH-10MR	VH-14MR	VH-20MR	VH-24MR	VH-28MR	VH-32MR	VH-40MR	VH-60MR
Input (X)	X0 ~ X5 (6 points)	X0 ~ X7 (8 points)	X0 ~ X13 (12 points)	X0 ~ X15 (14 points)	X0 ~ X17 (16 points)	X0 ~ X17 (16 points)	X0 ~ X27 (24 points)	X0 ~ X43 (36 points)
Output (Y)	Y0 ~ Y3 (4 points)	Y0 ~ Y5 (6 points)	Y0 ~ Y7 (8 points)	Y0 ~ Y11 (10 points)	Y0 ~ Y13 (12 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y27 (24 points)

- The VH-40MR is composed of a VH-32MR Main Unit and a VH-8X Expand Module.
- The VH-60MR is composed of a VH-32MR Main Unit and a VH-28XYR Expand Module.

- The X/Y assigned identify numbers diagram and descriptions for VH series Expansion Units:



- The VH-10MR , VH-14MR and VH-16MT-DI Main Unit are not available to use expand functions.
- The VH-20MR, VH-24MR, VH-28MR, VH-32MR and VH-20AR Main Unit occupied I/O identify numbers are X0 ~ X17/Y0 ~ Y17. So, the first Expansion Module assigned I/O identify numbers will start at X20/Y20.
- The VH-40MR Main Unit occupied I/O identify numbers are X0 ~ X27/Y0 ~ Y17.
- The VH-60MR Main Unit occupied I/O identify numbers are X0 ~ X47/Y0 ~ Y27.
- The VH-8XYR Expansion Module would occupy 8 input points and 8 output points.
- The VB-28XYR Expansion Module would occupy 24 input points and 8 output points.
- The maximum Input/Output points: 64 input points, X0 ~ X77
  64 output points, Y0 ~ Y77
- The statement about I/O expand:
  The VH series PLC Main Unit and Expansion Unit included a power supply unit, but the Expansion Module does not have a power unit, those modules need a power source to get power (from a Main Unit or Expansion Unit).
  Two important connecting limits from a Main Unit or Expansion Unit to Expansion Modules:
  (1) The amount of Expansion Modules ≦6
  (2) All equipments using the power form the power source unit (including the power source unit itself and Expansion Modules), the amount of "ON" status relays ≦32

2-3 Auxiliary Coil/Flag (M)

The PLC includes considerable internal Auxiliary Coils/Flags (M), the function of Auxiliary Coil/Flag (M) is a status (ON/OFF) storage, which provided data for the processing demand. The method of operate the Auxiliary Coils/Flags (M) is the same way to operate the Output Coils (Y), but the contact of Auxiliary Coil/Flag (M) can not directly drive an external load. The assigned Auxiliary Coil/Flag (M) identify number uses a decimal number and there are three functions to make the differentiation, the functions are list below :

(1) General Stable Auxiliary Coil/Flag

During the PLC operation (the power is "ON") the General Stable Auxiliary Coils will storage status, but all data in the coils will disappear when turn off the power or a power failure occurs. After the power retrieved, all data will be reset as initial status (OFF) in the coils.

(2) Latched Auxiliary Coil/Flag

During the PLC operation the Latched Auxiliary Coils will storage status, and all data in the coils will not disappear when turn off the power or a power failure occurs. After the power retrieved, the coils still kept the data as the moment before power failure occurs. Using a new status to overwrite the old status is the only way to change status in a Latched Auxiliary Coil.

(3) Special Diagnostic Auxiliary Coil/Flag

Every single Special Diagnostic Auxiliary Coil has its special function. Some of the assigned Special Diagnostic Auxiliary Coil only has a contact but without a output coil which is used the same identified number, it can not drive the coil in a program. Do not use any indefinite Special Diagnostic Auxiliary Coil. As regards the detail of the Special Diagnostic Auxiliary Coil, please refer to Section 2-13 "Special Coil and Special Register".

Series	General Stable Auxiliary Coil/Flag	Latched Auxiliary Coil/Flag	Special Diagnostic Auxiliary Coil/Flag
M	M0 ~ M1999, Total 2000 points	M2000 ~ M5119, Total 3120 points	M9000 ~ M9255, Total 256 points
VB	M0 ~ M1999, M4000 ~ M5119, Total 2000 points	M2000 ~ M3999, Total 2000 points	M9000 ~ M9255, Total 256 points
VH	M0 ~ M383, Total 384 points	M0 ~ M1999, Total 2000 points	M9000 ~ M9255, Total 256 points

2-4 State Coil (S)

The State Coil (S) is the basic component of the STL (STep Ladder chart). The assigned State Coil (S) identify number uses a decimal number and there are four functions to make the differentiation, the functions are list below :

(1) Initial State Coil

The Initial State Coil is used for initiation of a SFC (Sequential Function Chart).

(2) General Stable State Coil

It is the State Coils used in a SFC for the general purpose. During the PLC operation, all data in the coils will be returned to invalidity when turn off the power or a power failure occurs.

(3) Latched State Coils

When a power failure occurs during the PLC operation, all data in the Latched State Coils will be retained.

(4) Annunciator Flags

The Annunciator Flags feature Latched function, driving the instruction ANS (FNC 46) as the contact for an annunciator, which is used to record relevant alert messages so that troubleshooting can be performed.

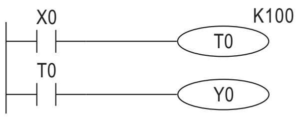
Series	Initial State Coil	General Stable State Coil	Latched State Coils	Annunciator Flags
M	S0 ~ S9, 10 points	S10 ~ S499, 490 points	S500 ~ S899, 400 points	S900 ~ S999, 100 points
VB	S0 ~ S9, 10 points	S10 ~ S499, 490 points	S500 ~ S899, 400 points	S900 ~ S999, 100 points
VH	S0 ~ S9, 10 points	—	S100 ~ S127, 118 points	—

2-5 Timer (T)

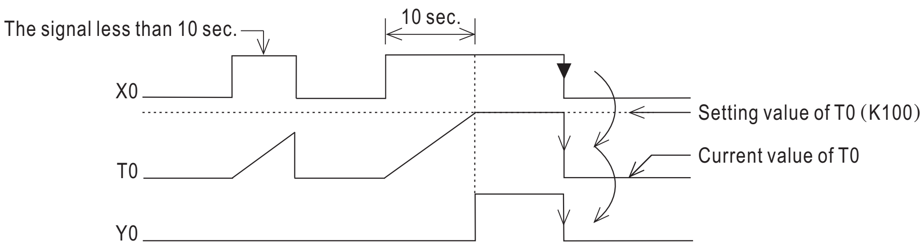
- The timers count the time by counting clock pulses.
  When the Current value = Setting value (the value designated to a Timer), the Timer contact will be activated (ON).
- To set the real Setting value of a Timer = Timer resolution × Designated number
- Timers can be set either directly by using the constant (K) to specify the maximum duration or indirectly by using the data stored in a Data Register (D). (Excluding the Special Data Registers D9000 ~ D9255)

Series	Non-retentive Timer					Retentive Timer	
	100 ms Timer 0.1 ~ 3276.7 sec.	M9028=OFF 100 ms Timer 0.1 ~ 3276.7 sec.	M9028=ON 10 ms Timer 0.01 ~ 327.67 sec.	10 ms Timer 0.01 ~ 327.67 sec.	1 ms Timer 0.001 ~ 32.767 sec.	1 ms Timer 0.001 ~ 32.767 sec.	100 ms Timer 0.1 ~ 3276.7 sec.
M	T0 ~ T199, 200 points	—	—	T200 ~ T245, 46 points	—	T246 ~ T249, 4 points	T250 ~ T255, 6 points
VB	T0 ~ T199, 200 points	—	—	T200 ~ T245, 46 points	—	T246 ~ T249, 4 points	T250 ~ T255, 6 points
VH	T0 ~ T31, 32 points	T32 ~ T62, 31 points		—	T63, 1 point	—	—

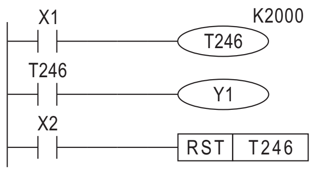
2-5-1 Non-retentive Timer



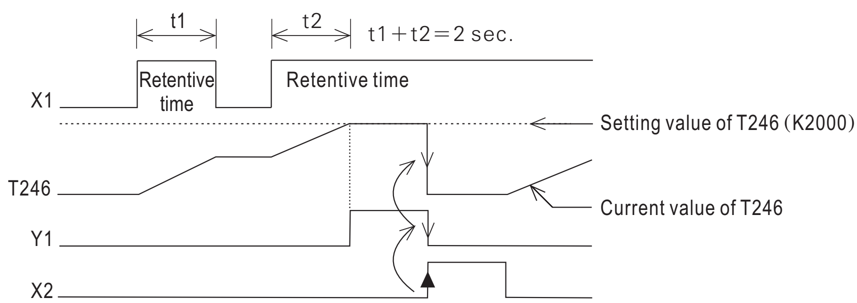
- When input contact X0 = "ON", the Current value of Timer T0 starts to count clock pulses (by 100ms), if the value reaches the Setting value K100 (10 sec.), the T0 contact will activated (ON).
- When input contact X0 = "OFF" or the power failure, the Current value of Timer will return to "0" and the contact will become "OFF".



2-5-2 Retentive Timer



- When input contact X1 = "ON", the Current value of Timer T246 starts to count clock pulses (by 1ms), if the Current value reaches the Setting value K2000 (2 sec.), the contact will activated (ON).
- During the counting time, T246 will stop counting if input contact X1 becomes "OFF" or PLC power failure. The current value will not be changed until the time when power reverted and input X1 received "ON" signal. When T246 resumes counting, the Current value will be retentively increased until Current value = Setting value K2000 (2 sec), and then the contact will become "ON".
- When input contact X2 = ON, the Current value of T246 will reset to "0" and the contact will become "OFF".

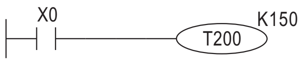


2-5-3 Attentions for Using Timer in Subroutine

For subroutines or inserted interruption subroutines, please use Timer T192 ~ T199. The timing action is updated once at the point when an "END" instruction is executed. The output contact is activated when a coil instruction or an "END" instruction is processed once the timers Current value has reached the Setting (maximum duration) value.

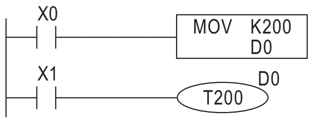
2-5-4 Specific Method for Setting Value

- Direct setting by a constant K



- T200 is a timer using a 10ms as the time unit resolution.
- If the Setting parameter = K150, then 10ms × 150 = 1500ms = 1.5 sec. , so the Timer T200 = 1.5 sec.

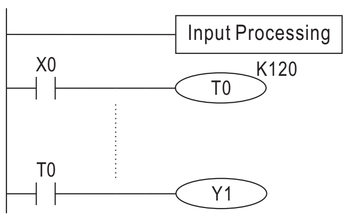
- Direct setting by a constant K

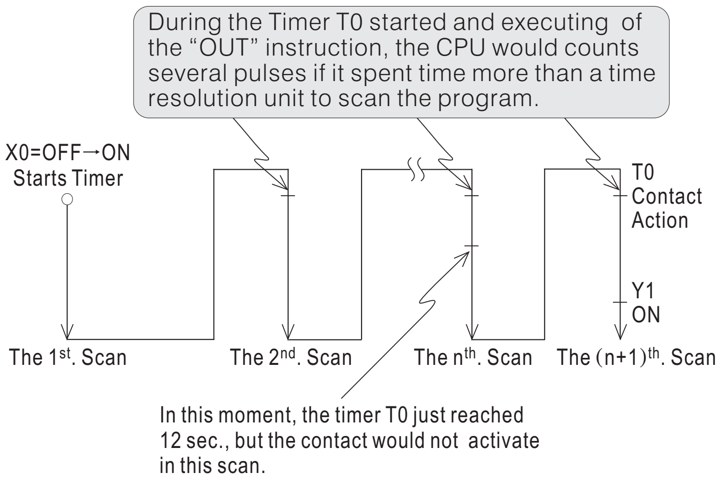


- T200 is a timer using a 10ms as the time unit resolution.
- T200 = 2 sec. if D0 = 200.
- T200 = 10 sec if D0 = 1000
- Counted time of T200 can be modified by changing the value of D0.

2-5-5 Timer Explicit Action and Accuracy

The action procedures of a timer (except the M, VB series T245 ~ T249 and VH series T63) is shown below:





From the action procedures above, the accuracy of the action, since the loop started to the contact "ON", is described as in the following:

$$T^{+T_s}_{-\alpha}$$

α : 0.01 sec. or 0.1 sec. for the timers 10ms or 100ms resolution
T : Setting time of Timer (sec.)
Ts : Scanning time (sec.)

- If in the program, the timer contact appears before the timer coil, the maximum timing error would extra 2Ts. If the setting value of the timer is "0", the output contact will starts the action in the next scan.
- For the interrupt timer (the M, VB series T245 ~ T249 and VH series T63), it starts to count time with 1ms Timer resolution pulse.

2-6 Counter (C)

- When the pulse input signal in a counter turned from "OFF" to "ON", the Current value of the counter will increases (+1 in a up count) / decreases (-1 in a down count) each time. If the Current value = Setting value, the output contact is activated and the coil turned "ON".
- Counters can be set either directly by using the constant (K) or indirectly by using the data stored in a Data Register (D). (Excluding the Special Data Registers D9000 ~ D9255)
- The characteristics of 16-bit and 32-bit Counters are displayed in the following table.

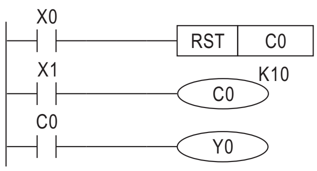
Item	16-bit Counter	32-bit Counter
Count Direction	Up Count	Convertible bi-directional, Up / Down Count
Available Setting Value Ranges	1 ~ 32,767 (1, if the Setting value exceeds beyond the range)	-2,147,483,648 ~ +2,147,483,647
Specified Setting Value	Constant K or Data Register	Same as left column, but each 32-bit value would occupy 2 Data Registers.
Change of Current Value	The Current value will not change when it reaches Setting value.	The Current value will continue to change when it reaches Setting value.
Output Contact	Retains "ON" when it reaches the Setting value	"ON", when Up Count reaches Setting value; "OFF", when Down Count reaches Setting value.
Reset Action	When the instruction RST is executed, the Current value will reset to "0" and the contact will return to "OFF".	
Current Value Register	16-bit	32-bit

- The assigned Counter identify numbers:

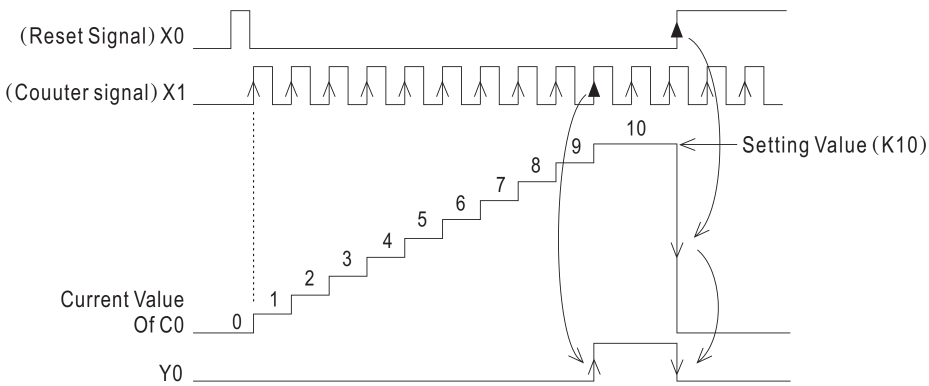
Series	16-bit Counter		32-bit Counter	
	General	Latched	General	Latched
M	C0 ~ C99, 100 points	C100 ~ C199, 100 points	C200 ~ C219, 20 points	C220 ~ C234, 15 points
VB	C0 ~ C99, 100 points	C100 ~ C199, 100 points	C200 ~ C219, 20 points	C220 ~ C234, 15 points
VH	C0 ~ C15, 16 points	C16 ~ C31, 16 points	—	—

2-6-1 16-bit Counter

- When the PLC power failed, the Current value in General Counters will be reset. But, the Latched Counters are able to retain the Current value, even after the PLC has been power failure, and the Current value will be accumulated right after the power is retrieved.

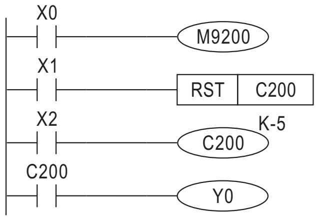


- If the input contact X1 turns OFF → ON once, the Current value of Counter C0 will increase "1". The value of Counter C0 is depend on input Counter Signal X1, the output contact C0 is activated (OFF → ON) when the Current value = 10. After this, the Current value remains unchanged (=10).
- If the input contact X0 = "ON", the instruction "RST" will executes, the Current value of C0 will reset to "0", and the contact will turn "OFF".

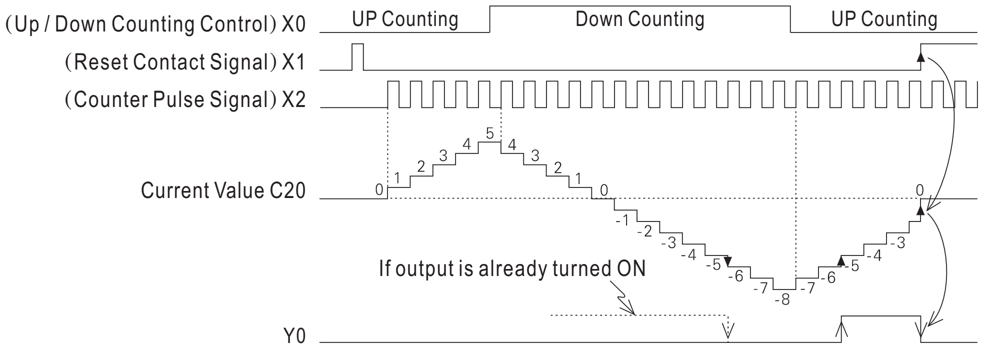


- The Counter's Setting value can using a Constant (K) or a Data Register (D).
- When the instruction "MOV" is used to transfer a value, which is greater than the counter Setting value, to the Current value Register. Until the input signal turning "ON", therefore the contact turns "ON" and the Counter's Current value would rewrites as the Setting value.

2-6-2 32-bit Counter



- X0 drives the Special Auxiliary Coil M9200 to define the Up/Down Count of C200, “OFF” is define as Up Count and “ON” as Down Count.
- When the input Counter Signal X2 = OFF → ON, the Counter C200 will
- When the C200’s Current value turns from “-6” into “-5”, the output contact will shift from “OFF” to “ON”. When the Current value turns from “-5” into “-6”, the output contact will shift from “ON” to “OFF”.
- When the Reset contact X0 = “ON”, the instruction RST will executes, the Current value of C200 will resets to “0” and the contact will turns “OFF”.

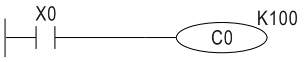


- Because the range of a 32-bit Counter value is between -2,147,483,647 to +2,147,483,647, if a counter counts beyond +2,147,483,647 the Current value will automatically change to -2,147,483,647. Similarly, counting below -2,147,483,647 will result in the current value in the Current value changing to +2,147,483,647. This type of counting technique is typical for “ring counters”.
- The Latched Counter is able to retain the Current value and contact status, even after the PLC has been power failure.
- A 32-bit Counter can be used as a 32-bit Data Register.
- When the instruction “DMOV” is used to transfer a value, which is greater than the counter Setting value, to the Current value Register. The next input pulse signal will be counted to Current value but the contact status will not be changes.
- The 32-bit UP/Down Counters C200 ~ C234 are using the Special Auxiliary Coils M9200 ~ M9234 to define as the Up/Down Count. The C200 is using M9200 to determine the direction as a Up/Down count, the C201 is using M9201,.... and so forth. Where if the Special Auxiliary Coil for the Counter is turned “ON”, the counter will be a Down counter; conversely, “OFF” for the Up counting.
- Counters can be set using either constants (K) or the data stored in Data Registers (D), and the value can be either positive or negative integer numbers. If using Data Registers, each 32-bit value would occupy 2 contiguous Data Registers.

2-6-3 The Appoint Method to Specify Setting Value

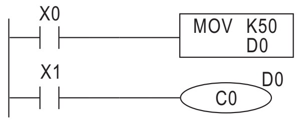
16-bit Counter

- Direct set by using constant (K)



- C0 becomes a Up Counter with 100 counts.

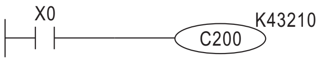
- Indirect set by using Data Register (D)



- C0 becomes a Counter with 50 counts, when D0=50.
- C0 becomes a Counter with 200 counts, when D0=200. To modify the count number of C0 by appointing the value of D0.

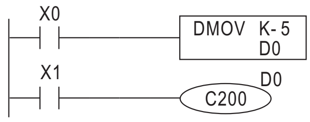
32-bit Counter

- Direct set by using constant (K)



- C200 becomes a UP/Down counter, and the Setting value is K43,210.

- Indirect set by using Data Register (D)



- Using the D1 and D0 to compose a 32-bit Register (D1 is for Up 16-bit; D0 is for Down 16-bit). When the value same as K-5, the C200 becomes a Up/Down Counter and the setting value is (-5).
- To modify the count number of C200 by appointing the value of D1 and D0.

2-7 High Speed Counter

There are 8 input points (X0 ~ X7) in the M series CPU module and VB, VH Series Main Unit. These 8 points have high speed input function such as High Speed Counter, External Interrupt Insertion and Speed Detection. If X0 ~ X7 are not applied to high speed input, they still can be used as common input points.

High Speed Counter receives high speed pulse inputs, it operates by the principle of inset interrupts to perform the purpose of high speed counting. All of the High Speed Counters are 32-bit Up/Down count devices, which provide latched function and can classified into 3 type of counters. The characteristics are shown as in the table below:

Assigned Counter ID No.	Counter Type	Count Direction	Default Range
C235 ~ C245	1-Phase High Speed Counter	Uses M9235 ~ M9245 to determine the direction of Up/Down count. "OFF" is for Up counting, and "ON" is for Down counting.	
C246 ~ C250	2-Phase High Speed Counter	Up/Down count has its individual input point, which count direction can be observed by M9246 ~ M9250. "OFF" means Up counting, otherwise "ON" means Down counting.	− 2,147,483,648 ~ +2,147,483,647
C251 ~ C255 (the VH series only provide C251 ~ C254)	A/B-Phase High Speed Counter	A/B-Phase input signal order determines the direction of Up/Down count. Up count: when the A-Phase signal is "ON", and then the B-Phase signal from "OFF" turns to "ON". Down count: when the A-Phase signal is "ON", and then the B-Phase signal from "ON" turns to "OFF". The count direction can be observed by M9251 ~ M9255, "OFF" is for Up counting, and "ON" is for Down counting.	

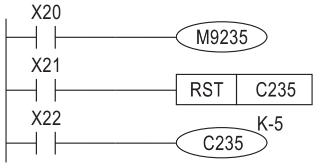
The following table lists the corresponding relationship between each high speed counter and X0 ~ X7 input points.

Input	1-Phase Counter											2-Phase Counter					A/B-Phase Counter					External interrupt insertion	Speed Detect
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255		
X0	U/D						U/D			U/D		U	U		U		A	A		A		I00□	○
X1		U/D					R			R		D	D		D		B	B		B		I10□	○
X2			U/D					U/D			U/D		R		R			R		R		I20□	○
X3				U/D				R			R			U		U			A		A	I30□	○
X4					U/D				U/D					D		D			B		B	I40□	○
X5						U/D			R					R		R			R		R	I50□	○
X6										S					S					S			
X7											S					S					S		

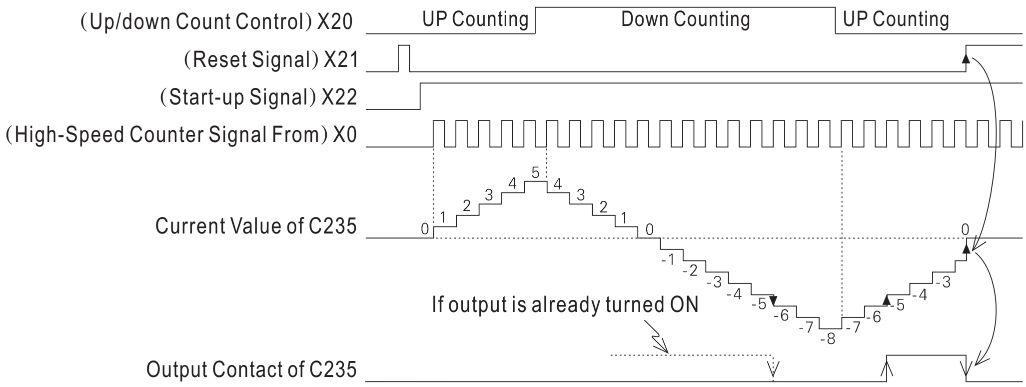
U: Up Counter Input; D: Down Counter input; A: A-Phase Counter Input; B: B-Phase Counter Input ;
U/D: Up / Down Count Input; R: Reset Counter Input; S: Start-up Counter Input

- In the table, C235 will occupies X0 input point, so if C235 is used, then other High-Speed Counters are driven by X0 (as listed in the table: C241, C244, C246, C247, C249, C251, C252 and C254) can not be used. And also, because the input X0 is occupied, the interrupt insertion and speed detection corresponding for X0 are useless .
- Since there is only X0 ~ X7 8 points high speed input, when some of the input points among X0 ~ X7 are occupied, other corresponding high-speed input functions can not repeated using same input point. Users must plan the system cautiously and operate the input points of X0 ~ X7 properly.
- The brief instruction in this page is only presented for High-Speed Counter. The actual planning should be referred to all functions of related high speed input point X0 ~ X7 and be considered altogether lest interference should occur.

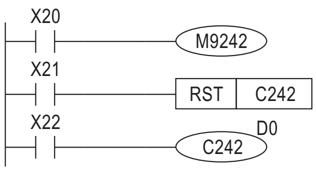
2-7-1 1-Phase High Speed Counter



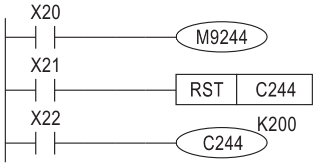
- X20 drives the special coil M9235 to determine the direction of Up/Down count to C235.
- When X22="ON", C235 is activation. From the previous counter table, the corresponding counted input for C235 is X0. Therefor C235 counts signal from X0 input point.
- When contact X21="ON", the instruction RST will be executed, the current value of C235 will be reset to "0", and the output contact will turn "OFF".
- C235 ~ C240 are 1-phase high speed counters featuring Software Startup Control and Software Return Control.



- When Start-up Signal X22="ON" and Pulse enters from X0 input point, the current value of C235 will be computed its Up/Down count.
- When the current value of the counter from -6 increased to -5 , the output contact will turn from "OFF" into "ON"; when the current value of the counter from -5 decreased to -6, the output contact will turn from "ON" into "OFF".
- If a counter counts beyond +2,147,483,647 the Current value will automatically change to -2,147,483,647. Similarly, counting below -2,147,483,647 will result in the current value in the Current value changing to +2,147,483,647. This type of counting technique is typical for "ring counters"
- When contact X21="ON", the instruction RST will be executed, the current value of C235 will be reset to "0", and the output contact will turn "OFF".
- The 1-Phase High Speed Counter C235 ~ C245 uses M9235 ~ M9245 to determine the direction of Up/Down count. "OFF" is for Up counting, and "ON" is for Down counting.

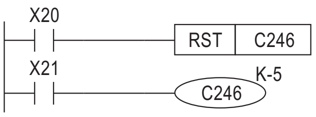


- X20 drives the special coil M9242 to determine the direction of Up/Down count to C242.
- When X22="ON", C242 is activation. From the previous counter table, the corresponding counted input for C242 is X2. Therefor C242 counts signal from X2 input point.
- When contact X21="ON", the instruction RST will be executed, the current value of C242 will be reset to "0", and the output contact will turn "OFF". If C242 is not reset by Software, the instruction RST may not be written.
- When X3="ON" (X3 is a hardware reset counter signal), the current value of C242 will be reset to "0", and its contact will turn "OFF".
- The setting value of C242 is configured depending on the contents of Data Registers D1 and D0
- C241 ~ C243 are 1-phase high speed counters featuring Software Start-up Control and Software/Hardware Reset Control.

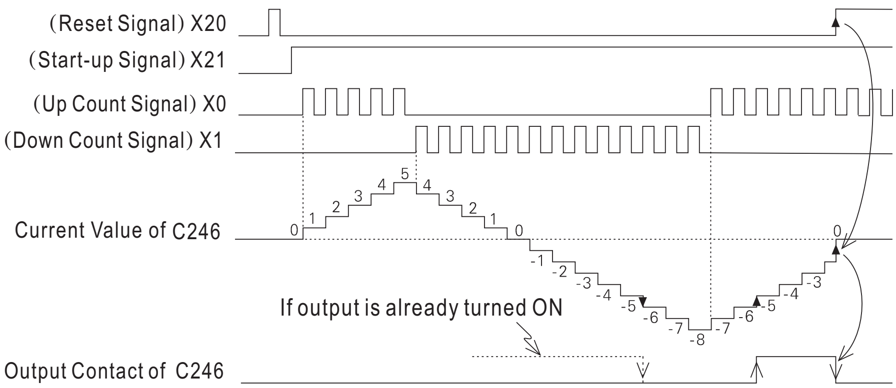


- X20 drives the special coil M9244 to determine the direction of Up/Down count to C244.
- When X22="ON" and X6="ON" (X6 is a hardware start counter signal), C244 is activation. From the previous counter table, the corresponding counted input for C244 is X0. Therefor C244 counts signal from X0 input point.
- When contact X21="ON", the instruction RST will be executed, the current value of C244 will be reset to "0", and the output contact will turn "OFF". If C244 is not reset by Software, the instruction RST may not be written.
- When X1="ON" (X1 is a hardware reset counter signal), the current value of C244 will be reset to "0", and its contact will turn "OFF".
- C244 ~ C245 are 1-phase high speed counters featuring Software/Hardware Start-up Control and Software/Hardware Return Control.

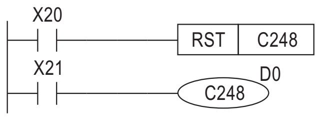
2-7-2 1-Phase High Speed Counter



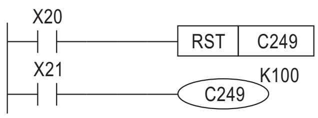
- When X21="ON", C246 is activation. Therefor C246 counts signal from X0 and X1 input points.
- When contact X20="ON", the instruction RST will be executed, the current value of C246 will be reset to "0", and the output contact will turn "OFF".
- C246 is a 2-phase high speed counters featuring Software Start-up Control and Software Reset Control.



- When Start-up Signal X21= "ON" and Pulse signal enters from X0 or X1 input point, the current value of C246 will be computed its Up/Down count.
  When X0= "OFF" → "ON", the current value of C246 will increase "1".
  When X1= "OFF" → "ON", the current value of C246 will decrease "1".
- When the current value of the counter from -6 increased to -5 , the output contact will turn from "OFF" into "ON" ; when the current value of the counter from -5 decreased to -6, the output contact will turn from "ON" into "OFF".
- If a counter counts beyond +2,147,483,647 the Current value will automatically change to -2,147,483,647. Similarly, counting below -2,147,483,647 will result in the current value in the Current value changing to +2,147,483,647. This type of counting technique is typical for "ring counters".
- The 2-Phase High Speed Counter C246 ~ C250 uses M9246 ~ M9250 to monitor the Up/Down count direction. "OFF" is Up counting, and "ON" is Down counting.



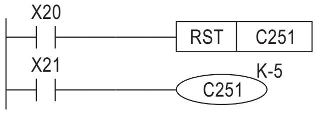
- When X21="ON", C248 is activation. From the previous counter table, the corresponding counted input for C248 is X3 and X4. Therefor C248 counts signal from X3 and X4 input point.
  When X3= "OFF"→"ON", the current value of C248 will increase "1".
  When X4= "OFF"→"ON", the current value of C248 will decrease "1".
- When contact X20= "ON", the instruction RST will be executed,the current value of C248 will be reset to "0", and the output contact will turn "OFF". If C248 is not reset by Software, the instruction RST may not be written.
- When X5= "ON" (X5 is a hardware reset counter signal), the current value of C248 will be reset to "0", and its contact will turn "OFF".
- The setting value of C248 is configured depending on the contents of Data Registers D1 and D0.
- C247 ~ C248 are 2-phase high speed counters featuring Software Start-up Control and Software/Hardware Reset Control.



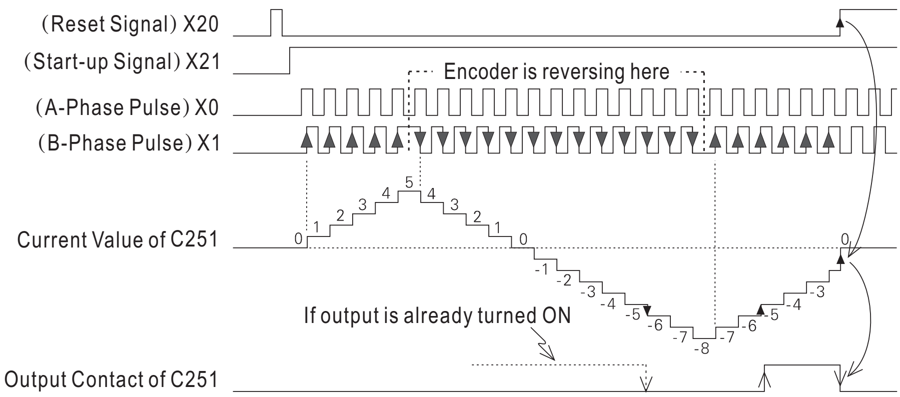
- When X21="ON" and X6= "ON" (X6 is a hardware start counter signal), C249 is activation. From the previous counter table, the corresponding counted input for C249 is X0 and X1. Therefor C249 counts signal from X0 and X1 input points.
  When X0= "OFF"→"ON", the current value of C249 will increase "1".
  When X1= "OFF"→"ON", the current value of C249 will decrease "1".
- When contact X20="ON", the instruction RST will be executed, the current value of C249 will be reset to "0", and the output contact will turn "OFF". If C249 is not reset by Software, the instruction RST may not be written.
- When X2= "ON" (X2 is a hardware reset counter signal), the current value of C249 will be reset to "0", and its contact will turn "OFF".
- C249 ~ C250 are 2-phase high speed counters featuring Software/Hardware Start-up Control and Software/Hardware Return Control.

2-7-3 A/B-Phase High Speed Counter

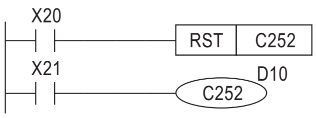
A/B-Phase High Speed Counter is used exclusively as the high speed counter receiving Rotary Encoder's A/B-Phase Pulse.



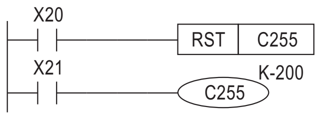
- When X21= "ON", C251 is activation and calculates the ON/OFF events of input X0 (the A Phase input) and input X1 (the B Phase input), and by the relationship of input signal sequence to execute Up/Down count.
- When contact X20= "ON", the instruction RST will be executed, the current value of C251 will be reset to "0", and the output contact will turn "OFF".
- C251 is a A/B-phase high speed counters featuring Software Start-up Control and Software Reset Control.



- When Start-up Signal X21= "ON" and A/B-Phase Pulse signal enters from X0 and X1 input point, the current value of C251 will be computed its Up/Down count.
  When X0 (A-Phase state)= "ON" and X1 (B-Phase state)= "OFF" → "ON", the current value of C251 will increase "1".
  When X0 (A-Phase state)= "ON" and X1 (B-Phase state)= "ON" → "OFF", the current value of C251 will decrease "1".
- The A/B-Phase High Speed Counter C251 ~ C255 uses M9251 ~ M9255 to monitor the Up/Down count direction. "OFF" is Up counting, and "ON" is Down counting.
- When a Rotary Encoder connected to a motor shaft, it will according to motor status (forward or reverse) to produce A/B-phase pulse signal. And then, the signal is transferred to the A/B-phase input points of C251, the current value of C251 will be increasing or decreasing correspond to motor runs forwarding or reversing.



- When X21="ON", C252 is activation and calculates A/B-Phase signal from X0 and X1 input points.
  When X0= "ON" and X1= "OFF" → "ON", the current value of C252 will increase "1".
  When X0= "ON" and X1= "ON" → "OFF", the current value of C252 will decrease "1".
- When contact X20= "ON", the instruction RST will be executed, the current value of C252 will be reset to "0", and the output contact will turn "OFF". If C252 is not reset by Software, the instruction RST may not be written.
- When X2= "ON" (X2 is a hardware reset counter signal), the current value of C252 will be reset to "0", and its contact will turn "OFF". The setting value of C252 is configured depending on the contents of Data Registers D111 and D10.
- C252 ~ C253 are A/B-phase high speed counters featuring Software Start-up Control and Software/Hardware Reset Control.

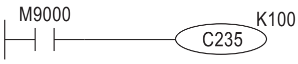


- When X21=“ON” and X7=”ON”(X7 is a hardware start counter signal), C255 is activation and calculates A/B-Phase signal from X3 and X4 input points.
  When X3= “ON” and X4= “OFF” → “ON”, the current value of C255 will increase “1”.
  When X3= “ON” and X4= “ON” → “OFF”, the current value of C252 will decrease “1”.
- When contact X20= “ON”, the instruction RST will be executed, the current value of C255 will be reset to “0”, and the output contact will turn “OFF”. If C255 is not reset by Software, the instruction RST may not be written.
- When X5= “ON” (X5 is a hardware reset counter signal), the current value of C255 will be reset to “0”, and its contact will turn “OFF”.
- C254 ~ C255 are A/B-phase high speed counters featuring Software/Hardware Start-up Control and Software/Hardware Return Control.

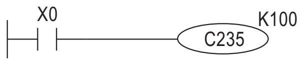
2-7-4 Precautions for Using High Speed Counteroutine

To activate High Speed Counter

- In the program, the conditional input contacts for activate High Speed Counters are NOT used to drive the counter coils. This is because the counter coils need to keep in status "ON" continuously to reserve the associated high speed input signals. Therefor, a normal non-high speed drive contact should be used to drive the high speed counter coil. If using non-high speed contacts direct drive the counters, it will cause wrong calculation.
- Ideally the special auxiliary contact M9000 should be used for activate. However, this is not compulsory.

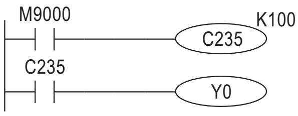


Correct program



Wrong program

The output of High Speed Counter



High Speed Counters receive high speed pulse inputs, they operate by the principle of inset interrupts to perform the purpose of high speed counting, they are irrelevant to Scan Time. So when the counter's Current value = Setting value, the counter's output contact (the status inside the memory) will be changed right away. But the status of Y0 as the chart above will be actually transferred to the output point only when the instruction END is executed. Which is still relevant to Scan Time and not a real-time transference. If a real-time output is desirable, users must use the high speed comparison instructions FNC53 (DHSCS), FNC54 (DHSCR) and FNC55 (DHSZ) exclusive for High Speed Counter.

Response Speed of High Speed Counter

- When a High Speed Counter is used in a program, the input point corresponding to the counter will be changed to a high speed input point (50μs response speed) automatically.
- When the instruction SPD is used in a program, the external input point specified by the instruction will be changed to a high speed input point (50μs response speed) automatically.
- The highest input count frequency of 1-Phase and 2-Phase High Speed Counter is up to 10 kHz. And the A/B-Phase High Speed Counter is up to 5 kHz.
- The highest count frequency accepted by the instruction SPD is up to 10 kHz.
- All count pulses of High Speed Counters and the instruction SPD is performed by interrupt insertion, where the total of the highest interrupt inserted frequency should not exceed 20 kHz (M, VB and VH series).

The calculation method of the total interrupt inserted frequency:

(Total 1-Phase Count Frequency) + (Total 2-Phase Count Frequency) + (Total A/B-Phase Count Frequency) × 2 + SPD Input Pulse Frequency = Total Interrupt Inserted Frequency (the value should not exceed 20 kHz)

2-8 Data Register (D)

A Data Register is a storage device capable of storing numeric data in 16/32-bit patterns. A single data unit contains 16 bits, while the MSB (Most Significant Bit) is used to indicate the data has a positive (0) or negative (1) bias, where the data ranging from -32,768 to +32,767 can be stored. However, two consecutive 16-bit registers can be used as a 32-bit register. The last 16 bits is defined as “lower” 16 bits and the first 16 bits is defined as “higher” 16 bits, while the MSB will always be found in the first higher 16 bits to given the positive (0) or negative (1) bias, where the data ranging from -2,147,483,648 to +2,147,483,647 can be stored.

The Data Register functions are list below :

(1) General Register

- When the PLC is turned from “RUN” to “STOP” or power failure occur, all of the general data registers have their current contents overwritten with a “0”. If the special auxiliary coil M9033= “ON” and PLC is switched from “RUN” to “STOP”, data can be retained in the general registers. But, power failure will still clear all contents to “0”.
- When M and VB series PLC is in the operation mode of Parallel connection (VH series does not have this function), D499 ~ D509 is used as the data transference area.

(2) Latched Register

- During the PLC operation the Latched Register will storage data, and all data in the Register will not disappear when turn off the power or a power failure occurs. It still kept the data as the moment before power failure occurs.
- Using the instructions RST and ZRST to reset the data in the Latched Register.
- It is available to add a Data Bank Expansion Card to extend the Latched Register size.

M series Data Bank Expansion Card: M-DB1

M series PLC provide a slot for M-DB1 Data Bank Expansion Card. To install a M-DB1 can add 64K Words Latched storage space. Using the Data Bank rewrite instruction DBWR (FNC91) and Data Bank read instruction DBRD (FNC90) to transfer data between Data Register and Data Bank.

Since the M-DB1 is using the Flash ROM technique to storage data, the rewrite operate limited is 10,000 times. So, when the program using the instruction DBWR to rewrite data into M-DB1, better change it to the instruction DBWRP. The DBWRP can avoid useless operate of rewrite, and then extend the lifespan of the Flash ROM.

VB series Data Bank Expansion Card: VB-DB1R

VB series PLC provide a slot for VB-DB1R Data Bank Expansion Card. To install a VB-DB1R can add 128K Words Latched storage space. Using the Data Bank rewrite instruction DBWR (FNC91) and Data Bank read instruction DBRD (FNC90) to transfer data between Data Register and Data Bank.

Since the VB-DB1R is using the SRAM technique plus Lithium battery to storage data, the rewrite operate times is unlimited. But the Lithium battery lifespan is around 5 years, must pay attention on the maintenance of data storage.

(3) File Register

Please refer to Section 2-9, the instruction on “File Register” for details.

(4) Special Diagnostic Register

Each Special Diagnostic Register has its specific purpose of use. Mostly it is used for storing the system status, error messages, monitoring status. The details are described in Section 2-13 “Special Coil and Special Register”.

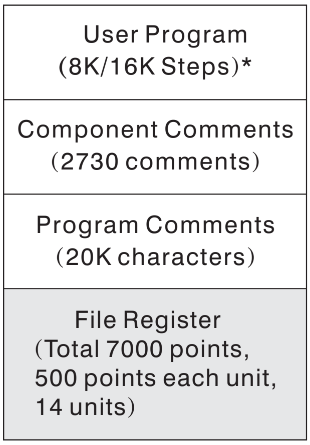
Series	General Register	Latched Register	File Register	Special Diagnostic Register
M	D0 ~ D6999, 7000 points	D7000 ~ D8191, 1192 points	D1000 ~ D7999, 7000 points	D9000 ~ D9255, 256 points
VB	D0 ~ D6999, D7512 ~ D8191, 7680 points	D7000 ~ D7511, 512 points	D1000 ~ D7999, 7000 points	D9000 ~ D9255, 256 points
VH	D0 ~ D127, 128 points	D128 ~ D255, 128 points	–	D9000 ~ D9255, 256 points

2-9 File Register (D)

The File Registers of M and VB Series PLC have 8192 points (D0 ~ D8191), where 7000 points (D1000 ~ D7999) can be planned and assigned as the identify numbers for File Register. The planning work is performed by peripherals (such as Ladder Master). The functions and characteristics of File Register are described below.

2-9-1 Structure and Characteristics of File Register

① The Outline of the Program Memory



The diagram of the Program Memory

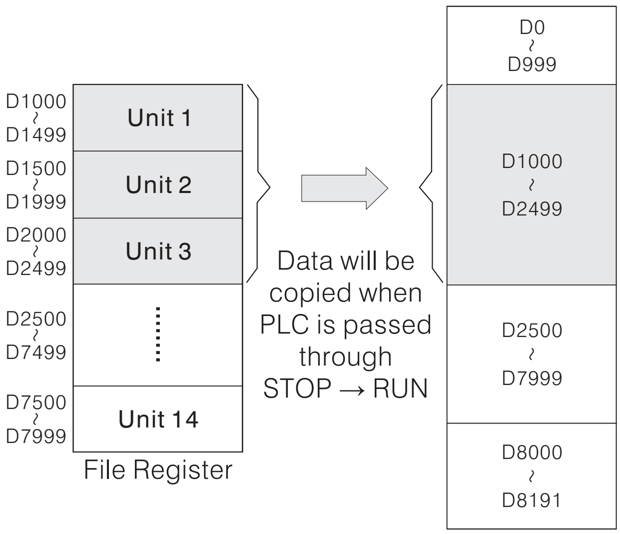
- Program Memory is using the built-in Flash ROM of CPU/Main Unit, or the Flash Rom in the Memory Card (if it is installed).
- Program Memory includes an 8K/16K* Steps User Program, 2730 Component Comments, 20K characters Program Comments and 7000 points of File Register. Each one has its own independent component partitions.
- A complete program will contains those four component partitions in the Program Memory. Thus, when the program is stored, opened, upload, downloaded or processed to copy the program into/from a memory card, the aforementioned component partitions shall be included.
- In the File Register, there are totally 7000 points, which are split into 14 units, 500 register points for each unit.

* VB2 / VB1 series: 16K Steps; VB0 series: 8K Steps;
M series: 8K Steps.

② Characteristics of File Register

- Since the File Register's content value is stored in the Non-Volatile component – Flash Rom, the data will not disappear when the power failure occurs.
- The relationship between Program and File Register is interdependent. File Register is a part of User Program, and the File Register's content will be influenced when the program is stored or retrieved. Accordingly, File Register is suitable for saving the system setting data; The Data Register is the data process and storage area during the program running, its content varies from time to time. Its characteristics are significantly different to the File Register.
- During the program processing, all the data under read or write operation are directed to Data Register. File Register write (M series only) and read (M and VB series) operations shall be directed by the instruction FNC15 (BMOV), which will be explained in Section 2-9-2.

③ Relationship between File Register and Data Register



File Register

Data Register

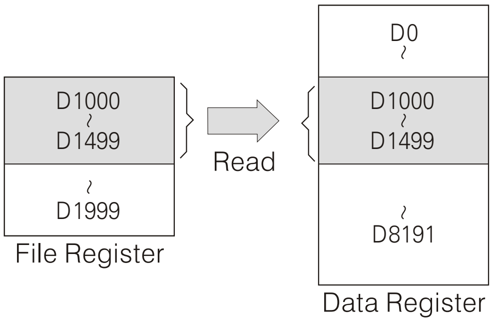
- The left chart explains the correlation between File Register and Data Register.
- The Ladder Master provides planning File Register and writing data functions.
- When users are planning the File Register, must divide D1000 ~ D7999 into 14 units (500 File Registers each). Beginning from the D1000, D1000 ~ D1499 is Unit1, D1500 ~ D1999 is Unit2 and so on. So if we are planning a 3-unit register, the range shall be D1000 ~ D2499 and there will be 1500 registers.
- Whenever PLC is passed through STOP → RUN, the content value of File Register will be automatically copied into the correlated Data Register.

2-9-2 File Register's Write/Read Operation

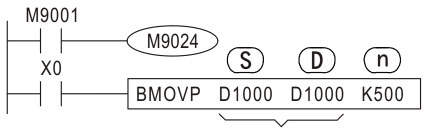
- The description below, all the File Register is hypothesized to be planned as 2-unit register, from D1000 ~ D1999 (1000 registers).
- File Register Write/Read Operation is implemented via the instruction FNC15 (BMOV). The M series provides File Register writing function, the VB series does not.
- The Special Coil M9024 is a control flag for the transfer direction of the instruction BMOV. The status (ON/OFF) of M9024 could designate the data transfer direction of the instruction BMOV.

When M9024 = "OFF" (S → D)   D0 ~ D99 → D100 ~ D199
When M9024 = "ON" (S ← D)   D0 ~ D99 ← D100 ~ D199

Reading from the File Register



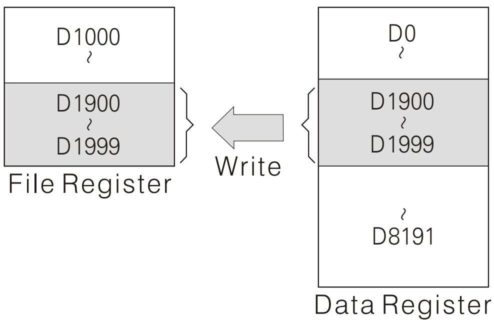
M9001: Always "OFF" Contact



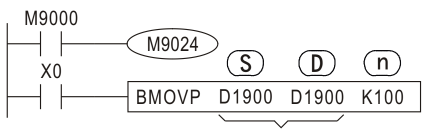
S and D assigned the same identify number of the File Register

- When appointed S and D using the same identified number of the File Register, which indicates the File Register via the instruction BMOV can do Write/Read Operation. As M9024 = "OFF" in the example above, the transfer direction of BMOV will be S → D to execute Read operation of the File Register.

Writing to the File Register (the VB series does not have the Writing operation)

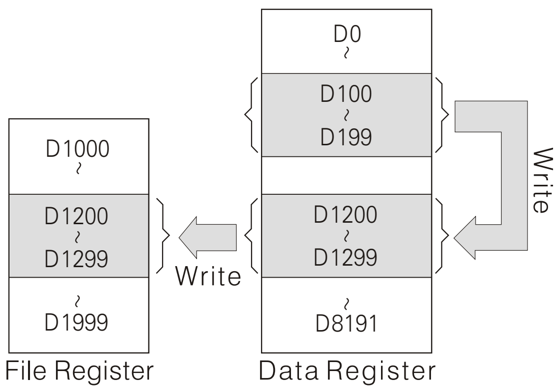


M9000: Always "ON" Contact

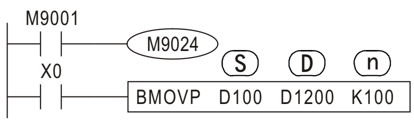


S and D assigned the same identify number of the File Register

- When appointed S and D using the same identified number of the File Register, which indicates the File Register via the instruction BMOV can do Write/Read Operation. As M9024 = "ON" in the example above, the transfer direction of BMOV will be S ← D to execute Write operation of the File Register.



M9001: Always "OFF" Contact



- When X0 = OFF → ON, the data in Data Register D100 ~ D199 will be transferred into the Data Register D1200 ~ D1299, and also written into File Register D1200 ~ D1299. But, if the Write operation of the File Register is not accessible, the Write operation to the Data Register will be executed.

- When appointed S and D using the same identified number of the File Register, the range is specified by "n", it can not exceed the range of File Register. Once it exceeds the range, which will deem as an instruction operational error, the instruction will not be executed.

2-9-3 Precautions for Using File Register

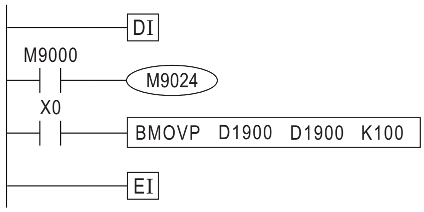
① Only the M series provided the File Register's Writing operation function, VB series can not write into the File Register.

② The File Register using Flash ROM memory to storage data, it is available to write more than 10,000 times, but still has the write times limited. When the program using the instruction DBWR to rewrite data into the File Register, better change it to the instruction DBWRP. The DBWRP can avoid useless operate of rewrite, and then extend the lifespan of the Flash ROM.

③ If the CPU module installed a Memory Card and in the program has a write operation for the File Register, must put the protective switch in “Writable” position at the card.

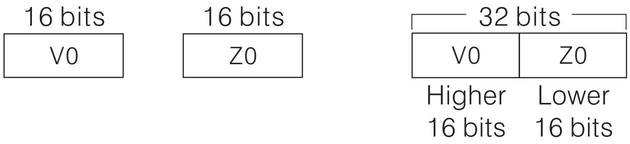
④ When the File Register’s Write operation is executed, every 64 points of File Register will spend 10ms to execute. And at the time the running program will be interrupted temporarily and the Watch Dog’s timing will be reset automatically.

⑤ Any interrupt insertion occurred during the File Register’s write operation, may cause errors to the execution results. So this is a suggestion: the interrupt insertion is prohibited to use when the write operation is executing. The chart shows below is using “DI” (Disable Interrupt) insertion to prohibited interrupts when the write operation is executing, after that using the “EI” (Enable Interrupt) insertion to regain interrupt.



2-10 Index Register (V) and (Z)

- The Index Register is a 16-bit register, the identified numbers are V0 ~ V7 and Z0 ~ Z7 (total 16 points).
- It's available to combine a Register V with a Register Z become a 32-bit Register. In the 32-bit applied instruction, V and Z can be assigned as a pair of register (V0, Z0) (V1, Z1)…(V7, Z7). Simply assign the Register Z, it can be assign the Operation Unit.



- Index Register can be used to decorate the Operand devices in the applied instruction. It can be used to modify the following devices under certain conditions; X, Y, M, S, P, T, C, D, K, H, K*n*X, K*n*Y, K*n*M and K*n*S.
- The use of Index Register will be explained in the Section 5-3 "General Principles of Applied Instructions".

2-11 Pointer (P) and Interrupt Pointer (I)

2-11-1 Pointer (P)

- The purpose of Pointer (P) is used to mark up a specific point in a program, and it is usually used to mark the destination of the CJ instruction or the start position of the CALL instruction's subroutine.
- The assigned numbers for the Pointers (P)

Series	Pointers (P)	Annotations
M	P0 ~ P255, 256 points	The Pointer P255 equals the position of END in a program.
VB	P0 ~ P255, 256 points	The Pointer P255 equals the position of END in a program.
VH	P0 ~ P63, 64 points	The Pointer P63 equals the position of END in a program.

2-11-2 Interrupt Pointer (I)

- The purpose of Interrupt Pointer (I) is used to mark up the start position of the interrupt subroutine of a program.
- The assigned numbers for the Interrupt Pointer (I):

Input Interrupt		Timer Interrupt	High Speed Counter Interrupt
External Input Terminal	Interrupt Pointer	Interrupt Pointer	Interrupt Pointer
X0	I00□	3 points: I6□□ I7□□ I8□□	6 Points: I010 I020 I030 I040 I050 I060
X1	I10□		
X2	I20□		
X3	I30□		
X4	I40□		
X5	I50□		
□ =1, indicates the interrupt during the rising □ =0, indicates the interrupt during the falling		□□ =01 ~ 99 indicate Timer Interrupt interval length, where the time interval will be 1 ~ 99ms	With the instruction FNC53 (DHSCS) to make a interrupt signal

- Interrupt Points can be discriminated into three types by functions: Input Interrupt, Timer Interrupt and High Speed Counter Interrupt.

①Input Interrupt: The rising or falling signal from the specific input terminal (X0 ~ X5) will produces A interrupt signal, it caused a interrupt to the running program, and jumps to the assigned Interrupt Pointer (I00□ ~ I50□) to execute the corres pondingly interrupted subroutine.

②Timer Interrupt: When the Timer Interrupt (I6□□ ~ I8□□) is written in the program, the PLC will automatically interrupt the running program at regular time (assigned by□□ of Timer Interrupt), and will jump to the assigned Interrupt Pointer to execute the correspondingly interrupted subroutine.

③High Speed Counter Interrupt: The FNC53 (DHSCS) High Speed Counter compare instruction's results can be assigned to execute the correspondingly interrupted subroutine. When the instruction DHSCS is assigned to process the interrupted subroutine (I010 ~ I060) and if the comparative results are equivalent to each other, the PLC will jump to the assigned Interrupt Pointer to execute the interrupted subroutine. Please consult the reference resources about the instruction FNC53 (DHSCS) for more detals.

- The application of Interrupt Pointer and the concepts of the interrupted subroutine will have detailed describe in the instructions IRET, EI and DI.

2-12 Numerical System

(1) Binary Number (BIN)

The value in PLC is operated and stored used the binary system. The binary number and relative terminology are given as follows:

① Bit: the basic of the binary number, each value of a Bit must be either “0” or “1”.

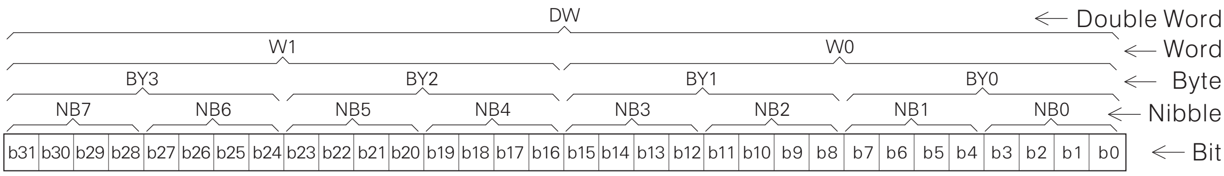
② Nibble: composed of 4 sequential bits.
Ex. b3 ~ b0 can express an one-Nibble hex value: 0 ~ F.

③ Byte: composed of 8 sequential bits.
Ex. b7 ~ b0 can express a two-Nibble hex value: 00 ~ FF.

④ Word: composed of 2 sequential bytes or 16 sequential bits.
Ex. b15 ~ b0 can express a four-Nibble hex value: 0000 ~ FFFF.

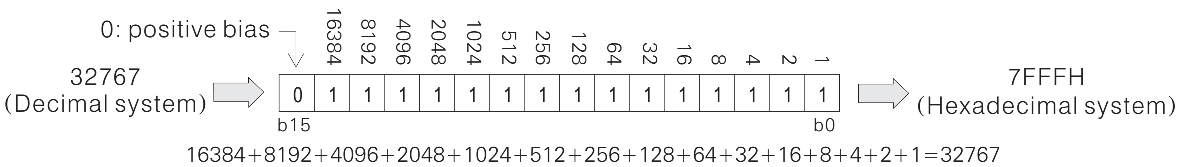
⑤ Double Word: composed of 2 sequential words, 4 sequential bytes or 32 sequential bits.
Ex. b31 ~ b0 can express an eight-Nibble hex value: 00000000 ~ FFFFFFFF.

⑥ The relations between every binary Bit, Nibble, Byte, Word and Double Word:



⑦ Expression of the value

For Word (16 bits) or Double Word (32 bits), the Most Significant Bit (MSB), e.g. The b15 of a Word or the b31 of a Double Word, gives the value positive or negative bias, where “0” for positive and “1” for negative. The rest bits, e.g. b14~b0 or b30 ~ b0, express the value size. It is a 16-bit value shows below.



⑧ Range of the value

The maximum range of the value expressed by 16 bits and 32 bits:

16 bits	–32,768 ~ 32,767
32 bits	–2,147,483,648 ~ 2,147,483,647

(2) Binary Number (BIN)

The assigned numbers of PLC’s external input and output terminals are displayed by the octal system. Ex.
external input ports: X0 ~ X7, X10 ~ X17
external output ports: Y0 ~ Y7, Y10 ~ Y17

(3) Decimal Number (DEC)

Decimal Number is the value system which people are familiar with. In PLC, a decimal number is always headed with a “K” in front of the value. Ex. K123 indicates a decimal number where the value is 123.
Application occasions of Decimal Number:

① Used as the setting value of T, C, for example, K10

② Used as the component number of M, S, T and C, for example, M9, S10, etc.

③ Used as an Operand device in the applied instruction, for example, MOV K1 D1.

(4) Binary Code Decimal (BCD)

BCD is to express a Decimal digit unit with a Nibble or 4 bits. Sequential 16 bits can express 4 Decimal digits. BCD is mainly used to read the input value of the Digital Switch (Thumbwheel input) or export the data to the 7-Segment Displayer for displaying the value.

(5) Hexadecimal Number (HEX)

In PLC, a Hex number is always headed with an "H", for example, H123 represents a Hex number and is valued 123.

(6) Bits of the numerical system and the numerical conversion table:

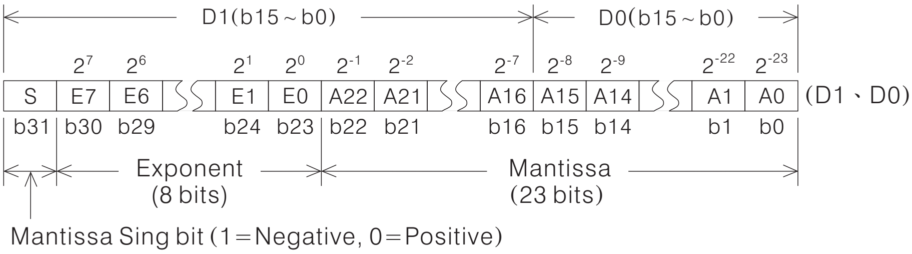
OCT	DEC	HEX	BIN		BCD	
0	0	00	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	1	01	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1
2	2	02	0 0 0 0	0 0 1 0	0 0 0 0	0 0 1 0
3	3	03	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1
4	4	04	0 0 0 0	0 1 0 0	0 0 0 0	0 1 0 0
5	5	05	0 0 0 0	0 1 0 1	0 0 0 0	0 1 0 1
6	6	06	0 0 0 0	0 1 1 0	0 0 0 0	0 1 1 0
7	7	07	0 0 0 0	0 1 1 1	0 0 0 0	0 1 1 1
10	8	08	0 0 0 0	1 0 0 0	0 0 0 0	1 0 0 0
11	9	09	0 0 0 0	1 0 0 1	0 0 0 0	1 0 0 1
12	10	0A	0 0 0 0	1 0 1 0	0 0 0 1	0 0 0 0
13	11	0B	0 0 0 0	1 0 1 1	0 0 0 1	0 0 0 1
14	12	0C	0 0 0 0	1 1 0 0	0 0 0 1	0 0 1 0
15	13	0D	0 0 0 0	1 1 0 1	0 0 0 1	0 0 1 1
16	14	0E	0 0 0 0	1 1 1 0	0 0 0 1	0 1 0 0
17	15	0F	0 0 0 0	1 1 1 1	0 0 0 1	0 1 0 1
20	16	10	0 0 0 1	0 0 0 0	0 0 0 1	0 1 1 0
⋮	⋮	⋮	⋮		⋮	
143	99	63	0 1 1 0	0 0 1 1	1 0 0 1	1 0 0 1

(7) Floating Point

The PLC was provided with Floating Point instructions therefore the PLC can calculate decimal numbers. The decimal numbers are storage and calculated in a PLC using two different pattern formats: Binary Floating Point Number and Decimal Floating Point Number. The expositions are showed below.

① Binary Floating Point Number

- Inside of the PLC, the Floating Point calculates and decimal number storages are using Binary Floating Point Numbers. A Binary Floating Point Number's value storage format is composed of 2 sequential registers. It is an example, using (D1,D0) to explain a format of a Binary Floating Point Number.



Binary Floating Point Number's value

$=\pm(2^0+A22\times2^{-1}+A21\times2^{-2}+.....+A1\times2^{-22}+A0\times2^{-23})$
$\times2^{(E7\times2^7+E6\times2^6+.....+E1\times2^1+E0\times2^0)}/2^{127}$

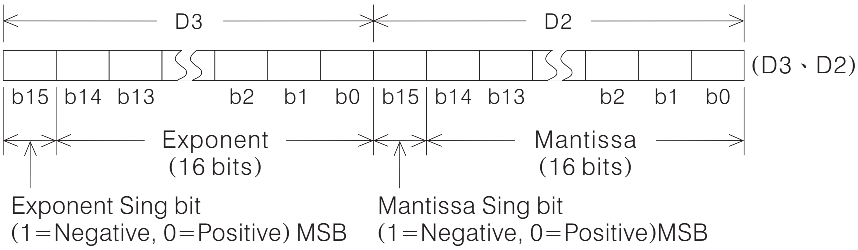
- If S=0, A22=1, A21=1, A20～A0=0
  E7=1, E6～E0=0

  Therefor, the Binary Floating Point Number's value storage in the register (D1,D0) is equal to
  $(2^0+1\times2^{-1}+1\times2^{-2}+.....+0\times2^{-23})\times2^{(1\times2^7+0\times2^6+.....+0\times2^0)}/2^{127}$
  $=1.75\times2^{128}/2^{127}=1.75\times2^1$

- A Binary Floating Point Number's value limit:
  Maximum modulus: 1.175×10^{-38} Minimum modulus: 3.402×10^{38}

② Decimal Floating Point Number

- A Decimal Floating Point Number's value storage format is also composed of 2 sequential registers. It is an example, using (D3,D2) to explain a format of a Decimal Floating Point Number.



Decimal Floating Point Number's value= (Mantissa) $\times10^{(\text{Exponent})}$
Mantissa $=\pm(1000～9999)$ Exponent= $-41～+35$

- If D2=1234, D3=-1

  Therefor, the Decimal Floating Point Number's value storage in the register (D3,D2) is equal to$1234\times10^{-1}=123.4$

- A Decimal Floating Point Number's value limit:
  Maximum modulus: 1175×10^{-41} Minimum modulus: 3402×10^{35}

- The Binary Floating Point Number and Decimal Floating Point Number can use the instructions to convert the value:

  FNC118 (DEBCD): To convert from a Binary Floating Point Number to a Decimal Floating Point Number.

  FNC119 (DEBIN): To convert from a Decimal Floating Point Number to a Binary Floating Point Number.

2-13 Special Coil and Special Register

In the tables below, the symbol "■" represents that itis not allowed to use a instruction to drive the coil or write the data to the program. And if the special coil or the special register is not listed in this table, which is reserved for the system and can not be used to drive the coil or write the data to the program either.

2-13-1 Table of Special Coil

Coil ID. No.	Instruction of Function	Series		
PLC Operation Status		M	VB	VH
■ M9000	An always "ON", "a" Contact, M9000 is "ON" during the running PLC.	○	○	○
■ M9001	An always "OFF", "a" Contact, M9001 is "OFF" during the running PLC.	○	○	○
■ M9002	Initial Pulse , "a" Contact, M9002 will be "ON" for a Scan Time when the moment PLC is STOP → RUN.	○	○	○
■ M9003	Initial Pulse , "b" Contact, M9003 will be "OFF" for a Scan Time when the moment PLC is STOP → RUN.	○	○	○
■ M9004	Error occurred. When one or more of the error flags M9060, M9063, M9066, M9067 are "ON", M9004= "ON" .	○	○	○
Clock Pulse		M	VB	VH
■ M9011	Oscillates 10ms cycles Pulse. "ON" 5ms/ "OFF" 5ms Pulse	○	○	○
■ M9012	Oscillates 100ms cycles Pulse. "ON" 50ms/ "OFF" 50ms Pulse	○	○	○
■ M9013	Oscillates 1sec. cycles Pulse. "ON" 0.5Sec/ "OFF" 0.5Sec Pulse	○	○	○
■ M9014	Oscillates 1min. cycles Pulse. "ON" 30Sec/ "OFF" 30Sec Pulse	○	○	○
System Status		M	VB	VH
■ M9005	M9005="ON" when the battery power of the Real Time Clock (RTC) is insufficient.	○	○	○
■ M9018	M9018="ON" when RTC is installed in the CPU module/Main Unit.	○	○	○
M9028	When M9028="OFF", T32 ~ T62 become an 100ms counter. When M9028="ON", T32 ~ T62 become an 10ms counter.			○
M9031	Clear the Non-Latched area memory. Current device settings are reset at next "END". All Coils Y, M, S, T, C are "OFF" and the current values of T, C, D become "0"; BUT except Special Coils M and D, which are not varied.	○	○	○
M9032	Clear the Latched area memory.	○	○	○
M9033	When M9033="ON" and RUN → STOP, the current value and statuses of T, C, D are retained.	○	○	○
M9034	All the outputs are disable. When M9034="ON", PLC's all external outputs are "OFF" but the program still operates normally.	○	○	○
M9039	Constant Scan Time duration. When M9039="ON", the PLC within a constant scan duration and defaulted by D9039.	○	○	○
M9083	For VB2 series only, to select the display range of I/O status. When M9083="OFF", shows the first 256 points; M9083="ON" shows the last 256 points.		○	
Flag		M	VB	VH
■ M9020	Zero Flag. M9020= "ON" when the result of an addition (ADD) or subtraction (SUB) is "0".	○	○	○
■ M9021	Borrow Flag. M9021= "ON" if any "Borrow" occurred to the result of the addition (ADD) or subtraction (SUB).	○	○	○
M9022	Carry Flag. M9022="ON" when any "Carry" occurred to the result of the addition (ADD) and subtraction (SUB).	○	○	○
■ M9029	Instruction execution completed flag. M9029="ON" when the executions of some applied instructions are completed (please refer to the relevant instructions).	○	○	○
■ M9131	Instruction execution completed flag for the identifies of instruction HSZ Multiple points comparison table has been processed.	○	○	
■ M9133	Instruction execution completed flag for the identifies of the instructions HSZ and PLSY (Pulse Y output at a set frequency) have been processed.	○	○	
■ M9199	Instruction execution completed flag for the identifies of instruction LINK (FNC80) or MBUS (FNC149) has been processed	○	○	
Assigning Specification of Applied Operation Instructions Mode		M	VB	VH
M9024	BMOV moves direction assigned. When M9024="OFF", S→D; Otherwise when M9024="ON", S ← D.	○	○	○
M9025	External HSC resets input mode. When M9025="OFF" and an external reset occurs, only the current value of HSC will be reset; when M9025="ON" and an external reset occurs, not only the current value of HSC will be reset but also the execution of relevant instructions will be restarted.	○	○	○
M9026	RAMP hold mode assigned. When M9026="OFF", a series of signals will be ramped by RAMP; Otherwise when M9026="ON", only one signal will be ramped by RAMP.	○	○	○
M9027	PR mode assigned. Please refer to PR (FNC 77) Instruction for details.	○	○	
M9130	Assigned the instruction HSZ to execute Multiple points compare mode.	○	○	
M9132	Assigned the instructions HSZ and PLSY to execute pulse variation frequency mode.	○	○	
M9161	Assigned an 8/16-bit process mode. When M9161="OFF" for a 16-bit process mode; and M9161="ON" for an 8-bit process mode	○	○	○

Note: Common alternatives are "a" and "b" identifiers for Normally Open (NO) ,Normally Closed (NC) states.

Coil ID. No.	Instruction of Function	Series		
Assigning Specification of Applied Operation Instructions Mode		M	VB	VH
M9167	HKY mode assigned. When M9167="OFF" for a "DEC" numeric mode, and M9167="ON" for a "HEC" numeric mode	○	○	
M9168	SMOV mode assigned. When M9168="OFF" for a "DEC" numeric mode, and M9168="ON" for a "HEC" numeric mode	○	○	○
Step Ladder Instruction Correlated Flags		M	VB	VH
M9040	STL transfer is prevented. When M9040="ON", the STL state transfer function is disabled.	○	○	○
■ M9046	STL state is ON. When M9047="ON" and any coil of S0 ~ S899="ON" than M9046="ON".	○	○	○
M9047	STL monitoring is enable. D9040 ~ D9047 will be active only when M9047="ON".	○	○	○
■ M9048	The annunciator monitoring has been enabled. When M9049="ON"and any coil of S900 ~ S999="ON", than M9048="ON".	○	○	
M9049	Enable annunciator monitoring. D9049 will be effective only when M9049="ON".	○	○	
Interrupt Prevented		M	VB	VH
M9050	Input interrupt I00□ is prevented.	○	○	○
M9051	Input interrupt I10□ is prevented.	○	○	○
M9052	Input interrupt I20□ is prevented.	○	○	○
M9053	Input interrupt I30□ is prevented.	○	○	○
M9054	Input interrupt I40□ is prevented.	○	○	○
M9055	Input interrupt I50□ is prevented.	○	○	○
M9056	Timer interrupt I6□□ is prevented.	○	○	○
M9057	Timer interrupt I7□□ is prevented.	○	○	○
M9058	Timer interrupt I8□□ is prevented.	○	○	○
M9059	High Speed Counter interrupt I010 ~ I060 is prevented.	○	○	○
Error Message		M	VB	VH
■ M9019	Real Time Clock setting error.	○	○	○
■ M9060	The M series I/O configuration error. When CPU detects a wrong I/O configuration, the PLC will stop, M9060= "ON" and the "ERR" LED of the CPU module will flash (1Hz).	○		
■ M9063	Wrong Parallel Link operation or wrong RS communication has been detected, M9063= "ON" but the PLC will keep running.	○	○	○
■ M9066	Program CHECK SUM error. PLC will stop, M9066= "ON" and the "ERR" LED of the CPU/Main module will flash (2Hz).	○	○	○
■ M9067	Operation error. If operation error occurs during program execution, then M9067="ON" but PLC will keep running.	○	○	○
M9068	Operation error latch. When M9068= "ON" and operation error occurs, the step number where operation errors occur will be latched in D9068.	○	○	○
Parallel Link Operation		M	VB	VH
■ M9070	When the Module is assigned as the Master station in a Parallel Link application, M9070= "ON".	○	○	
■ M9071	When the Module is assigned as the Slave station in a Parallel Link application, M9071= "ON".	○	○	
■ M9072	When the Parallel Link is operating, M9072= "ON".	○	○	
■ M9162	When the Parallel Link is operating in the High-speed transfer mode, M9162= "ON". This flag is based on the Master station's M9162 status.	○	○	
VB Series DIP Switch Status		M	VB	VH
■ M9080	The 2nd DIP switch status in the Main Unit.		○	
■ M9081	The 3rd DIP switch status in the Main Unit.		○	
■ M9082	The 4th DIP switch status in the Main Unit.		○	
VB Series Multi-Functional Display Setting Mode		M	VB	VH
M9084	Monitor function.		○	
M9085	Setting function.		○	
M9086	Progressive adding (+) function.		○	
M9087	Progressive subtracting (-) function.		○	
■ M9088	Error flag.		○	

Coil ID. No.	Instruction of Function	Series		
CP2 MODEM Dial-Up		M	VB	VH
M9100	CP2 Dial-Up start up flag.	○	○	
■ M9101	CP2 Dial-Up unsuccessful.	○	○	
RS Instruction		M	VB	VH
M9122	RS Data transmission flag.	○	○	○
M9123	RS Data receive completed.	○	○	○
■ M9124	M9124 will show up the "CD" signal when PLC's CP2 COM Port is connected with a MODEM.	○	○	
M9129	RS Data transmission Time-Out flag.	○	○	○
CPU LINK		M	VB	VH
■ M9183	CPU LINK communication unsuccessful (Master Station).	○	○	
■ M9184	CPU LINK communication unsuccessful (1st. Slave Station).	○	○	
■ M9185	CPU LINK communication unsuccessful (2nd. Slave Station).	○	○	
■ M9186	CPU LINK communication unsuccessful (3rd. Slave Station).	○	○	
■ M9187	CPU LINK communication unsuccessful (4th. Slave Station).	○	○	
■ M9188	CPU LINK communication unsuccessful (5th. Slave Station).	○	○	
■ M9189	CPU LINK communication unsuccessful (6th. Slave Station).	○	○	
■ M9190	CPU LINK communication unsuccessful (7th. Slave Station).	○	○	
The 32-bit Counter Count Direction Control		M	VB	VH
M9200 ~ M9234	When M92□□ = "OFF", the C2□□ is operated as a up counter. When M92□□ = "ON", the C2□□ is operated as a down counter.	○	○	
Controlling and Monitoring of High Speed Counter Count Direction		M	VB	VH
M9235 ~ M9245	When M92□□ = "OFF", the C2□□ is operated as a up counter. When M92□□ = "ON", the C2□□ is operated as a down counter.	○	○	○
■ M9246 ~ ■ M9255	When C2□□ is operated a up count, M92□□ = "OFF". When C2□□ is operated a down count, M92□□ = "ON".	○	○	○
The VB1 series position control instructions' relative flags (for VB1 series only)		M	VB	VH
M9140	If M9140="ON", the clear signal is sent to the servo when the return to zero point is complete.		○	
M9141	Interrupt signal logic reverse flag for Y0. For DVIT instruction only. OFF: normal logic (trigger by risen edge); ON: reverse logic (trigger by fallen edge).		○	
M9142	Interrupt signal logic reverse flag for Y1.		○	
M9143	Interrupt signal logic reverse flag for Y2.		○	
M9144	Interrupt signal logic reverse flag for Y3.		○	
M9145	Y0 pulse output stop immediately.		○	
M9146	Y1 pulse output stop immediately.		○	
M9147	Y2 pulse output stop immediately.		○	
M9148	Y3 pulse output stop immediately.		○	
■ M9149	Y0 pulse output monitor, "ON"=busy.		○	
■ M9150	Y1 pulse output monitor, "ON"=busy.		○	
■ M9151	Y2 pulse output monitor, "ON"=busy.		○	
■ M9152	Y3 pulse output monitor, "ON"=busy.		○	
The VB1 series hardware high speed counters' relative flags (for VB1 series only)		M	VB	VH
M9194	To activate the interrupt I050 for HHSC1. When (present value)=(setting value) of the HHSC1, no interrupt if M9194="OFF"; otherwise the interrupt routine will process immediately if M9194="ON".		○	
M9195	To activate the interrupt I060 for HHSC2. When (present value)=(setting value) of the HHSC2, no interrupt if M9195="OFF"; otherwise the interrupt routine will process immediately if M9195="ON".		○	
■ M9196	The counting direction of HHSC1, M9196="OFF"=counts up; M9196="ON"=counts down.		○	
■ M9197	The counting direction of HHSC2, M9197="OFF"=counts up; M9197="ON"=counts down.		○	

2-13-2 Instruction Table of Special Register

Register ID	Instruction of Function	Series		
PLC Operation Status		M	VB	VH
D9000	Time Setting of Watch Dog Timer. The WDT default value is 200ms and it can be changed by writing D9000, which is transferred from Program system when PLC power is "ON". (unit: 1ms)	○	○	○
■ D9004	Error coil ID number. When M9004= "ON", the content value may be 9060, 9063, 9066 or 9067 to indicate the error coil identification.	○	○	○
■ D9010	Current operation scan time (unit: 1ms)	○	○	○
■ D9011	Min. scan time (unit: 1ms)	○	○	○
■ D9012	Max. scan time (unit: 1ms)	○	○	○
System Status		M	VB	VH
■ D9001	Display the PLC's model and version. Model: M1 Series : 01 VB0 Series : 20 VB1 Series : 22 VB2 Series : 24 VH Series : 21 0 1 1 0 0 Version:V1.00	○	○	○
■ D9002	Capacity size of Memory. "16" indicates 16K Steps, "8" indicates 8K Steps, "2" indicates 2K Steps	○	○	○
■ D9003	Type of Memory. 00H indicates a built-in 8K Steps Flash Memory of PLC. 10H indicates an extend 8K Steps Flash Memory Card.	○	○	○
D9020	Input points (X0 ~ X7) filter response time setting. (unit: 1ms) The default value is 10ms and the available range for M and VB series is 0 ~ 60ms, VH series is 0 ~ 15ms.	○	○	○
D9039	Constant Scan Time duration. The default value is 0ms and it can be changed by setting D9039, which is transferred from Program system when PLC power is "ON". (Unit: 1ms)	○	○	○
Time Value of Real Time Clock		M	VB	VH
D9013	Seconds value. (0 ~ 59)	○	○	○
D9014	Minute value. (0 ~ 59)	○	○	○
D9015	Hour value. (0 ~ 23)	○	○	○
D9016	Day value. (1 ~ 31)	○	○	○
D9017	Month value. (1 ~ 12)	○	○	○
D9018	Year value: 1990 ~ 2089 (4 digits)	○	○	○
D9019	Weekday value: 0 (Sun.) ~ 6 (Sat.)	○	○	○
Step Ladder Instruction Correlated		M	VB	VH
■ D9040	1st (the lowest) active STL step When M9047= "ON", the step point ID numbers which are in action, they will be stored in D9040 ~ D9047. Where the D9040 will be stored the lowest ID number, the second lowest one will be stored in D9041 and so forth.	○	○	○
■ D9041	2nd active STL step	○	○	○
■ D9042	3rd active STL step	○	○	○
■ D9043	4th active STL step	○	○	○
■ D9044	5th active STL step	○	○	○
■ D9045	6th active STL step	○	○	○
■ D9046	7th active STL step	○	○	○
■ D9047	8th active STL step	○	○	○
■ D9049	When M9049= "ON", it stores the lowest currently active Annunciator in D9049.	○	○	
Error Message		M	VB	VH
■ D9063	Error code identifying Parallel link or RS communication error.	○	○	○
■ D9067	Error code identifying Operation error.	○	○	○
D9068	Latched the step address number of Operation error.	○	○	○
■ D9069	Step address number of Operation error.	○	○	○
Multi-Functional Display Functions		M	VB	VH
D9080	VB series: Multi-Functional Display mode setting. Please refer to Ch. 2-13-4 "VB Series Multi-Functional Display". VH series: Error Code Display function. Please refer to Ch. 2-13-5 "VH Series Error Code Display Function".		○	○
D9081	VB series Multi-Functional Display operation Auxiliary Register. Please refer to Ch. 2-13-4 "VB Series Multi-Functional Display".		○	
D9082			○	
D9083			○	
D9084			○	

Register ID	Instruction of Function		Series		
For VH-20AR analog I/O only			M	VB	VH
D9090	To organize the input modes of AIN1 ～ AIN4				○
D9091	Averaged input value from AIN1	• Data values refresh at every Scan Time. • The contain values of D9091 ~ D9094 are averaged of 8 sampling times.			○
D9092	Averaged input value from AIN2				○
D9093	Averaged input value from AIN3				○
D9094	Averaged input value from AIN4				○
D9095	To organize the output modes of AO1 and AO2				○
D9096	Digital value for AO1 output	• Analog outputs refresh at every Scan Time. • The digital value of analog outputs will be reset when the PLC "STOP"			○
D9097	Digital value for AO2 output				○
CP2 Communication Port			M	VB	VH
D9110 ~ D9113	Dial-Up number Registers. To store numbers for the MODEM to execute the Dial-Up function.		○	○	
D9121	The local station number for the CP2 to execute Computer Link or MODBUS communication.		○	○	○
■ D9122	The amount of residual data to be transferred by the instruction RS.		○	○	○
■ D9123	The amount of the data already received by the instruction RS.		○	○	○
D9124	To assign the Data Header code of instruction RS.		○	○	○
D9125	To assign the Data Terminator code of instruction RS.		○	○	○
D9129	To assign the data network "time-out" timer value of instruction RS or MBUS instruction.		○	○	○
High Speed Process Instruction			M	VB	VH
■ D9130	Used as a Counter to contain the number of the current record being processed in the HSZ comparison table.		○	○	
■ D9131	Used as a Counter to contain the number of the current record being processed in the HSZ comparison table when the PLAY operation has been enabled.		○	○	
■ D9132	Lower 16 bits	Used as a Register to contain the source (output pulse frequency) data for the PLSY instruction when used with the HSZ comparison table.	○	○	
■ D9133	Higher 16 bits				
■ D9134	Lower 16 bits	Used as a Register to contain a copy of value for current comparison when the HSZ comparison table and combined PLSY out put are used.	○	○	
■ D9135	Higher 16 bits				
■ D9136	Lower 16 bits	Used as a counter to contain the total number of pulses that have been output using the PLSY instruction. (NOT for VB1 series)	○	○	○
■ D9137	Higher 16 bits				
■ D9140	Lower 16 bits	Used as a counter to contain the total number of pulses that have been output to Y0 using the PLSY instruction.(NOT for VB1 series)	○	○	○
■ D9141	Higher 16 bits				
■ D9142	Lower 16 bits	Used as a counter to contain the total number of pulses that have been output to Y1 using the PLSY instruction.(NOT for VB1 series)	○	○	
■ D9143	Higher 16 bits				
CPU LINK			M	VB	VH
■ D9172	The time of communication "Time Out".		○	○	
■ D9177	The number of Slave Station in the network.		○	○	
■ D9178	The domain of components for transferred.		○	○	
■ D9179	The retry times for communication.		○	○	
■ D9201	Current network operation scan time.		○	○	
■ D9202	Max. network operation scan time.		○	○	
■ D9203	A counter to record the communication error occurred at the Master station.		○	○	
■ D9204	A counter to record the communication error occurred at the 1st Slave station.		○	○	
■ D9205	A counter to record the communication error occurred at the 2nd Slave station.		○	○	
■ D9206	A counter to record the communication error occurred at the 3rd Slave station.		○	○	
■ D9207	A counter to record the communication error occurred at the 4th Slave station.		○	○	
■ D9208	A counter to record the communication error occurred at the 5th Slave station.		○	○	
■ D9209	A counter to record the communication error occurred at the 6th Slave station.		○	○	
■ D9210	A counter to record the communication error occurred at the 7th Slave station.		○	○	
■ D9212	The communication error code of the 1st Slave station.		○	○	
■ D9213	The communication error code of the 2nd Slave station.		○	○	
■ D9214	The communication error code of the 3rd Slave station.		○	○	
■ D9215	The communication error code of the 4th Slave station.		○	○	
■ D9216	The communication error code of the 5th Slave station.		○	○	
■ D9217	The communication error code of the 6th Slave station.		○	○	
■ D9218	The communication error code of the 7th Slave station.		○	○	

Register ID	Instruction of Function		M	VB	VH
Index Register V, Z			M	VB	VH
D9180	Z0 Index Register		○	○	○
D9181	V0 Index Register		○	○	○
D9182	Z1 Index Register		○	○	○
D9183	V1 Index Register		○	○	○
D9184	Z2 Index Register		○	○	○
D9185	V2 Index Register		○	○	○
D9186	Z3 Index Register		○	○	○
D9187	V3 Index Register		○	○	○
D9188	Z4 Index Register		○	○	○
D9189	V4 Index Register		○	○	○
D9190	Z5 Index Register		○	○	○
D9191	V5 Index Register		○	○	○
D9192	Z6 Index Register		○	○	○
D9193	V6 Index Register		○	○	○
D9194	Z7 Index Register		○	○	○
D9195	V7 Index Register		○	○	○
The VB1 series position control instructions relative special registers (for VB1 series only)			M	VB	VH
D9140	Lower 16 bits	Current value registers of output pulse number (32-bit) from Y0		○	
D9141	Upper 16 bits				
D9142	Lower 16 bits	Current value registers of output pulse number (32-bit) from Y1		○	
D9143	Upper 16 bits				
D9144	Lower 16 bits	Current value registers of output pulse number (32-bit) from Y2		○	
D9145	Upper 16 bits				
D9146	Lower 16 bits	Current value registers of output pulse number (32-bit) from Y3		○	
D9147	Upper 16 bits				
D9148	To assign the input points of the interrupt signals of DVIT instruction. (the default value is H3210)			○	
D9149	Bias speed setting for the ZRN, DRVI , DRVA and DVIT instructions are operating. If the setting value > (D9151,D9150) / 10, then D9149 = (D9151,D9150) / 10			○	
D9150	Lower 16 bits	Maximum speed setting for the ZRN, DRVI, DRVA and DVIT instructions are operating, the default value = 200,000 Hz, the available range is 10 ~ 200,000 Hz. When the setting value exceeds acceptable value, it will equal to the largest acceptable value.		○	
D9151	Upper 16 bits				
D9152	Acceleration/Deceleration time setting for the ZRN, DRVI , DRVA and DVIT instructions are operating, the default value = 100 mS, the available range is 50 ~ 5,000 mS.			○	
The VB1 series hardware high speed counters' relative special registers (for VB1 series only)			M	VB	VH
D9224	The operating type of HHSC1. To input "0" = disable the function of HHSC1, "1" ~ "18" are 18 different counting modes.			○	
D9225	The operating type of HHSC2. To input "0" = disable the function of HHSC2, "1" ~ "18" are 18 different counting modes.			○	
D9226	Lower 16 bits	The present value of HHSC1.		○	
D9227	Upper 16 bits				
D9228	Lower 16 bits	The present value of HHSC2.		○	
D9229	Upper 16 bits				
D9230	Lower 16 bits	The setting value of HHSC1.		○	
D9231	Upper 16 bits				
D9232	Lower 16 bits	The setting value of HHSC2.		○	
D9233	Upper 16 bits				

2-13-3 Error Message/Code Description

Error Message

Coil ID. Number.	Title	The Time of Detecting Error Message	PLC Status	Status of the ERR LED
M9060	The M series I/O configuration error.	When Power is “OFF” → “ON” and “STOP” → “RUN”	STOP	Flash with 1Hz
M9063	Wrong Parallel Link operation or RS communication	When the paired stations signal is received	RUN	OFF
M9066	Check Sum Error	When Power is “OFF” → “ON” and “STOP” → “RUN”	STOP	Flash with 2 Hz
M9067	Operation Error	During the running program	RUN	OFF

Operation Error Code (the contains of D9067)

Error Code	Detail
0	No error message
6702	More than 5 Level of Call instruction have been nested together.
6703	More than 2 Level of Interrupt Insert have been nested together.
6704	More than 5 Level of FOR / NEXT have been nested together.
6705	An incompatible device has been specified as an operand for an applied instruction.
6706	An device has been specified exceed of the allowable range for an applied instruction operand.
6708	Error FROM / TO instruction

RS Communication Instruction Error Code (the contains of D9063)

Error Code	Detail
0	No error message.
6301	Parity，framing error.

CPU Link Communication Error Code (the contains of D9212 ~ D9218)

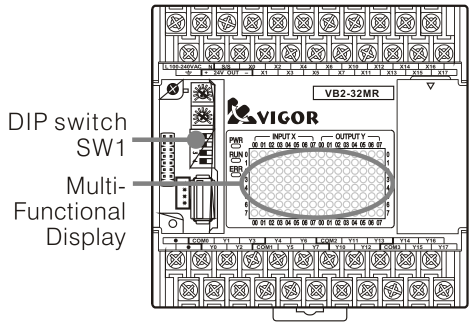
Error Code	Detail
00H	No error message.
01H	The communication has been Time Out.
05H	The communication has Check Sum Error.

MEMO

2-13-4 VB Series Multi-Functional Display

On the Main Unit of VB series PLC, it built-in a 16×8 points matrix LED Multi-Functional Display. When it conjugations with the user program can be used as a brief monitor of Human Machine Interface.

Inside of the left side cap, the second jumper of the DIP switch (SW1-2) isused to control the Multi-Functional Display. When the SW1-2= "OFF", the screen will display the I/O status; When the SW1-2= "ON", the screen will become the Multi-Functional Display.



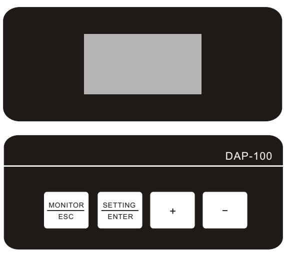
When the SW1-2="OFF", the screen will display the I/O status; When the SW1-2="ON", the screen will become the Multi-Functional Display.
By way of M9083 and SW1-3 , To select the indicate area : (when the SW1-2="OFF")

M9083="OFF"		M9083="ON"	
SW1-3="OFF" (VB0,VB1,VB2)	SW1-3="ON" (VB1,VB2)	SW1-3="OFF" (VB2)	SW1-3="ON" (VB2)
X0～X77 ; Y0～Y77	X100～X177 ; Y100～Y177	X200～X277 ; Y200～Y277	X300～X377 ; Y300～Y377

The Multi-Functional Display provides 8 mode types (Mode 0 ~ 7) and the operation setting is depend on the content of D9080. Changes the content of D9080 during the running program will change the display mode of the Multi-Functional Display.

Mode	D9080	D9081	Function	Content of the screen
Mode 0	K0	Disable	I/O status monitor	I/O points "ON"/ "OFF" status
Mode 1	K1	Indicator (K_n)	Value, word, chart display	The bit of D_n~D_{n+7} "ON"/ "OFF" status
Mode 2	K2	Indicator (K_n)	Error Code display	"E"+a 3-digit number of D_n
Mode 3	K3	Indicator (K_n)	A 4-digit number (0000 ~ 9999) display	A 4-digit number of D_n
Mode 4	K4	Indicator (K_n)	Two of 2-digit numbers (00 ~ 99) display	2-digit number of D_{n+1} & 2-digit number of Dn
Mode 5	K5	Indicator (K_n)	One word and a 3-digit number display	A word of D_{n+1} and a 3-digit number of Dn
Mode 6	K6	See the reference	The mode is for Data Access Panel	A word and a 3-digit number
Mode 7	K7	Indicator (K_n)	A 5-digit number (0 ~ 32,767) display	A 5-digit number of D_n

The Data Access Panel DAP-100 is a useful accessory, which is designed to join with the Multi-Functional Display together, become a simplified Human Machine Interface. They have the best economic effect because combine the Display and DAP-100.

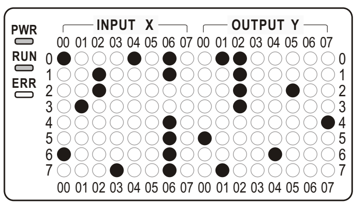


Item	Specification
Components	Display screen cover + 4 keys setting board
Material of surface	Smooth PC plastic, 0.254mm thickness
Key specification	12×12 TACT SWITCH
Key lifespan	500,000 Times
PLC interface	4 Input points of the PLC
Type of connect	Screw-Clamp Terminal
Facade size (W)×(H)	Both the display screen cover and 4 keys setting board are 110mm×45mm

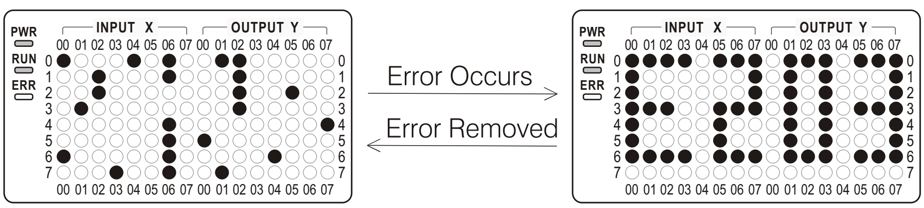
(1) Display Mode 0: I/O Status Monitor

This mode will post the I/O status at the screen. The function as same as when the SW1-2 put in "OFF" position.

D9080 = 0 (Display Mode 0)
The screen displays "ON"/ "OFF" status of I/O

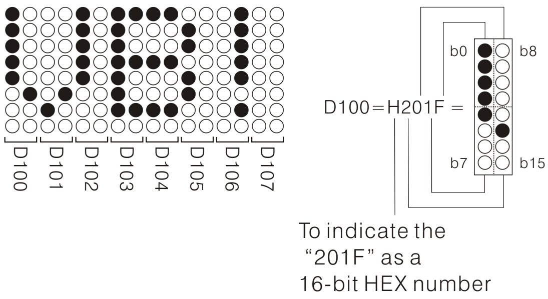


The main function of this mode is joined with other display mode to make the display screen more flexible. For example: Set the screen at mode 0, it will display the I/O status. But when the error occurs, than the screen will become the error code Display.



(2) Display Mode 1: Value, Numbers, Letters and Chart Display

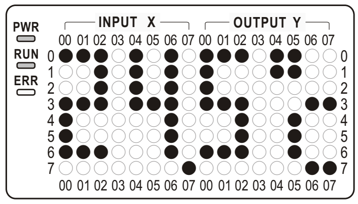
This mode is assigned the D9081 as a Indicator Register, and its content value (K*n*) will channeled the indicator to the Register D*n*. And the contents of D*n*~ D*n*+7 are 8 Registers total (16×8=128 bits), which will be used the bit type to display in the screen (128 points LED).



D9080 = K1 (Mode1)
D9081 = K100 (Display the contents of D100 ~ D107)
D100 = H201F
D101 = H2040
D102 = H001F
D103 = H497F
D104 = H4949
D105 = H0036
D106 = H5F00
D107 = H0000

- The example for display a temperature value:

  As the program chart below, the program will be used the content value number of D0 (pickup the last 3 digits and the unit is 0.1 ℃) to display in the left side of screen and in the right side of the screen will be showed the "℃" symbol.




```
M9002
--| |----[MOV K1 D9080]                 To assign the display at Mode 1.
         [MOV K2 D9081]                 Let the indicator channeled to D2.
M9002
--| |----[DMOV H007F417F D1000]         The chart for digit "0"
         [DMOV H00007F00 D1002]         The chart for digit "1"
         [DMOV H004F4979 D1004]         The chart for digit "2"
         [DMOV H007F4949 D1006]         The chart for digit "3"
         [DMOV H007F080F D1008]         The chart for digit "4"
         [DMOV H0079494F D1010]         The chart for digit "5"
         [DMOV H0079497F D1012]         The chart for digit "6"
         [DMOV H007F0101 D1014]         The chart for digit "7"
         [DMOV H007F497F D1016]         The chart for digit "8"
         [DMOV H007F494F D1018]         The chart for digit "9"
M9000
--| |----[SMOV D0 K3 K1 D10 K1]
         [MUL D10 K2 D12]
         [MOV D12 Z0]
         [DMOV D1000Z0 D2]
M9000
--| |----[SMOV D0 K2 K1 D10 K1]
         [MUL D10 K2 D12]
         [MOV D12 Z0]
         [DMOV D1000Z0 D4]
         [WOR D5 H8000 D5]              To crate the decimal point in the screen.
M9000
--| |----[SMOV D0 K1 K1 D10 K1]
         [MUL D10 K2 D12]
         [MOV D12 Z0]
         [DMOV D1000Z0 D6]
M9000
--| |----[DMOV H88887303 D8]            To crate the "℃" chart in the screen.
```

The Registers D1000 ~ D1018 are used to setup the dot charts of numbers 0 ~ 9. This block of program could replace by the function of "Tools \Edit Display Graph..." in the Ladder Master. When using the table, this block could skip.

SMOV D0 K3 K1 D10 K1: To pick up the value of 3rd (hundreds) position digit from D0.

MUL D10 K2 D12 / MOV D12 Z0 / DMOV D1000Z0 D2: Let the chart of 3rd digit value to display in the screen of hundreds digit place.

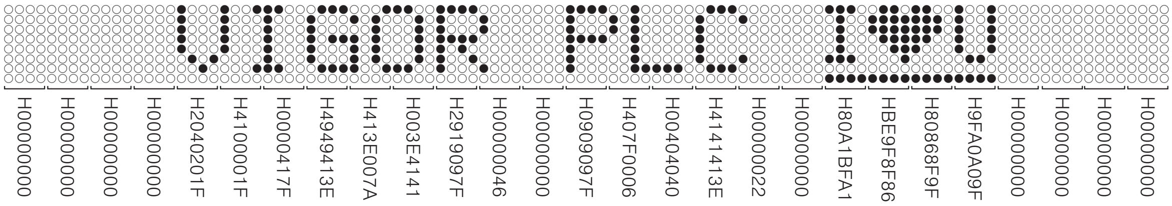
SMOV D0 K2 K1 D10 K1: To pick up the value of 2nd (tens) position digit from D0.

MUL D10 K2 D12 / MOV D12 Z0 / DMOV D1000Z0 D4: Let the chart of 2nd digit value to display in the screen of tens digit place.

SMOV D0 K1 K1 D10 K1: To pick up the value of 1st (units) position digit from D0.

MUL D10 K2 D12 / MOV D12 Z0 / DMOV D1000Z0 D6: Let the chart of "1st." digit value to display in the screen of units digit place.

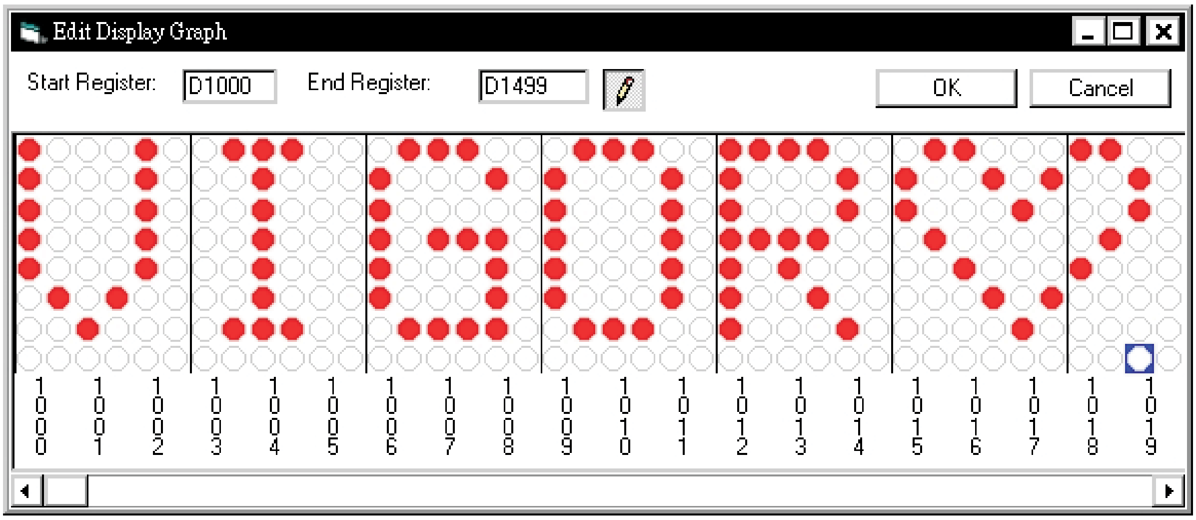
- Uses the "Rolling Chart " to display information
  ① Build the chart table, and then let the content of D9081 channeled to the beginning of the table.
  ② Use a given specific timing alternate (around 0.3 sec.) increased the content of D9081.
  ③ The chart table will be showed in the screen.
- The example for display a "Moving sign" :




```
M9002
--| |--+--[MOV  K1  D9080]         To assign the display at Mode 1.
       |
       +--[MOV  K1000  D9081]      Let the indicator channeled to the Register D1000.
M9002
--| |--+--[FMOV  H0  D1000  K8]
       +--[DMOV  H2040201F  D1008]
       +--[DMOV  H4100001F  D1010]
       +--[DMOV  H0000417F  D1012]
       +--[DMOV  H4949413E  D1014]
       +--[DMOV  H413E007A  D1016]
       +--[DMOV  H003E4141  D1018]
       +--[DMOV  H2919097F  D1020]
       +--[DMOV  H00000046  D1022]
       +--[DMOV  H00000000  D1024]
       +--[DMOV  H0909097F  D1026]
       +--[DMOV  H407F0006  D1028]
       +--[DMOV  H00404040  D1030]
       +--[DMOV  H4141413E  D1032]
       +--[DMOV  H00000022  D1034]
       +--[DMOV  H00000000  D1036]
       +--[DMOV  H80A1BFA1  D1038]
       +--[DMOV  HBE9F8F86  D1040]
       +--[DMOV  H80868F9F  D1042]
       +--[DMOV  H9FA0A09F  D1044]
       +--[FMOV  H0  D1046  K8]
T200               K30
--|/|-----(T200)                    To set up the moving speed of "Rolling Chart".
T200
--| |--+--[INCP  D9081]            The incrementally of  the Pointer value.
       |
       +--[CMP  D9081  K1046  M0]  Check up to reply if it runs to the end of the table.
M1
--| |-----[MOVP  K1000  D9081]     When it displays to the end of the table, the Pointer will be channeled
                                   to the beginning of the table, repeat displays the table again and again.
```

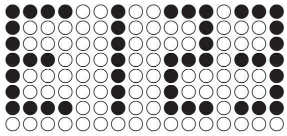
To build the dot graph table which is shows above (start-up from D1000). This block of program could replace by the function of "Tools \Edit Display Graph..." In the Ladder Master. When using the table, this block could skip.

- The programming tool software iLadder Masteriprovide the tool: “Tools \Edit Display Graph...”, that is for to create the display chart easily. It can be edit the graph of letters, numbers and symbols from keyboard directly. Also, it is possible to use cursor to create an individual graph.
- This edit function will create and store data into the corresponding File Register, and the it is a part of user program. So, cleverly to use the Edit Display Graph function could save the the user program size for create the graph and it is easy to maintain.
  For more detail about the File Register, please reference the section “2-9 File Register (D)”.



(3) Display Mode 2: Error Code Display

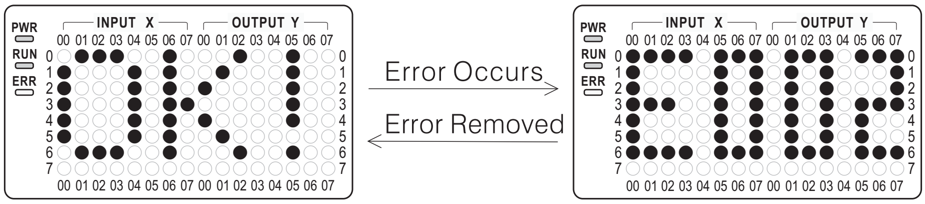
This mode is assigned the D9081 as a Indicator Register, and its content value (K*n*) will channeled the indicator to the Register D*n*. The last 3 digits number of content in D*n* will be displayed in right side of the screen and the left side of the screen will display an "E" symbol to indicate it displaying an error code.

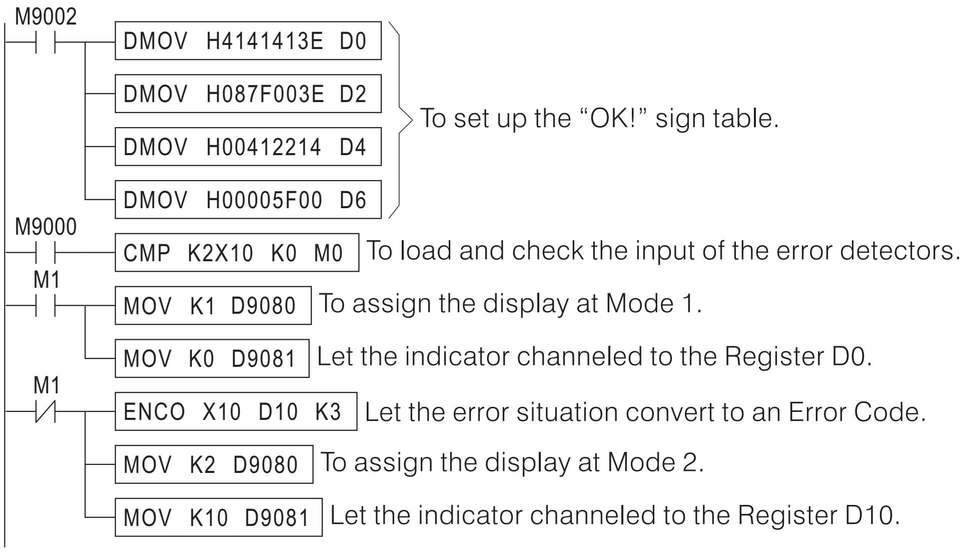


D9080 = K2 (Mode 2)
D9081 = K100 (To display the last 3 digits number of content in D100)
D100 = K123

- The example for display an "Error Code" :

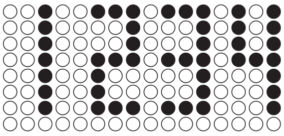
We assume the PLC input points X10 ~ X17 connect with 8 error detectors (ex. Motor over load, Over the limits...) When the error occurs, it will be showed the corresponding error code in the screen; Otherwise it will be showed an "OK !" sign in the the screen.





(4) Display Mode 3: To display a 4-digit number(0000 ~ 9999)

This mode is assigned the D9081 as a Indicator Register, and its content value (K*n*) will channeled the indicator to the Register D*n*. The last 4 digits number of content in D*n* will be displayed in the screen.



D9080 = K3 (Mode 3)
D9081 = K100 (To display the last 4 digits number of content in D100)
D100 = K1234

- Example 1:

We assume the PLC input points X0 ~ X7 connect with 8 switch contacts. When the the contact of X0= "ON", it will display the content value of D0 in the screen; When the the contact of X1= "ON", it will display the content value of D1 in the screen, and so forth.

Contact	Instruction	Description
M9002 / M1	MOV K3 D9080	To assign the display at Mode 3.
	MOV K0 D9081	Let the indicator channeled to the Register D0.
M9000	CMP K2X0 K0 M0	To load and check the input switches status.
M1 (NC)	ENCO X0 D10 K3	To encode the switch status.
	MOV D10 D9081	Let the indicator channeled to the Register, which is the switch appointed.

- Example 2:

We use the program to display the contents of D0~D7 in the screen and also the appointed output points Y0 ~ Y7 will be turned "ON" as indicant. Use the Analog Potentiometer VR1 to give a value for display timing alternate.

Contact	Instruction	Description
M9000	VRRD K1 D8	To read the value of VR1 and put the value to Register D8
M9002	MOV K3 D9080	To assign the display at Mode 3.
M9000	MOV C0 D9081	Use the current value of C0 as a Pointer.
T0 (NC)	T0 D8	Use the value of VR1 to assign the setting value of Timer T0.
T0	C0 K8	Use the timing alternate of T0 to increase the value of C0.
C0	RST C0	When finished the D0 ~ D7 display sequence, to reset the C0.
M9000	DECO C0 Y0 K3	When displaying the sequence of D0 ~ D7, to make the appointed outputs points Y0 ~ Y7 turned "ON"

- Example 3:

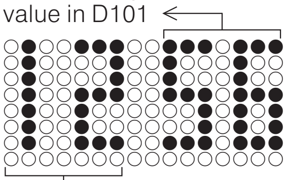
We use the program to display the value of Analog Potentiometer VR1 in the screen, and assigned the value as the setting value of T0.
Usually, using the value of VR1 and VR2 to setting the Timers, only depend on intuition without real measure. Since the VB series provided the Multi-Functional Display, to adjust the Analog Potentiometer VR1 and VR2 become clear and definite.
This program example is to make a description of VR1 and VR2 combine with Multi-Functional Display.

Contact	Instruction	Description
M9002	MOV K3 D9080	To assign the display at Mode 3.
	MOV K0 D9081	Let the target channeled to the Register D0.
M9000	VRRD K1 D0	To read the value of VR1 and put it into D0.
T0 (NC)	T0 D0	Use the content value of D0 to set up the setting value of Timer T0.
T0	ALTP Y0	Use the timing alternate of T0 to drive the output point Y0 flashed "ON" / "OFF".

(5) Display Mode 4: To display two 2-digit numbers (00 ~ 99)

This mode is assigned the D9081 as a Indicator Register, and its content value (K*n*) will channeled the indicator to the Register D*n*. The last 2 digits number of content in D*n* will be displayed in left side of the screen and the right side of the screen will be displayed the last 2 digits number of content in D_{n+1}.

the last 2 digits of content value in D101

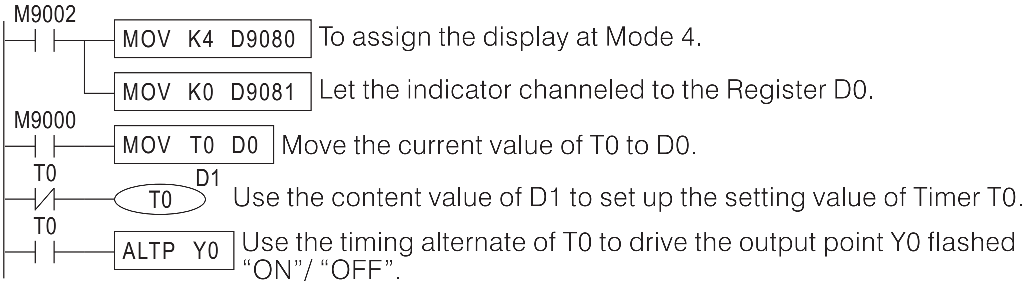


the last 2 digits of content value in D100

D9080 = K4 (Mode 4)
D9081 = K100 (To display the last 2 digits of content value in D100 and D101)
D100 = K56
D101 = K12

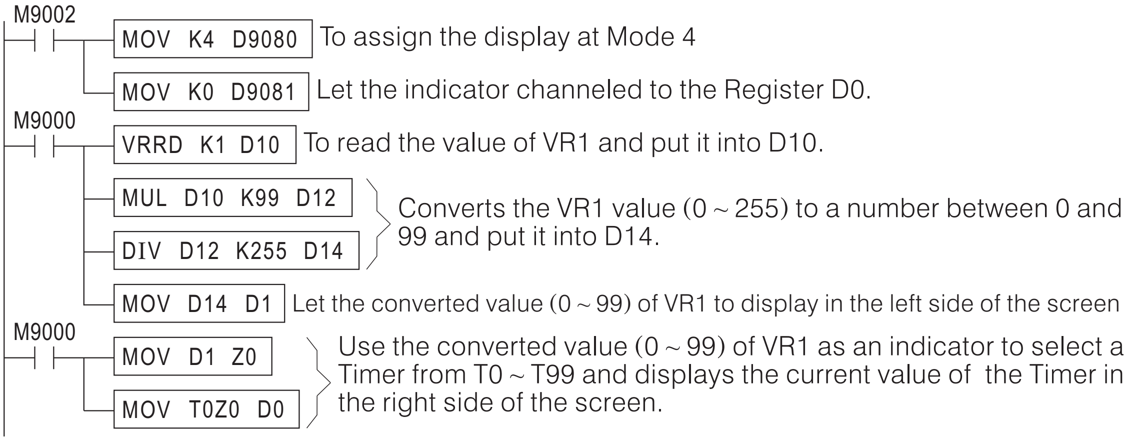
- Example 1:

This program will display the setting value of T0 in the left side of screen and put the current value in the right side.



- Example 2:

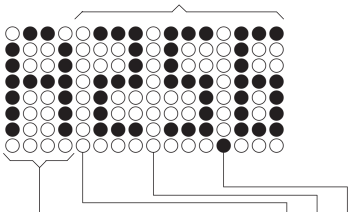
This program will select a current value from T0 ~ T99 and display the value in the screen. Use the value of VR1 to pick up a corresponding Timer from T0 ~ T99 and displays the ID number of the Timer has been selected in the left side of screen, displays the current value of the Timer in the right side.



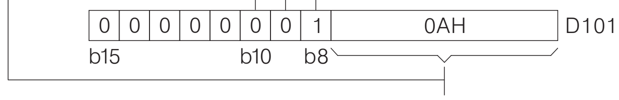
(6) Display Mode 5: To display a letter and a 3-digit number

This mode is assigned the D9081 as a Indicator Register, and its content value (K*n*) will channeled the indicator to the Register D*n*. The last 3 digits number of content in D*n* will be displayed in right side of the screen and a letter is specified by b7 ~ b0 of D*n*+1, on the left side of the screen. The location of decimal point shows is specified by b10 ~ b8 of D*n*+1. Please refer to the following example for details.

The content value in D100



D9080 = K5 (Mode 5)
D9081 = K100 To display the last 3 digits number of content in D100.
To display the letter, which is specified by the contents of b7 ~ b0 in D101.
To display the decimal point, which is specified by the contents of b10 ~ b8 in D101.
D100 = K256
D101 = H10A



The convert table between the number code and the letter to display

Number code	Display letter	Number code	Display letter	Number code	Display letter	Number code	Display letter
00H	0	10H	G	20H	W	30H	m
01H	1	11H	H	21H	X	31H	n
02H	2	12H	I	22H	Y	32H	o
03H	3	13H	J	23H	Z	33H	p
04H	4	14H	K	24H	a	34H	q
05H	5	15H	L	25H	b	35H	r
06H	6	16H	M	26H	c	36H	s
07H	7	17H	N	27H	d	37H	t
08H	8	18H	O	28H	E	38H	u
09H	9	19H	P	29H	f	39H	v
0AH	A	1AH	Q	2AH	g	3AH	w
0BH	B	1BH	R	2BH	h	3BH	x
0CH	C	1CH	S	2CH	i	3CH	y
0DH	D	1DH	T	2DH	j	3DH	z
0EH	E	1EH	U	2EH	k		
0FH	F	1FH	V	2FH	l		

- This mode can be applied to a multi-data display, where the data title is shown on the left side and the data content is shown on the right.

- Example :

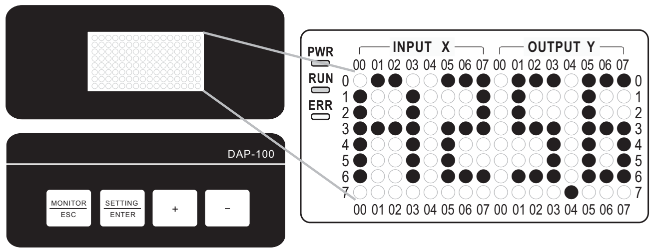
  This program will select a content value from D0 ~ D9 and display the value in the screen. Let the title of D0 ~ D9 are A ~ J. And use the value of VR1 to pick up a corresponding Register from D0 ~ D9 and displays the title of the Register has been selected in the left side of screen, displays the current value of the Register in the right side.

```
M9002
─┤├──┬── MOV K5 D9080        To assign the display at Mode 5.
     └── MOV K10 D9081       Let the indicator channeled to the Register D10.
M9000
─┤├──┬── VRRD K1 D12         To read the value of VR1 and put it into D12.
     ├── MUL D12 K9 D14      } Converts the VR1 value (0 ~ 255) to a number between 0 and
     ├── DIV D14 K255 D16    } 9 and put it into D16.
     ├── ADD D16 H0A D18     Shifts the content value of D16 (add "0AH") and put it into D18.
     └── MOV D18 D11         To display the title of selected Register in the left sid of screen.
M9000
─┤├──┬── MOV D16 Z0          } Use the converted value (0 ~ 9) of VR1 as an indicator to select a
     └── MOV D0Z0 D10        } Register from D0 ~ D9 and displays the constant value of the
                               Register in the right side of the screen.
```

(7) Display Mode 6: Data Programmer Mode

This mode is design to collocate with a Data Access Panel (DAP-100) for setting and watching the argument and data in the program (the contents of Data Registers).

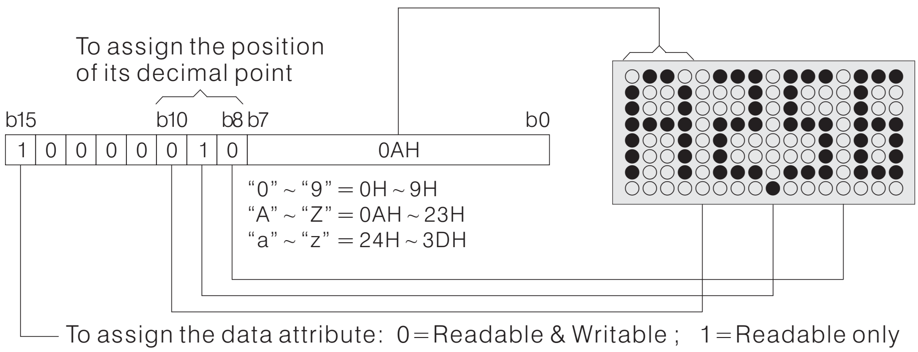
This mode shows the same screen as displayed in Mode 5. Read the instructions on Mode 5 before reading the instructions in this section. This mode can set multiple sets of data with 4 push-button switches.



The Special Register and the Special Coil used in this mode are explained as below:

<1> D9080: To indicate the Mode (D9080 = K6)

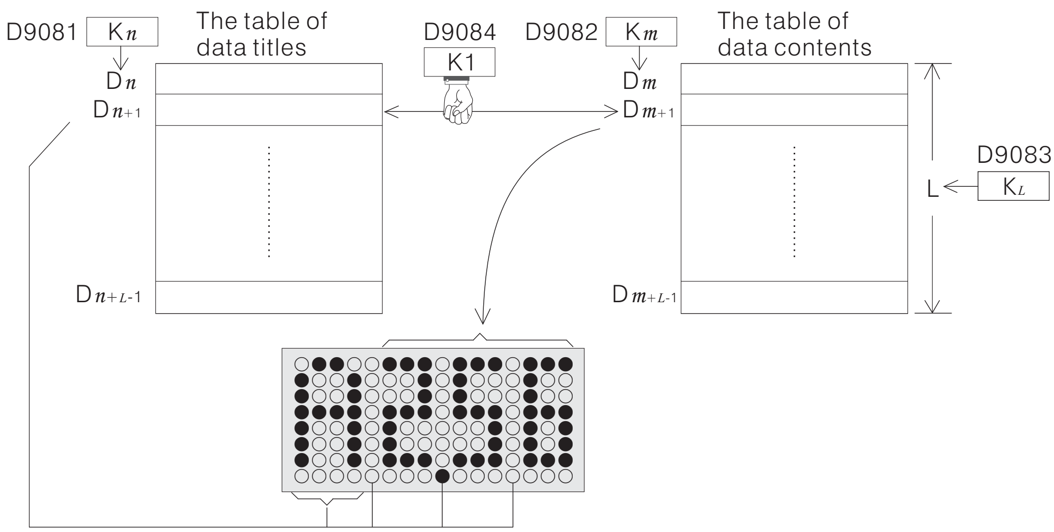
<2> D9081: The Indicator Register for the table of data titles. Its content value (Kn) will channeled the indicator to the Register Dn, where the Dn is the beginning Register for the table of data titles, and the table-length is decided by D9083(L). Each Register in this table can assign a data title, the position of its decimal point, and the data attribute (R/RW).



<3> D9082: The Indicator Register for the table of data titles. Its content value (Km) will channeled the indicator to the Register Dm, where the Dm is the beginning Register for the table of data titles, and the table-length is decided by D9083(L). Each Register in this table can store a 3-digit number (0 ~ 999).

<4> D9083: Use the Register to assign the table-length. Its content value (KL) designates the table-length (the table of data titles and the table of data contents).

<5> D9084: Use the Register as a task indicator. Its content value Kp (=K0 ~ K[L-1]) will channeled the indicator to the table of data titles and the table of data contents, and displays the constant value of the corresponding table in the screen.



<6> The numbers monitoring/programming functions of the Data Programmer Mode are Performed with 5 Special Coils (such 5 Special Coils only perform the corresponding functions in this mode). This mode is available to use external input signals to drive the corresponding Special Coils, and it fulfills the practical application from simple external operation.

M9084: Monitoring function. When this contact turns “ON”, the screen shows the table Contents, which is directed by D9084.

M9085: Setting function. When the contact turns “ON”, the data setting function is Accessed.

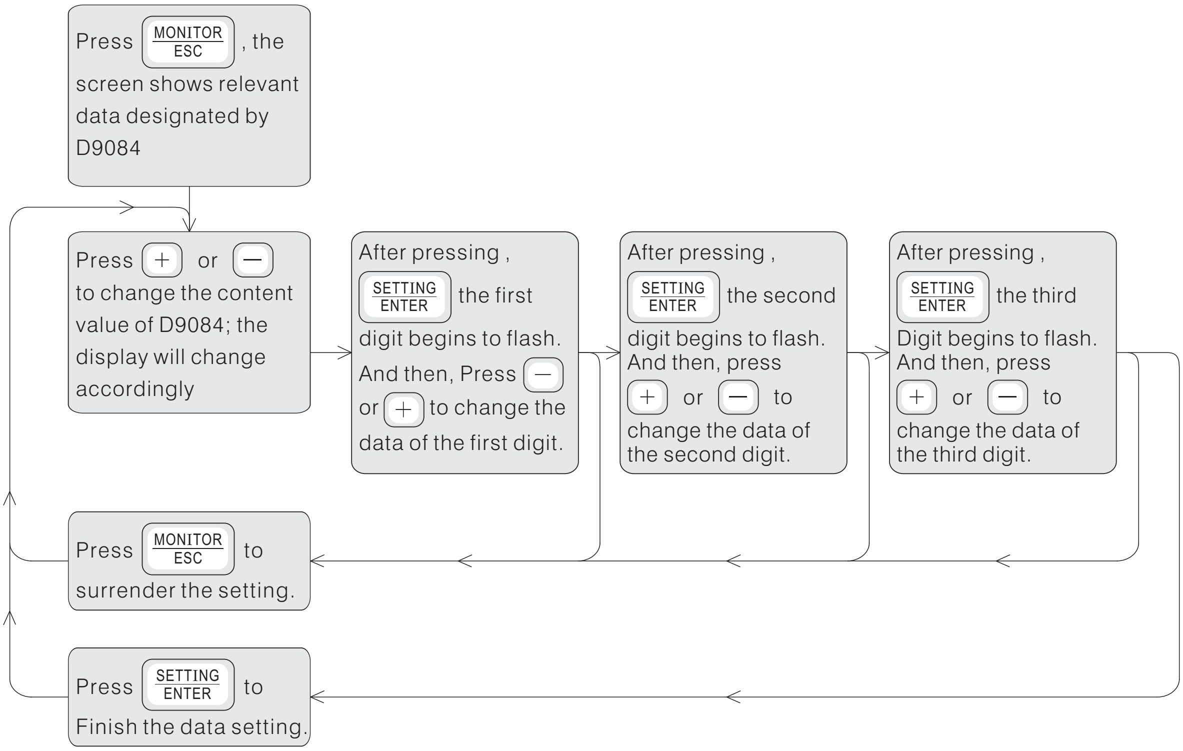
M9086: Increasing function (+).

M9087: Decreasing function (-).

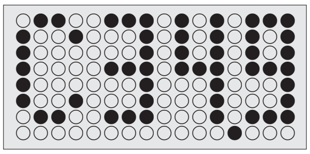
M9088: The error signals output. When the data attribute is set to be readable only, and the setting or writing function is to be performed, then M9088 will become “ON” for a scan time.

Assume the Special Coils (M9084 ~ M9087) are driven by the external push-button switches.

The operation process of the mode is shown as follows:



- Example :



D9080 = K6 (Mode 6)

D9081 = K7000 The beginning Register for the table of data titles is D7000 (latched).

D9082 = K7010 The beginning Register for the table of data contents is D7010 (latched).

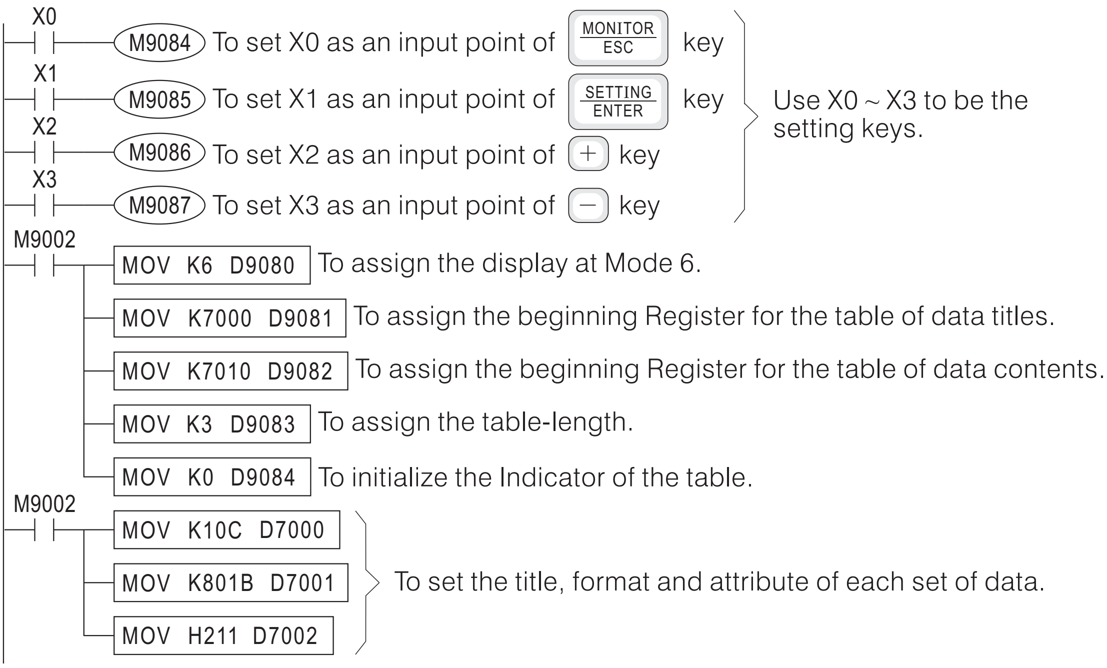
D9083 = K3 The table-length is "3", indicating there are 3 sets of data.

D9084 = K0 As the content value of the current task indicatorIs "0", The first set of data in the table will be displayed in the screen.

D9084: K0

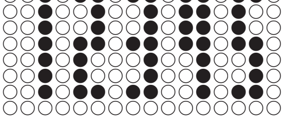
Data Title	Displayed Value	Data Attribute		The table of data titles		Table of data contents
C	34.9	Readable &Writable	D7000	10CH	D7010	349
R	128	Readable only	D7001	801BH	D7011	128
H	1.00	Readable &Writable	D7002	211H	D7012	100

Stepladder Chart program



(8) Display Mode 7: To display a 5-digit number (0 ~ 32767)

This mode is assigned the D9081 as a Indicator Register, and its content value (Kn) will channeled the indicator to the Register Dn. The content in Dn will be displayed in the screen.



D9080 = K7 (Mode 6)

D9081 = K100 (To display the content of D100)

D100 = K12345

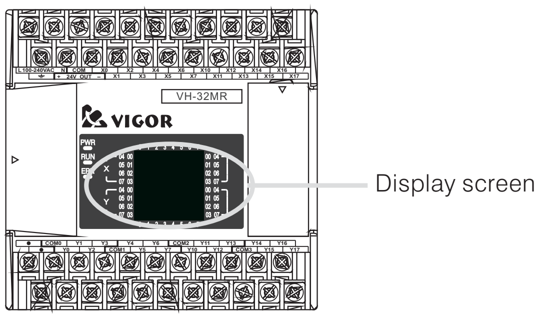
- The function and operation of this mode are as same as mode 3, please refer to mode 3 for the examples.

MEMO

2-13-5 VH Series Error Code Display Function

The VH series PLC Main Unit (exclude VH-10MR and VH-14MR) built-in an 8×8 points matrix LED screen, which is not only displaying the I/O status, also has 109 error codes (01 ~ 99 and E0 ~ E9) display function.

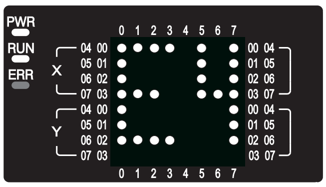
The error code display function helps to display the condition of machine error, and then increases the maintenance effect. It is a very useful and economical function.

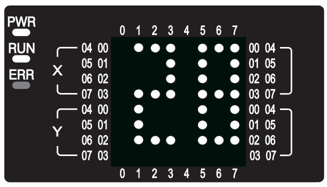


The VH series PLC using D9080 Special Register to control the display function.

Contents of D9080	The contents in the screen
0	To display the I/O status*
1 ~ 99	To display the number 01 ~ 99
100 ~ 109	To display the error code E0 ~ E9

* When SW1-2 = "OFF", it will indicate the status of X0 ~ X37 and Y0 ~ Y37; When SW1-2 = "ON", it will indicate the status of X40 ~ X77 and Y40 ~ Y77





- Example :

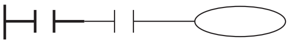
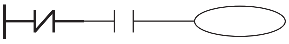
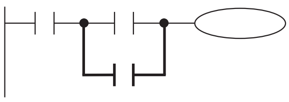
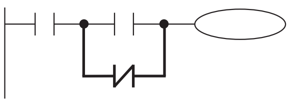
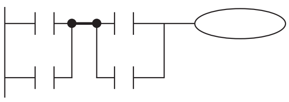
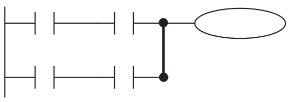
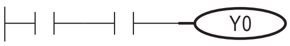
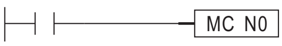
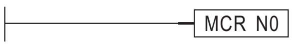
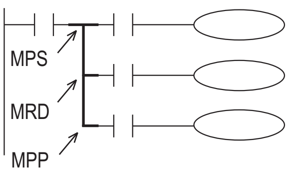
  We assume the PLC input points X0 ~ X7 connect with 8 error sensors (ex. Motor overload, out the limitations,.....) When error occurs, the screen will display the corresponded error code (E0 ~ E7). Otherwise, if there is no error has been detected, the screen will display the PLC's I/O status.

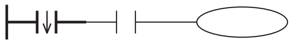
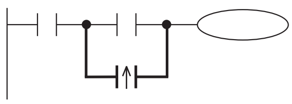
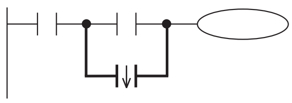
```
 M9000
 ─┤ ├───[CMP  K2X0  K0  M0]     To detect errors.
 M1
 ─┤ ├───[MOV  K0  D9080]        No error has been detected, then K0 → D9080, the screen will display the
                                PLC's I/O status.
 M1
 ─┤/├─┬─[ENCO  X0  D0  K3]      When error has been detected, then the status of X0 ~ X7 will encoded
      │                         into D0.
      ├─[ADD  D0  K100  D3]     Shifts (add K100) the content value of D0, let D0=K100 ~ K107.
      │
      └─[MOV  D0  D9080]        To display the corresponded error code (E0 ~ E7) in the screen.
```

MEMO

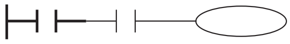
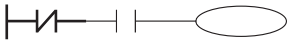
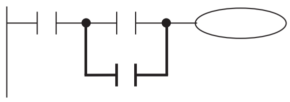
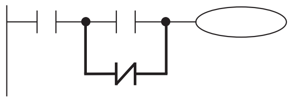
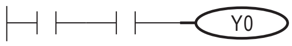
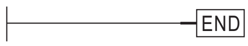
3 Basic Instructions

3-1 Basic Instruction Table

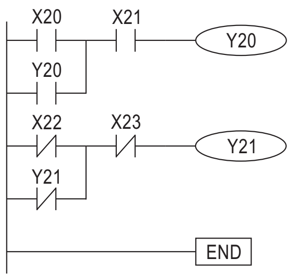
Mnemonic	Format	Devices	Function
LD (LOAD)		X, Y, M, S, T, C	Initial logical operation contact type NO (Normally Open)
LDI (LOAD INVERSE)		X, Y, M, S, T, C	Initial logical operation contact type NC (Normally Closed)
AND (AND)		X, Y, M, S, T, C	Serial connection of NO (Normally Open) contacts
ANI (AND INVERSE)		X, Y, M, S, T, C	Serial connection of NC (Normally Closed) contacts
OR (OR)		X, Y, M, S, T, C	Parallel connection of NO (Normally Open) contacts
ORI (OR INVERSE)		X, Y, M, S, T, C	Parallel connection of NC (Normally Closed) contacts
ANB (AND BLOCK)		—	Serial connection of multiple parallel circuits
ORB (OR BLOCK)		—	Parallel connection of multiple contact circuits
OUT (OUT)		X, Y, M, S, T, C	Final logical operation type coil drive
PLS (PULSE)		Y, M (excluding special M coil)	Rising edge pulse
PLF (PULSE FALLING)		Y, M (excluding special M coil)	Falling edge pulse
SET (SET)		Y, M, S	Sets component permanently "ON"
RST (RESET)		Y, M, S, T, C, D	Resets component permanently "OFF"
MC (MASTER CONTROL)		N0 ~ N7	Denotes the start of a master control block
MCR (MC RESET)		N0 ~ N7	Denotes the end of a master control block
MPS (POINT STORE)		—	Stores the current result of the internal PLC operations
MRD (POINT READ)		—	Reads the current result of the internal PLC operations
MPP (POINT POP)		—	Pops (recalls and removes) the currently stored result
END (END)		—	Force the current program scan to end
NOP (NO OPERATION)	—	—	No operation or null step

Mnemonic	Format	Devices	Function
LDP (LOAD PULSE)		X, Y, M, S, T, C	Initial logical operation Rising edge pulse
LDF (LOAD FALLING PULSE)		X, Y, M, S, T, C	Initial logical operation Falling edge pulse
ANDP (AND PULSE)		X, Y, M, S, T, C	Serial connection of Rising edge pulse
ANDF (AND FALLING PULSE)		X, Y, M, S, T, C	Serial connection of Falling edge pulse
ORP (OR PULSE)		X, Y, M, S, T, C	Parallel connection of Rising edge pulse
ORF (OR FALLING PULSE)		X, Y, M, S, T, C	Parallel connection of Falling edge pulse
INV (INVERSE)		—	Invert the current result of the internal PLC operations

3-2 LD,LDI,AND,ANI,OR,ORI,OUT and END

Mnemonic	Format	Devices	Function
LD (LOAD)		X, Y, M, S, T, C	Initial logical operation contact type NO (Normally Open)
LDI (LOAD INVERSE)		X, Y, M, S, T, C	Initial logical operation contact type NC (Normally Closed)
AND (AND)		X, Y, M, S, T, C	Serial connection of NO (Normally Open) contacts
ANI (AND INVERSE)		X, Y, M, S, T, C	Serial connection of NC (Normally Closed) contacts
OR (OR)		X, Y, M, S, T, C	Parallel connection of NO (Normally Open) contacts
ORI (OR INVERSE)		X, Y, M, S, T, C	Parallel connection of NC (Normally Closed) contacts
OUT (OUT)		Y, M, S, T, C	Final logical operation type coil drive
END (END)		—	Force the current program scan to end

Ladder Chart Format

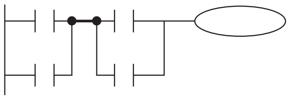
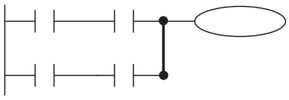


Instructions Format

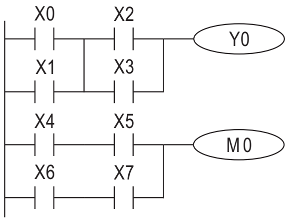
LD	X20	Initial logical operation contact type NO (Normally Open)
OR	Y20	Parallel connection of NO (Normally Open) contacts
AND	X21	Serial connection of NO (Normally Open) contacts
OUT	Y20	Final logical operation type coil drive
LDI	X22	Initial logical operation contact type NC (Normally Closed)
ORI	Y21	Parallel connection of NC (Normally Closed) contacts
ANI	X23	Serial connection of NC (Normally Closed) contacts
OUT	Y21	Final logical operation type coil drive
END		Force the current program scan to end

- The OUT T and OUT C Instructions will be specified in Section 3-8.
- When the PLC executes the END instruction, it forces that program to end the current scan and carry out the updating processes for both inputs and outputs. All instructions in the program after the END instruction will not be executed.
- The END instruction can be inserted into the middle of the program, it helps program debugging as the section after the END instruction is disabled and isolated from the area that is being checked.

3-3 Instruction ANB and ORB

Mnemonic	Format	Devices	Function
ANB (AND BLOCK)		_	Serial connection of multiple parallel circuits
ORB (OR BLOCK)		_	Parallel connection of multiple contact circuits

Ladder Chart Format

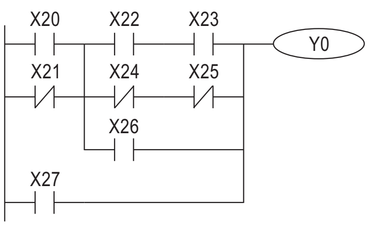


Instructions Format

LD	X0	Initial logical operation contact type NO (Normally Open)
OR	X1	Parallel connection of NO (Normally Open) contacts
LD	X2	Initial (the starting point of the circuit block) logical operation contact type NO (Normally Open)
OR	X3	Parallel connection of NO (Normally Open) contacts
ANB		Serial connection of multiple parallel circuits
OUT	Y0	Final logical operation type coil drive
LD	X4	Initial logical operation contact type NO (Normally Open)
AND	X5	Serial connection of NO (Normally Open) contacts
LD	X6	Initial (the starting point of the circuit block) logical operation contact type NO (Normally Open)
AND	X7	Serial connection of NO (Normally Open) contacts
ORB		Parallel connection of multiple contact circuits
OUT	M0	Final logical operation type coil drive

- To declare the starting points of the circuit block, please use an LD or LDI instruction. After completing the serial circuit block, connect it to the preceding block in series/parallel using the ANB/ORB instruction.
- When using ANB/ORB instructions in a batch, use no more than 8 LD and LDI instructions in the definition of the program blocks (to be connected in serial/parallel). Ignoring this will result in a program error.
- Please refer to the following program example, it is used both the ANB and ORB instructions in a circuit block.

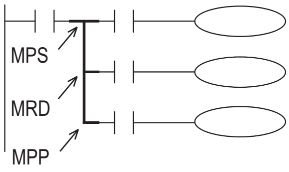
Ladder Chart Format



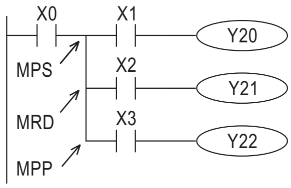
Instructions Format

```
LD    X20
ORI   X21
LD    X22
AND   X23
LDI   X24
ANI   X25
ORB
OR    X26
ANB
OR    X27
OUT   Y0
```

3-4 Instruction MPS, MRD and MPP

Mnemonic	Format	Devices	Function
MPS (POINT STORE)		—	Stores the current result of the internal PLC operations
MRD (POINT READ)		—	Reads the current result of the internal PLC operations
MPP (POINT POP)		—	Pops (recalls and removes) the currently stored result

Ladder Chart Format

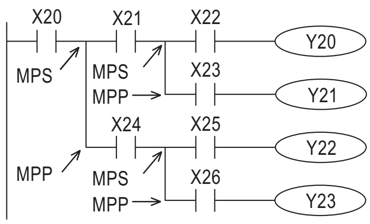


Instructions Format

LD	X0	Initial logical operation contact type NO (Normally Open)
MPS		Stores the current result of the internal PLC operations
AND	X1	Serial connection of NO (Normally Open) contacts
OUT	Y20	Final logical operation type coil drive
MRD		Reads the current result of the internal PLC operations
AND	X2	Serial connection of NO (Normally Open) contacts
OUT	Y21	Final logical operation type coil drive
MPP		Pops (recalls and removes) the currently stored result
AND	X3	Serial connection of NO (Normally Open) contacts
OUT	Y22	Final logical operation type coil drive

- The MPS instruction stores the connection point of the ladder circuit so that further coil branched can recall the value later.
- The MRD instruction recalls or reads the previously stored connection point data and forces the next contact to connect to it.
- The MPP instruction pops (recalls and removes) the stored connection point data of the last array and removes the connection point from the result. The last contact or coil circuit must connect to an MPP instruction.
- In any continuous connection circuit block, the difference between the number of the active MPS instruction and the number of the active MPP instruction shall be no greater than 11; When all connection circuit blocks are ended, the total number of the MPS instruction and the total number of the MPP instruction have been used in the program must be the same (there must has a MPP instruction corresponding to every signal MPS instruction).
- A Multiple-connection program example:

Ladder Chart Format



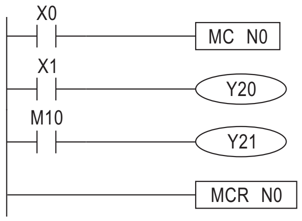
Instructions Format

```
LD    X20          OUT   Y22
MPS                MPP
AND   X21          AND   X26
MPS                OUT   Y23
AND   X22
OUT   Y20
MPP
AND   X23
OUT   Y21
MPP
AND   X24
MPS
AND   X25
```

3-5 Instruction MC and MCR

Mnemonic	Format	Devices	Function
MC (MASTER CONTROL)	MC N0	N0 ~ N7	Denotes the start of a master control block
MCR (MC RESET)	MCR N0	N0 ~ N7	Denotes the end of a master control block

Ladder Chart Format

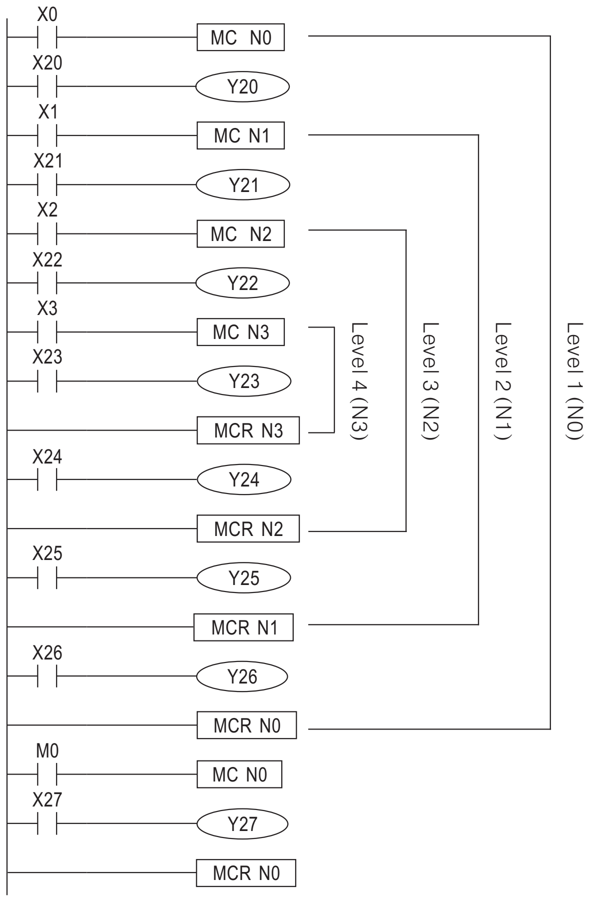


Instructions Format

LD	X0	X0 is a conditional contact
MC	N0	Become a master control block which is controlled by X0.
LD	X1	
OUT	Y20	If X0="ON" then: Status of Y20 = Status of X1; Status of Y21 = Status of M10
LD	M10	
OUT	Y21	If X0="OFF" then: Y20="OFF"; Y21="OFF"
MCR	N0	

- When input point X0 (conditional contact) is "ON", all instructions between the MC and MCR instructions will be executed.
- When input point X0 (conditional contact) is "OFF", all instructions between the MC and MCR instructions will NOT be executed. All Timers and the coils which are driven by the OUT instruction, will be turned "OFF"; while the status of Retentive Timers, Counters and the coils driven by the SET / RST instruction will be kept.
- Use an MC instruction to shift the bus line (LD, LDI points) to a point after the conditional contact and use an MCR instruction to return to the original bus line.
- A master control block allows contains another master control blocks inside, which makes a nest level. This structure at the most can use 8 level (N0 ~ N7). The top nest level shall be N0, and then, N1, N2…, and the deepest level shall be N7.
- A multiple-level program example:

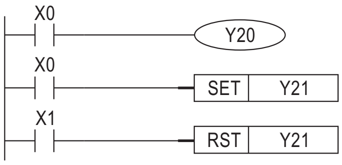
Ladder Chart Format



3-6 Instruction SET and RST

Mnemonic	Format	Devices	Function
SET (SET)	SET Y0	Y, M, S	Sets a bit device permanently “ON”
RST (RESET)	RST Y0	Y, M, S, T, C, D	Resets a bit device permanently “OFF”

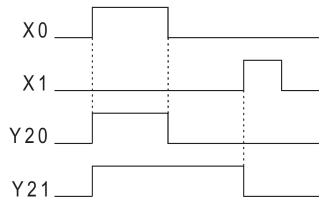
Ladder Chart Format



Instructions Format

```
LD    X0
OUT   Y20
LD    X0
SET   Y21
LD    X1
RST   Y21
```

Active I/O Duration Time Sheet

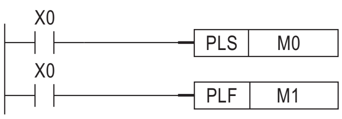


- The SET instruction sets the output coil permanently “ON” when it has been operated.
- The RESET instruction resets the output coil permanently “OFF” or resets the current value of a Timer, Counter or Register to zero.
- The SET instruction and the RESET instruction can be used for the same output coil as many times as necessary.
- The RST C instruction will be specified in Section 3-8.

3-7 Instruction PLS and PLF

Mnemonic	Format	Devices	Function
PLS (PULSE)	PLS Y0	Y, M (excluding special M coil)	Rising edge pulse
PLF (PULSE FALLING)	PLF Y0	Y, M (excluding special M coil)	Falling edge pulse

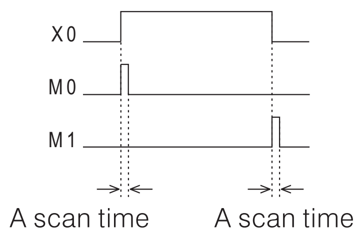
Ladder Chart Format



Instructions Format

```
LD    X0
PLS   M0
LD    X0
PLF   M1
```

Active I/O Duration Time Sheet



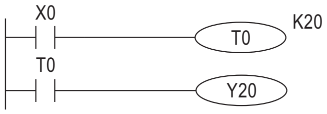
- When X0=“OFF”→ “ON”, M0 will output a pulse for a scan time.
- When X0=“ON”→ “OFF”, M0 will output a pulse for a scan time.

3-8 Instruction OUT and RST for a Timer or Counter

If the OUT instruction is used for the component T or C, it must input a setting value.

Timer

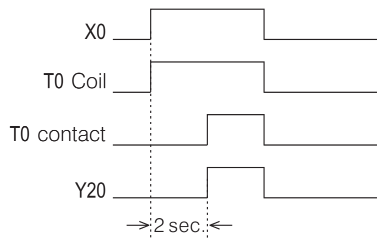
Ladder Chart Format



Instructions Format

```
LD    X0
OUT   T0
      K20
LD    T0
OUT   Y20
```

Active I/O Duration Time Sheet



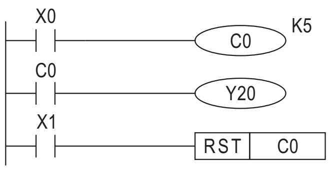
- The setting value of a Timer can be set either use a K (Constant) or Data Register D (Parameter).
- The Operative Range of the setting value:

Timer ID No.	Timing Unit	Type of the Timer	The Operative Range	Real Setting Time
T0 ~ T199 (T0 ~ T62)	100mS	General Timer	1 ~ 32,767 (If the setting value beyond the range, it will default to 1)	0.1 ~ 3276.7 sec.
T200 ~ T245 (T32 ~ T62)	10mS			0.01 ~ 327.67 sec.
(T63)	1mS			0.001 ~ 32.767 sec.
T246 ~ T249	1mS	Retentive Timer		0.001 ~ 32.767 sec.
T250 ~ T255	100mS			0.1 ~ 3276.7 sec.

- The Timer ID No. in the midst of square brackets () are for the VH series.
- To reset the Current values of Retentive Timer T246 ~ T255 must using the RST instruction.

Counter

Ladder Chart Format



Instructions Format

```
LD    X0
OUT   C0
      K5
LD    C0
OUT   Y20
LD    X1
RST   C0
```

The Action Exposition

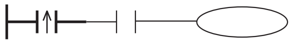
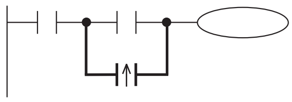
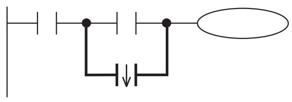
- When X0= "OFF" → "ON", C0 will executes up count once, until the Current Value of C0=5 and then the output contact of C0= "ON", where the Current Value will not be increased anymore and the contact will stay permanently "ON".
- When X1="ON", the Current Value of C0 will be reset to "0" and the contact of C0 will become "OFF".

- The setting value of a Counter can be stet either use a K (Constant) or Data Register D (Parameter).
- The Operative Range of the setting value:

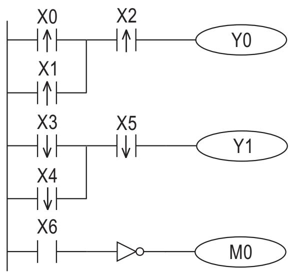
Counter ID No.	Type of the Counter		The Operative Range
C0 ~ C99 (C0 ~ C15)	General	16 bits, Up Counter	1 ~ 32,767 (If the setting value beyond the range, it will default to 1)
C100 ~ C199(C16 ~ C31)	Latched		
C200 ~ C219	General	32 bits, Up/Down Counter	− 2 , 147 , 483 , 648 ~ 2 , 147 , 483 , 647
C220 ~ C234	Latched		
C235 ~ C255 (C235 ~ C254)	High Speed Counter (Latched)		

- The Counter ID No. in the midst of square brackets () are for the VH series.
- When using High Speed Counters, please refer to the section 2-7 "High Speed Counter".

3-9 Instruction LDP, LDF, ANDP, ANDF,ORP, OPF and INV

Mnemonic	Format	Devices	Function
LDP (LOAD PULSE)		X, Y, M, S, T, C	Initial logical operation Rising edge pulse
LDF (LOAD FALLING PULSE)		X, Y, M, S, T, C	Initial logical operation Falling edge pulse
ANDP (AND PULSE)		X, Y, M, S, T, C	Serial connection of Rising edge pulse
ANDF (AND FALLING PULSE)		X, Y, M, S, T, C	Serial connection of Falling edge pulse
ORP (OR PULSE)		X, Y, M, S, T, C	Parallel connection of Rising edge pulse
ORF (OR FALLING PULSE)		X, Y, M, S, T, C	Parallel connection of Falling edge pulse
INV (INVERSE)		—	Invert the current result of the internal PLC operations

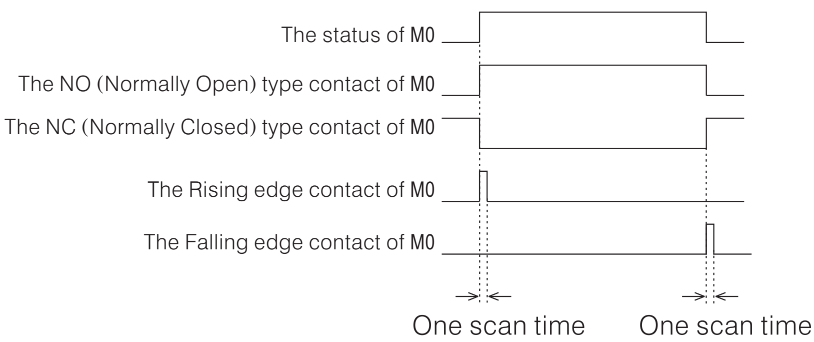
Ladder Chart Format



Instructions Format

LDP	X0	Initial logical operation Rising edge pulse
ORP	X1	Parallel connection of Rising edge pules
ANDP	X2	Serial connection of Rising edge pulse
OUT	Y0	Final logical operation type coil drive
LDF	X3	Initial logical operation Falling edge pulse
ORF	X4	Parallel connection of Falling edge pules
ANDF	X5	Serial connection of Falling edge pulse
OUT	Y1	Final logical operation type coil drive
LD	X6	Initial logical operation contact type NO (Normally Open)
INV		Invert the current result of the internal PLC operations
OUT	M0	Final logical operation type coil drive

- The Rising edge contact will be active for one program Scan Time after the associated device status changes from “OFF” to “ON”.
- The Falling edge contact will be active for one program Scan Time after the associated device status changes from “ON” to “OFF”.

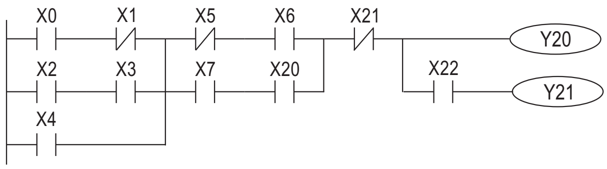


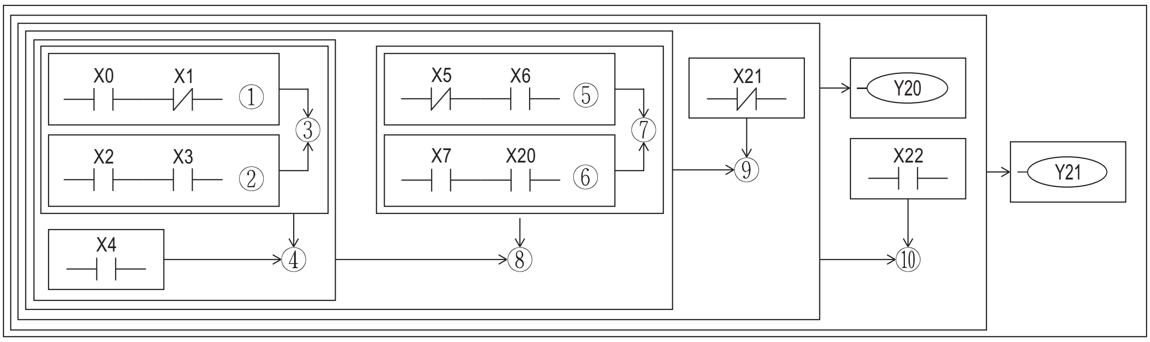
- The output contact status of the Rising or Falling edge ON/OFF is produced by OUT, SET, RST, PLS and PLF instructions; BUT, if the status of a bit component is changed by an instruction, its Rising or Falling edge contact WILL NOT get a output.
  For example, to operate the instruction `CMP D0 D1 M0` may change the statuses of M0 ~ M2, but the statuses change will not make the Rising or Falling edge contact outputs at the moment. If use the Rising or Falling edge contact of M0, M1 or M2 in the program, it may cause a wrong response.

3-10 Significant Notes For Programming

3-10-1 The Ladder Chart Format Converts To The Instruction Format

The rule to convert a program from Ladder Chart to Instruction Format is follower the order: from left to right and from top to bottom.



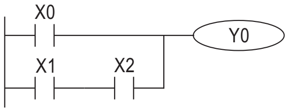


LD	X0	A Serial connection Block ①
ANI	X1	
LD	X2	A Serial connection Block ②
AND	X3	
ORB		Parallel Connection Block Circuits
OR	X4	Parallel Connection of NO Contact
LDI	X5	A Serial connection Block ⑤
AND	X6	
LD	X7	A Serial connection Block ⑥
AND	X20	
ORB		Parallel Connection Block Circuits
ANB		Serial Connection of Multiple Parallel Circuits
ANI	X21	Serial Connection of NC Contact
OUT	Y20	Output
AND	X22	Serial Connection of NO Contact
OUT	Y21	Output

3-10-2 Programming Technics

1. If the program used Parallel Connection Block Circuits, then put a bigger serial connection block on the upper place, which will be simpler and easier for the programming.

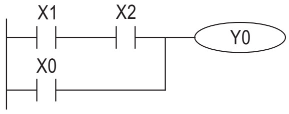
Ladder Chart Format



Instructions Format

LD	X0
LD	X1
AND	X2
ORB	
OUT	Y0

Ladder Chart Format

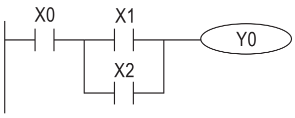


Instructions Format

LD	X1
AND	X2
OR	X0
OUT	Y0

2. It's recommended to place a circuit with more parallel link contacts on the left side.

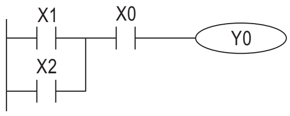
Ladder Chart Format



Instructions Format

LD	X0
LD	X1
OR	X2
ANB	
OUT	Y0

Ladder Chart Format

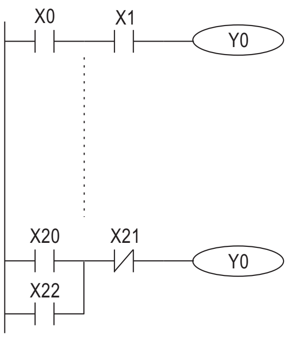


Instructions Format

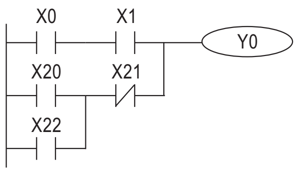
LD	X1
OR	X2
AND	X0
OUT	Y0

3. Reuse a output coil or Double Coiling is not a incorrect syntax. But the coil operation designated last is the effect coil. Hence, conditional signal contacts should be revised, and use of the output coil of the same ID number should be avoided.

Ladder Chart Format

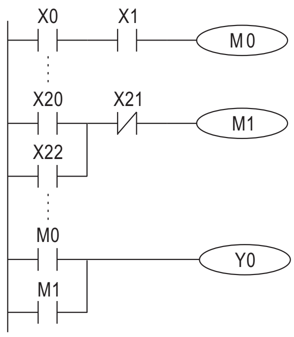


Ladder Chart Format



or

Ladder Chart Format



MEMO

4. Sequential Function Chart (SFC) and Step Ladder Chart

4-1 Introduction The Sequential Function Chart (SFC)

4-1-1 Basic Structure of a SFC

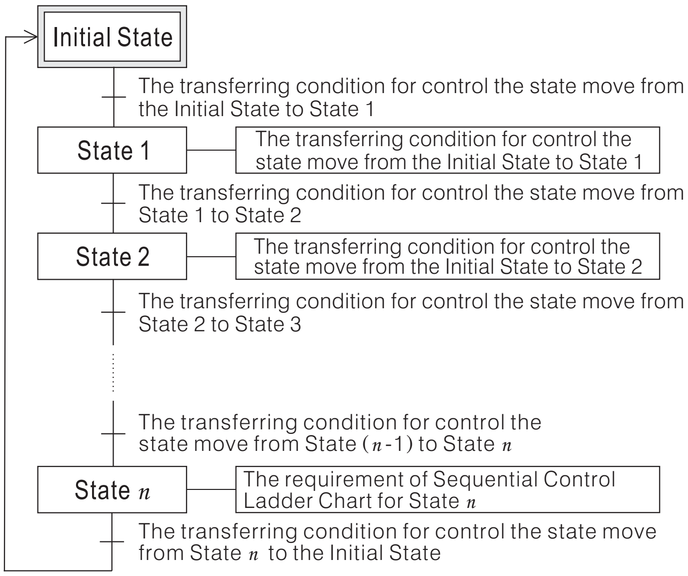
In the universe of Automatic Control, the Electro-Control system should work closely with machine movements to get the result of the Automatic Control, i.e. the synergistic integration technology of Mechatronics, which has become popular in recent years. However, it's quite a difficult job to learn such a complicated sequential control design for machinery engineers, therefor the SFC (Sequential Function Chart) is developed accordingly.

The SFC is designed for create a easily way to understand about the moves of a machine, also it has the following features:

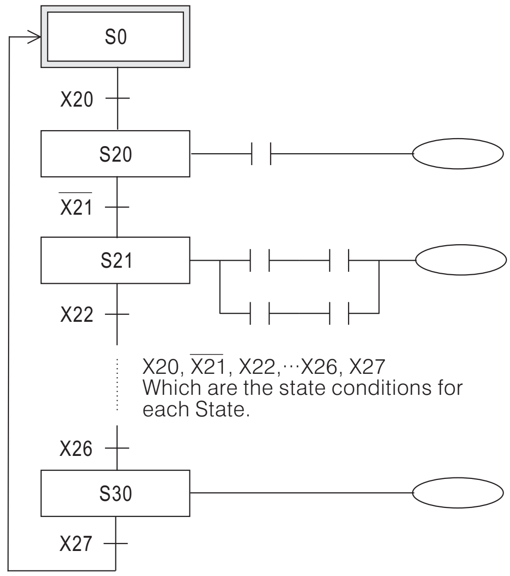
(1) It is not necessary to design the special sequence for constantly state changing of stepladders, the PLC will execute internal links and double coils under different state. Simple sequence design for every state will prompt normal machine works.

(2) Even a person is not the machine designer, who can easily lean all actions and conduct trials, adjustment, error detection and maintenance.

The Flow Chart Diagram of SFC



The Actual SFC



The left diagram is a Flow Chart of SFC and the right diagram is the actual SFC corresponding to the left one. The PLC will execute to start from the Initial State, then complete State 1 → State 2 →... → State *n* in sequence based on State conditions and achieve a cycle of control.

4-1-2 Basic components of SFC

1. States

(1) Initial State

The first state to execute after PLC runs. Ordinarily the Initial State is achieved by using the startup initial pulse. The Initial State is represented by a frame with double sidelines.

(2) Effective State

The Effective States refer to the execution state of PLC. Under an effective state, PLC will execute the following actions in sequence:

① Driving the coil of the output point, timer or counter relative to the state.

② Resetting the last pasted action, i.e. turning the actions which are relative to the last state into "OFF".

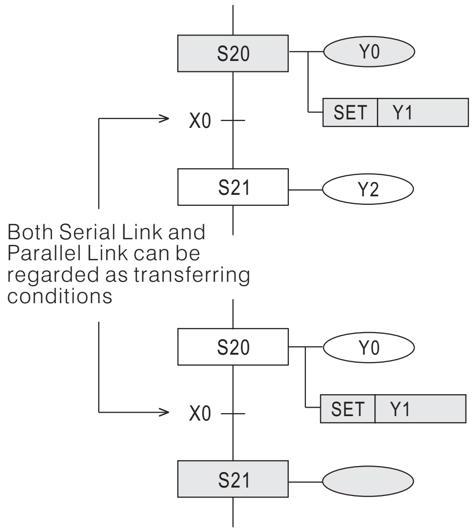
③ Transferring the machine action to the next state when the transferring condition is authorized. In generally there is a connecting line to connect the states, and it indicates the direction of the signal.

2. Transferring Condition

There is a line segment connecting the states, and on the line put a perpendicular short line which is used to express the related conditions driving the states transferred.

4-1-3 State and Action of SFC

Under an effective state, if the action of SFC uses a output coil, the difference between using the instructions OUT and SET to drive the output coil will be:



When S20= "ON", Y0 and Y1 are also "ON"

When X0 (the transferring condition)= "OFF" → "ON", the effective state will be transferred from S20 to S21 (When S21= "ON", then S20= "OFF").

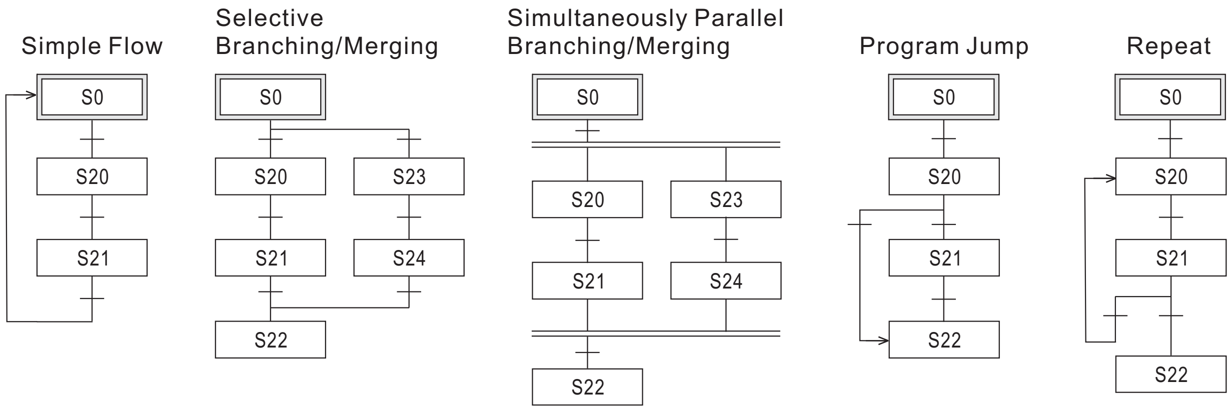
When S21= "ON", Y2 will be "ON" and Y0 becomes "OFF". But, because Y1 is driven by the SET instruction, so Y1 still keeps "ON".

Attention!

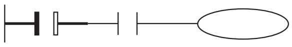
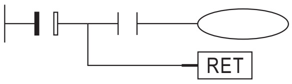
When the effective state transferring from S20 to S21, there will be one scan time both status of S20 and S21 are "ON".

4-1-4 Types of SFC

According to flow control methods, SFC has 5 basic types:



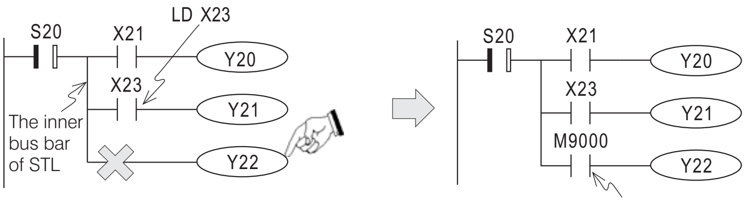
4-2 Step Ladder Instruction

Mnemonic	Format	Devices	Function
STL		S	STep Ladder starts
RET		—	RETurning to standard ladder, Step Ladder ends.

A step point is composed of an STL instruction and a device S. An STL instruction occurring in the program refers that the program has already entered into the STL state controlled by Step Flows. The RET instruction indicates the end of the Step Ladder Chart. Subsequently the initial logical operation is reset to an ordinary SLC state. An SFC completed should be converted into a Step Ladder Chart, and the following importances should be noted during the conversion:

(1) Output Driving Method

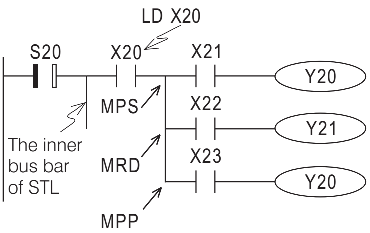
As in the left diagram referred below. If inside the Step point has an LD or LDI instruction, a output coil can not directly connected with inner bus bar of the STL.



Must insert an always “ON”, “a” contact

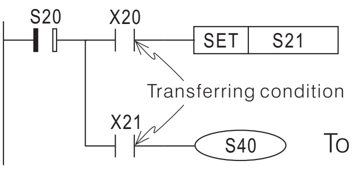
(2) Location of Instruction MPS, MRD and MPP

The MPS, MRD and MPP instructions can not be directly used for Step point’s inner bus bar, unless an LD or LDI instruction has been used previously.



(3) Transferring Method of Step Point

As in the diagram referred below, these two instructions, SET S21 and OUT S40 are the instructions driving another Step point, and when the command is transferred to another Step point, the previous Step point itself will be reset to “OFF” automatically. The difference is that the SET instruction is used to drive an immediately following STL step point, but the OUT instruction is used for loops and jumps to drive a Step point which is not immediately following.



To drive an immediately following STL step, which typically will have a larger STL state number than the current step.

To drive a Step point which is not immediately following.

(4) Function of Instruction RET

Since the RET instruction represents the end of a step, the RET instruction will appear eventually after a series of Step points. A program may be written many Steps, each Step should put an instruction RET in the end. The instruction RET can be used as many times as required.

(5) Applicable Basic Instructions for Step Ladder Chart

Basic instructions can be used between two of STL instructions or used between STL instruction and RET instruction.

Operational State		LD, LDI, AND, ANI, OR, ORI, SET, RST, PLS, PLF, OUT, NOP, LDP, LDF, ANDP, ANDF, ORP, ORF, INV	ANB, ORB MPS, MRD, MPP	MC, MCR
Initial State General State		✓	✓	×
Branching State Merging State	Output Processing	✓	✓	×
	Transfer Process	✓	×	×

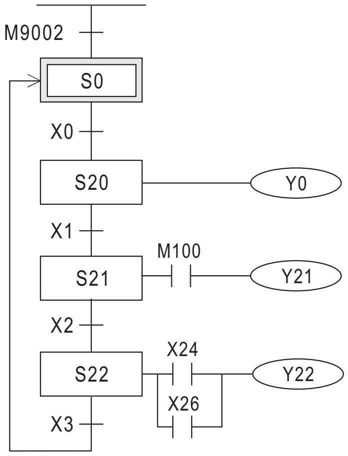
- STL instructions are prohibited in subprograms.
- Instruction CJ is not prohibited in Step Ladder Chart but it makes the program more complicated, so it’s recommended that do not use the CJ instruction in Step Ladder Chart.

4-3 Relation between SFC and Step Ladder Chart

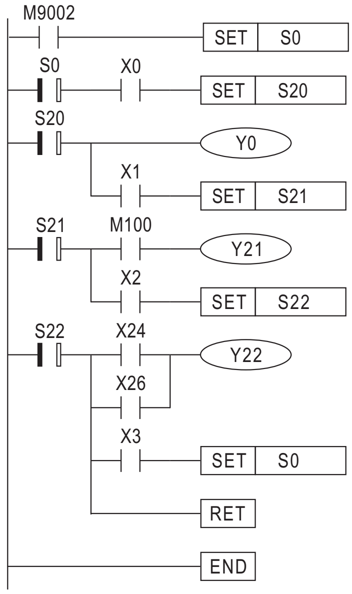
4-3-1 Simple-flow SFC and Step Ladder Chart

Simple Flow: A flow without branching and merging

(a) SFC



(b) Step Ladder Chart Format



(c) Instruction Format

```
LD    M9002
SET   S0
STL   S0
LD    X0
SET   S20
STL   S20
OUT   Y0
LD    X1
SET   S21
STL   S21
LD    M100
OUT   Y21
LD    X2
SET   S22
STL   S22
LD    X24
OR    X26
OUT   Y22
LD    X3
SET   S0
RET
END
```

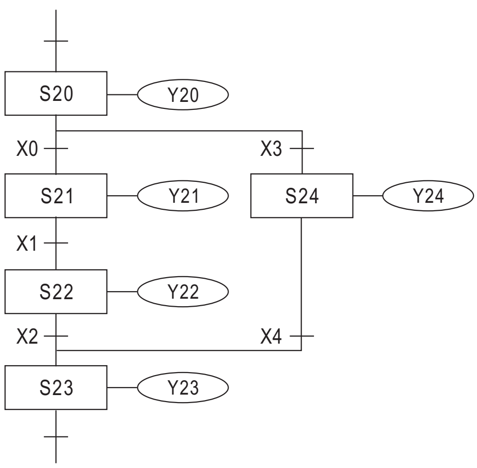
In Diagram (a) of SFC, each state provides three functions: driving processing for loading, assigning transferred devices and transferring conditions. Such SFC, in the format of Step Ladder Chart, is displayed as in Diagram (b), in which we adopt ─┫▯─ as the symbol for use of STL instructions, and these instructions are provided with transferring and auto reset functions.

4-3-2 Selective Branching / Merging SFC and Step Ladder Chart

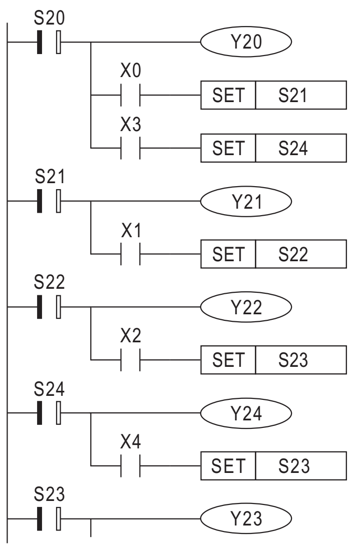
Selective Branching: To select one of the branching flow for state transferring.

Selective Merging: To join branching flows into a simple flow.

(a) SFC



(b) Step Ladder Chart Format



(c) Instruction Format

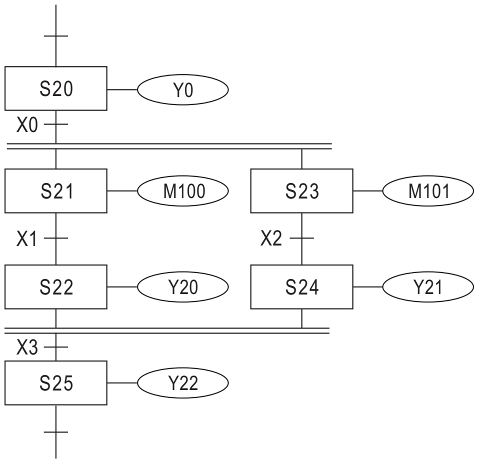
```
STL   S20
OUT   Y20
LD    X0
SET   S21
LD    X3
SET   S24
STL   S21
OUT   Y21
LD    X1
SET   S22
STL   S22
OUT   Y22
LD    X2
SET   S23
STL   S24
OUT   Y24
LD    X4
SET   S23
STL   S23
OUT   Y23
```

4-3-3 Simultaneously Parallel Branching / Merging SFC and Step Ladder Chart

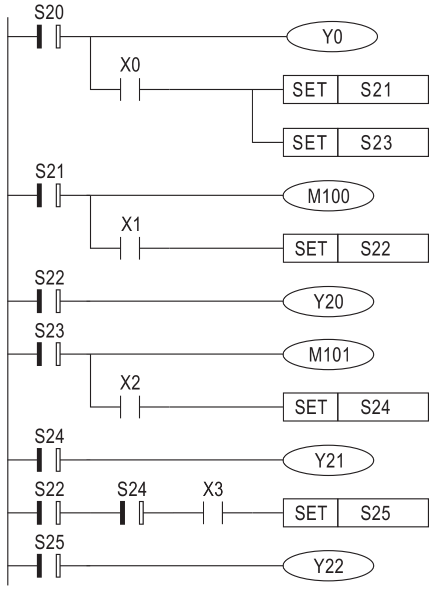
Simultaneously Parallel Branching: The first State of each branching flow becomes effective when the transferring condition is authorized.

Simultaneously Parallel Merging: To transfer the effective state to the next state when the last state of each branching state becomes effective and the transferring condition is authorized.

(a) SFC



(b) Step Ladder Chart Format



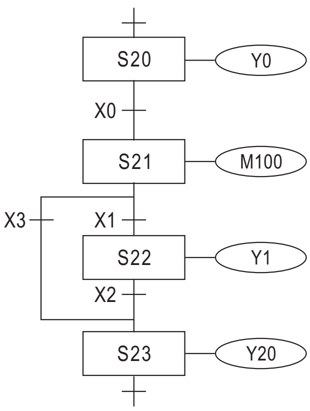
(c) Instruction Format

```
STL   S20
OUT   Y0
LD    X0
SET   S21
SET   S23
STL   S21
OUT   M100
LD    X1
SET   S22
STL   S22
OUT   Y20
STL   S23
OUT   M101
LD    X2
SET   S24
STL   S24
OUT   Y21
STL   S22
STL   S24
LD    X3
SET   S25
STL   S25
OUT   Y22
```

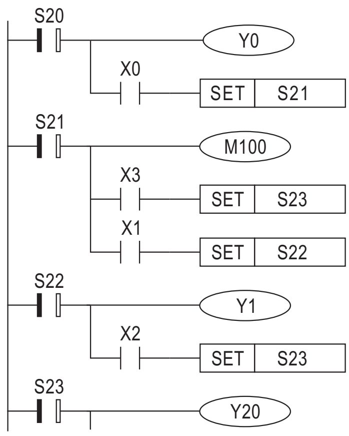
4-3-4 Jump SFC and Step Ladder Chart

Jump: To transfer the effective state to any state forward or any state in other flow.

(a) SFC



(b) Step Ladder Chart Format



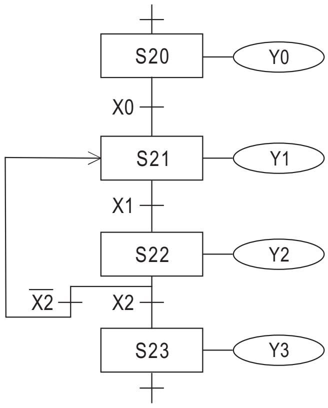
(c) Instruction Format

```
STL   S20
OUT   Y0
LD    X0
SET   S21
STL   S21
OUT   M100
LD    X3
SET   S23
LD    X1
SET   S22
STL   S22
OUT   Y1
LD    X2
SET   S23
STL   S23
OUT   Y20
```

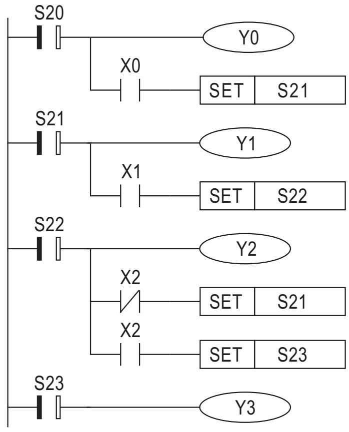
4-3-5 Repeat SFC and Step Ladder Chart

Repeat: When a flow is ended or the transferring condition is authorized, transferring the effective state to the initial state or any state in the front.

(a) SFC



(b) Step Ladder Chart Format



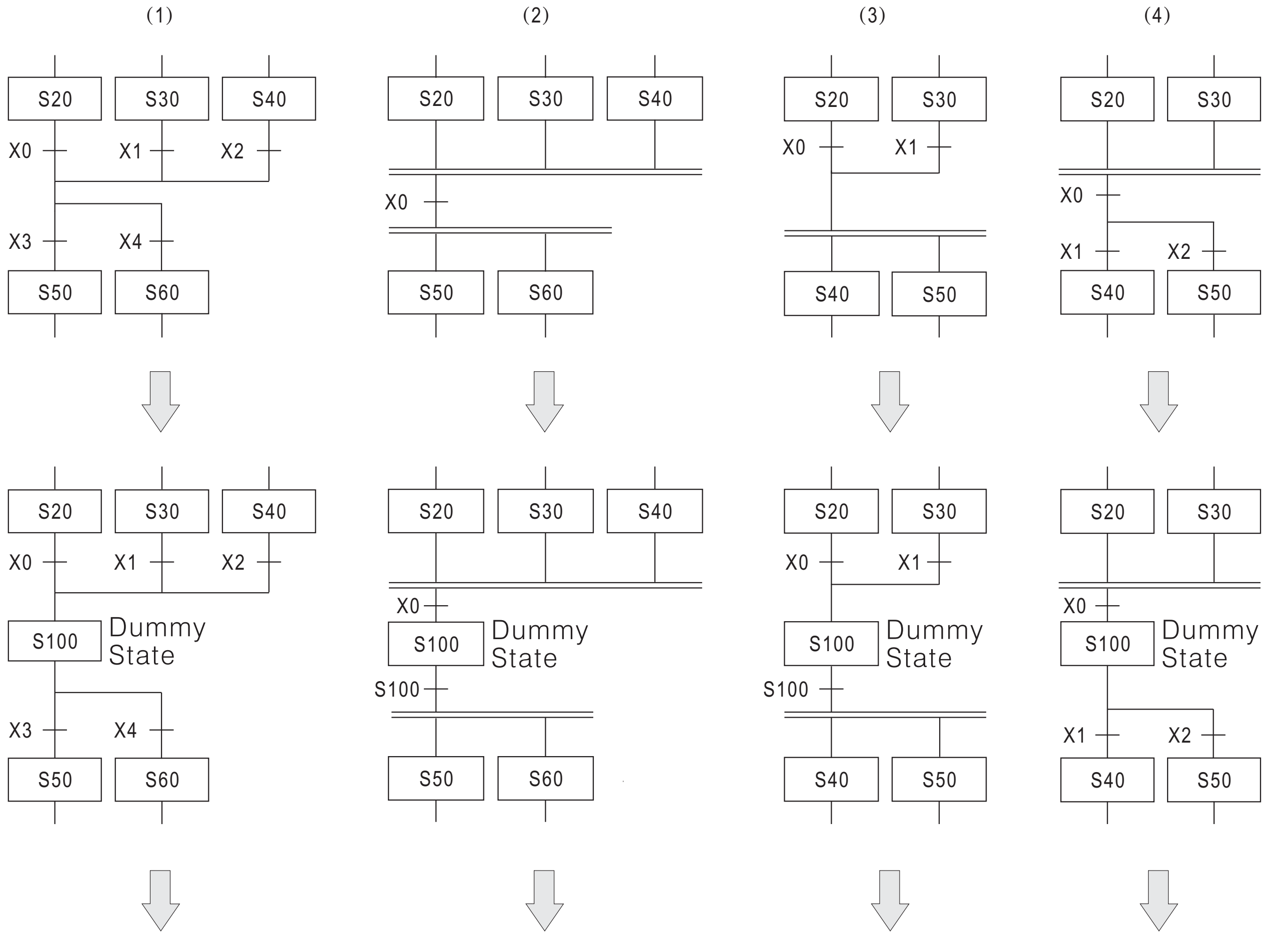
(c) Instruction Format

```
STL    S20
OUT    Y0
LD     X0
SET    S21
STL    S21
OUT    Y1
LD     X1
SET    S22
STL    S22
OUT    Y2
LDI    X2
SET    S21
LD     X2
SET    S23
STL    S23
OUT    Y3
```

4-4 Complex Branching, Merging Flows

4-4-1 Dummy State

It's recommended to set a null step point between merging and branching, when the branching processes right after merging. The null step point is called "Dummy State", because the Step point is only used for connection. Proper use of Dummy State will make SFC programming easier. The application of Dummy State is shown as below:




```
STL   S20
LD    X0
SET   S100
STL   S30
LD    X1
SET   S100
STL   S40
LD    X2
SET   S100
STL   S100
LD    X3
SET   S50
LD    X4
SET   S60
```

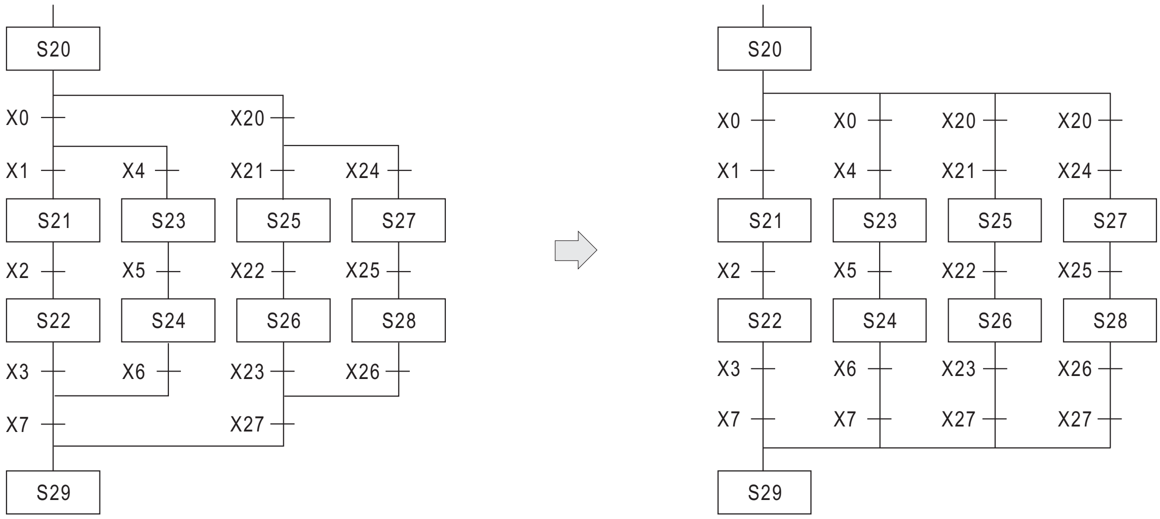
```
STL   S20
STL   S30
STL   S40
LD    X0
SET   S100
STL   S100
LD    S100
SET   S50
SET   S60
```

```
STL   S20
LD    X0
SET   S100
STL   S30
LD    X1
SET   S100
STL   S100
LD    S100
SET   S40
SET   S50
```

```
STL   S20
STL   S30
LD    X0
SET   S100
STL   S100
LD    X1
SET   S40
LD    X2
SET   S50
```

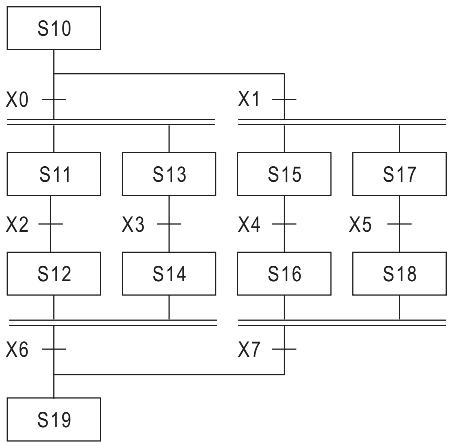
4-4-2 The Special Note for Branching and Merging

(1) If the original SFC is similar to the left side of SFC diagram, please rewrite it as the diagram as in the right.

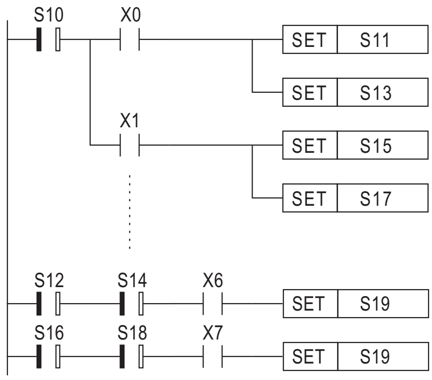


(2) For converting the left SFC to a Step Ladder Chart format, the branching and merging flows are rewritten as follows:

(a) SFC



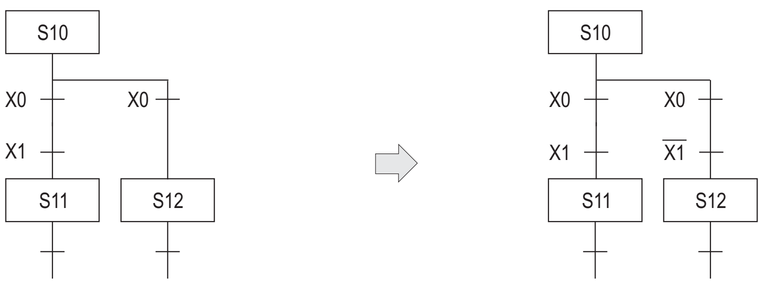
(b) Step Ladder Chart Format



(c) Instruction Format

```
STL   S10
LD    X0
SET   S11
SET   S13
LD    X1
SET   S15
SET   S17
  :
STL   S12
STL   S14
LD    X6
SET   S19
STL   S16
STL   S18
LD    X7
SET   S19
```

(3) To write a SFC program, the condition setting of transfer must be well-defined. For example, the diagram shows in the left side, which is unclear to indicate it as a Selective Branching or a Simultaneously Parallel Branching. Please rewrite the SFC as the right side diagram.



4-5 The Special Notes for Programming with Step Ladder Instructions

(1) If two states are using a specific Timer and the states are not next to each other. The Timer (which is using a same ID. number in two states) can be assigned different setting values in two states.

(2) It is available to use any Serial / Parallel links for the output of each state.

(3) It is also available to use Serial / Parallel link for the transferring condition of each state.

(4) If using an OUT instruction to drive an output in a state, the output status would be turned “OFF” after the effective state has been transferred.
If using a SET instruction to drive an output in a state, the output status would be still “ON” after the effective state has been transferred.

(5) When transferring the effective state between two states, there will be a scan time in which these two states are “ON”.

(6) If there is a Counter put after an STL contact point, the Counter will execute the reset function only when the STL contact point is “ON”.

(7) STL instructions are only effective to Step coil S. Step coil S can be used as general Auxiliary coil. But, after STL contact points, SET and RST are only two effective instructions for Step coil S.

(8) After STL contact points, MC and MCR instructions are not allowed to use.

(9) When designing a Step Ladder Chart, the sequence and ID. numbers of Step coils are unrelated.

(10) There is no limit on the number of Selective Branching, but at most 8 transferring states can be merged for Simultaneously Parallel Branching on a merging point, while the remaining states should be merged by another merging points in the program.

(11) A Step coil cannot use STL instructions repeatedly.

(12) The MPS instruction cannot be used directly after STL contact points.

(13) STL instructions are not allowed to use in subprograms.

(14) Although for STL instructions, the Jump instruction is not restricted to use. Because it would make processing procedures of programs more complicated, it is recommended avoid to use.

4-6 Special Coil and Special Register Related to SFC

In the table below, the symbol “■” represents that itis not allowed to use the instruction to drive the coil or write the data to the program.

Special Coil

Coil ID. NO.	Instruction of Function
M9040	STL transfer is prevented. When M9040= “ON”, the STL state transfer function is disabled.
■ M9046	STL state is “ON”. When M9047= “ON” and any coil of S0~S899=“ON” than M9046= “ON”.
M9047	STL monitoring is enable. D9040 ~ D9047 will be active only when M9047= “ON”.

Special Register

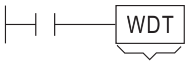
Register ID. NO.	Instruction of Function	
■ D9040	1st (the lowest) active STL step	When M9047=“ON”, the step point ID. numbers which are in action will be stored in D9040 ~ D9047. Where the smallest one will be stored in D9040, the second smallest one will be stored in D9041 and so forth.
■ D9041	2nd active STL step	
■ D9042	3rd active STL step	
■ D9043	4th active STL step	
■ D9044	5th active STL step	
■ D9045	6th active STL step	
■ D9046	7th active STL step	
■ D9047	8th active STL step	

5 General Rules for Applied Instructions

5-1 Formats of Applied Instructions

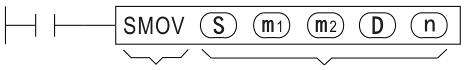
Instruction and Operand

- Each applied instruction has its unique instruction mnemonic, e.g. ADD, CMP…, Etc.
- Some applied instructions are made up by themselves:



Instruction

- Most of the applied instructions are constituted by themselves and some "Operands":



Instruction Operand

As shown above S, m1, m2, D, n are Operands. There are many types of Operand in applied instructions, their symbolic meanings are:

S :Source Operand (device). It usually refers to the Operand with unchanged contents after executed. S1, S2,… represent multiple source Operands for an instruction.

D :Destination Operand (device). It usually refers to the Operand in which instruction execution outcomes are stored. D1, D2,… represent multiple destination Operands for an instruction.

m, n : Those Operands used to specify operational constants. But some m, n of instruction can use Register D to execute indirect specification. m1, m2, n1, n2… Represent multiple m, n.

Devices for Operand

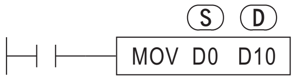
- Based on the needs, each applied instruction owns different number of Operands. And each applied instruction has different device ranges. The ranges of each Operand device are shown as in the following table:

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

- The "M" in the table above does not include Special Coil M9000 ~ M9255.
- The "D" in the table above does not include Special Register D9000 ~ D9255, and "SD" is specially pointed to D9000 ~ D9255.
- The "VZ index" in the table above indicates whether the Operand can be modified by Index Register V, Z.
- Under the applied instructions, if V, Z or SD is specified as the Operand Device, using V or Z for modification is prohibited.
- After organized, bit devices are displayed as K*n*X, K*n*Y, K*n*M, K*n*S to store data.
- T, C in the table above refer the current value registers of Timer (T) and Counter (C).
- All of T0 ~ T255, C0 ~ C199 and D are 16-bit registers. When the instruction specifies the process of 32-bit data, continuous two 16-bit registers will be occupied. For example, if a 32-bit Operand instruction specifies to D100, then a 32-bit register (composed of D101 and D100) will be used. while D101 will assigned for higher 16 bits and D100 for lower 16 bits. The same rules are also plied to T and C.
- 32-bit Counters (C200 ~ C255) only can be used as Operands of 32-bit instructions.

16-bit and 32-bit Instructions

- Because of different Operand value sizes, some of the applied instructions can be classified into 16-bit instructions and 32-bit instructions.



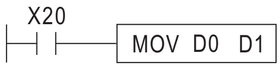
A 16-bit instruction, the content of D0 is transferred to D10.

A 32-bit instruction, the contents of (D1, D0) are transferred to (D11, D10).

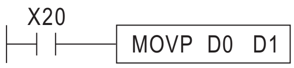
- A 32-bit instruction is displayed with a "D" (to be added directly BEFORE the instruction mnemonic), e.g. MOV represents a 16-bit instruction, while DMOV represents a 32-bit instruction.
- The device ID. number specified by an Operant of a 32-bit instruction can be an even or odd number. In order not to get confusion, it is recommended to use an even number, if it is possible.
- 32-bit Counters, C200 ~ C255, only can be used as Operands of 32-bit instructions.

Pulse Execution Instructions

- Based on requires, some applied instruction can be classified into sequential execution instructions and pulse execution instructions.



Sequential execution instruction: When X20= "ON", the instruction will be executed once in each scan cycle.



Pulse execution instruction: The instruction is only executed once when X20="OFF" → "ON".

- A pulse instruction displayed with a "P" (to be added directly AFTER the instruction mnemonic), e.g. MOV represents a sequential execution instruction, while MOVP represents a pulse execution instruction.
- Suitable using pulse execution instructions to replace sequential execution instructions in a program, can cut down unnecessary execution time.
- When X20= "OFF", both MOV and MOVP instuctions are not executed.

5-2 Data Process of Applied Instructions

- The X, Y, M and S are called bit devices, because they have only two different status ("ON"or "OFF"). But the T, C and D are called word devices because they are specially used to store data. Some bit devices can be a group together as a word device pattern, shown in the form of K*n*X, K*n*Y, K*n*M and K*n*S. This organized bits become a word device, that can be used in applied instructions for storage of data.

- When bit devices are organized as a word device, each digit of a hexadecimal word is composed by 4 bit devices. The K*n* portion of the statement identifies the range of devices included. The "*n*" can be a number from the range 1 to 8 and it actual represents 4**n* bit devices (*n* digits hexadecimal word). Hence all groups of bit devices are divisible by 4.

  K1M0 refers to a one-digit of hexadecimal word device, that is composed of M0 ~ M3.

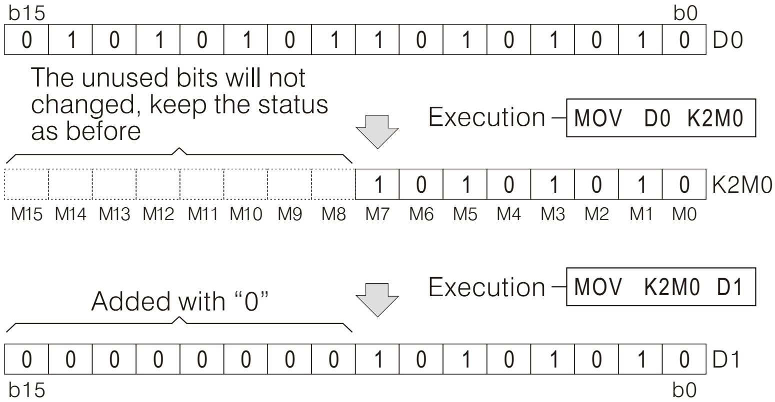
  K2M0 refers to a two-digit of hexadecimal word device, that is composed of M0 ~ M7.

  K4M0 refers to a four-digit of hexadecimal word device, that is composed of M0 ~ M15.

  K5M0 refers to a five-digit of hexadecimal word device, that is composed of M0 ~ M19.

  K8M0 refers to an eight-digit of hexadecimal word device, that is composed of M0 ~ M31.

- Data transference between registers and word devices which are composed of bit devices, the change should study up by the example below.



- When bit devices are organized as a word device, the header ID number of bit device can be specified as any legally device. But itis recommended that K*n*X and K*n*Y specify the ID number started with "0" such as X0, X20, Y20, Y30..., while K*n*M and K*n*S specify the ID number which is multiple of "8", such as M0, M8, M16.... The recommendations can improve system efficiency.

- When the Operand of an applied instruction is transformed to few sequential devices, the sequential ID number at different types are referred as below:

  ① Word Device (16 bits)

  D0, D1, D2, D3···············
  T0, T1, T2 ·····················
  C0, C1, C2 ····················

  ② Double-word Device (32 bits)

  D0 (D1, D0), D2 (D3, D2), D4 (D5, D4)············
  T0 (T1, T0), T2 (T3, T2), T4 (T5, T4)················
  C200, C201, C202 ················

  ③ Word Device Composed of Bit Devices

  K1X20, K1X24, K1X30, K1X34 ············
  K2Y20, K2Y30, K2Y40, K2Y50 ············
  K3M0, K3M12, K3M24, K3M36 ·········
  K4S0, K4S16, K4S32, K4S48 ·············

5-3 Using Index Register V and Z to Modify Operands

Index Register

16-bit	16-bit
V0	Z0
V1	Z1
V2	Z2
V3	Z3
V4	Z4
V5	Z5
V6	Z6
V7	Z7
Higher 16-bit	Lower 16-bit
32-bit	

- There are 16 of 16-bit Index Registers, V0 ~ V7 and Z0 ~ Z7, in M, VB and VH Series PLC.
- When using Index Registers V and Z in a 32-bit applied instruction, it must specifies an Index Register Z and the relative Index Register V will be taken over. For example, specifying Z0 will use two Index Registers(V0, Z0), the V0 is for higher16 bits and Z0 for lower 16 bits.
- The device at an applied instruction which can be modified by Index Register V, Z is shown below:

  X, Y, M, S, P, T, C, D, K, H, K*n*X, K*n*Y, K*n*M, K*n*S

- The following cannot be modified by V, Z:

  ① V, Z (themselves)

  ② SD (D9000 ~ D9255)

  ③ The *n* of K*n*

  ④ Used for Jump Destination or Subprogram Pointer P

- The followings are examples for operand modified by V and Z at an applied instruction.

  ① For a 16-bit applied instruction, when Z0=3

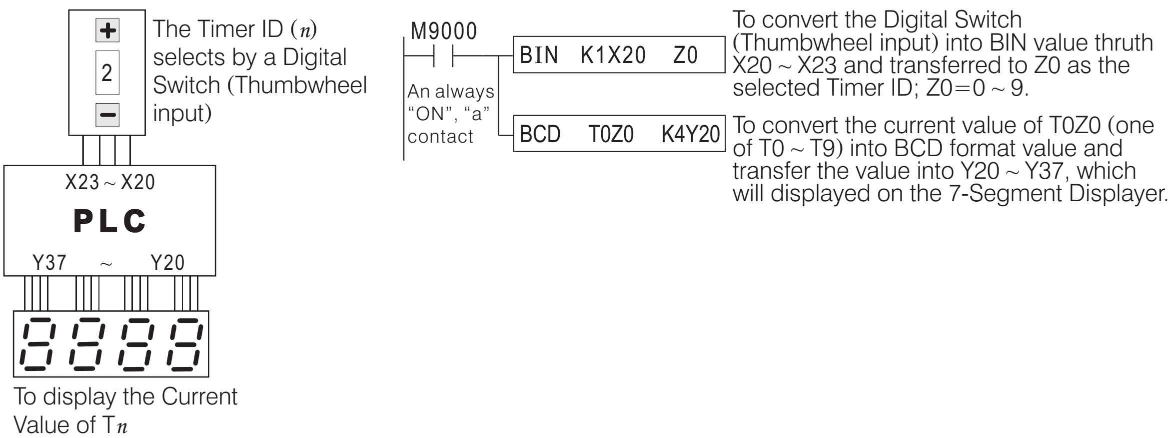
  Y20Z0=Y23
  T5Z0=T8
  D0Z0=D3
  K1M10Z0=K1M13

  ② For a 32-bit applied instruction (Index Registers V and Z will be taken over), when (V1, Z1)=8

  Y20Z1=Y30
  D0Z1=D8
  K8M40Z1=K8M48

- Example of use of Index Register

  Under the program and external inputs below, using Z0 to modify T0, and easily display the current value of T0 ~ T9 on the external 7-Segment Displayer.



5-4 The Special Notes for Using Applied Instructions

Flags

- The execution results are relate to Applied Instructions and causes some changes to corresponding flags:

  M9020: Zero Flag
  M9021: Borrow Flag
  M9022: Carry Flag
  M9029: Instruction Execution Completed Flag

- The execution results are relate to Applied Instructions and causes some changes to corresponding flags:

Limits on Using Applied Instructions

- Some of the Applied Instructions can only appear once in the program, the list showing below are those instructions.

  MTR (FNC52) PWM (FNC58)
  SORT (FNC69) HKY (FNC71) DSW (FNC72) SEGL (FNC74)
  PR (FNC77) LINK (FNC89) MBUS(FNC149)

  Using Index Registers to modify the instructions in operands, which will perform a better effect for the above-mentioned instructions.

- Some of the applied instructions can be used as many times as required, but the instruction executed at the same moment are limited the number of times.

  ① The instructions DHSCS, DHSRC and DHSZ executed in the program at same time, the number of times at most will be 6 in total.

  ② Only one RS instruction can be executed at the same time in the program.

Floating Point Instructions

- The list of relative Applied Instructions for processing floating point values.

FLT (FNC49)	DECMP (FNC110)	DEZCP (FNC111)	DEBCD(FNC118)
DEBIN (FNC119)	DEADD (FNC120)	DESUB (FNC121)	DEMUL(FNC122)
DEDIV (FNC123)	DESQR (FNC127)	INT (FNC129)	DSIN(FNC130)
DCOS (FNC131)	DTAN (FNC132)		

- Every floating point number will occupys two registers.
- The format of floating point number store in registers, please refer to Section 2-12 “Numerical System”.
- If the source operands of floating point operation instructions are assigned to constant numbers K or H, the instructions will let the constant numbers transform to a BIN floating point numbers for the processing.
- When using the floating point operation functions, please pay attention to the format of operands.

MEMO

6 Applied Instructions

M, VB and VH Series PLC has many applied instructions, each instruction has its specific function. PLC will achieve a complicated control system and diminish programming codes and programming development time effectively by using of these instructions. We hope readers will have an in-depth understanding of the applied instructions and make the best use of them.

6-1 Applied Instruction Table

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
Program Flow	00		CJ	P	Conditional jump	○	○	○	110
	01		CALL	P	Call subroutine	○	○	○	111
	02		SRET		Subroutine return	○	○	○	111
	03		IRET		Interrupt return	○	○	○	112
	04		EI		Enable interrupt	○	○	○	112
	05		DI		Disable interrupt	○	○	○	112
	06		FEND		First end	○	○	○	113
	07		WDT	P	Watch Dog Timer refresh	○	○	○	114
	08		FOR		Start of a FOR-NEXT loop	○	○	○	115
	09		NEXT		End of a FOR-NEXT loop	○	○	○	115
Compare and Move	10	D	CMP	P	Compare	○	○	○	118
	11	D	ZCP	P	Zone compare	○	○	○	119
	12	D	MOV	P	Move	○	○	○	120
	13		SMOV	P	Shift move	○	○	○	121
	14	D	CML	P	Compliment	○	○	○	122
	15		BMOV	P	Block move	○	○	○	123
	16	D	FMOV	P	Fill move	○	○	○	124
	17	D	XCH	P	Exchange	○	○	○	125
	18	D	BCD	P	Converts BIN → BCD	○	○	○	126
	19	D	BIN	P	Converts BCD → BIN	○	○	○	126
Arithmetic and Logical Operations	20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○	128
	21	D	SUB	P	Subtraction (S1) − (S2) → (D)	○	○	○	129
	22	D	MUL	P	Multiplication (S1)×(S2) → (D+1,D)	○	○	○	130
	23	D	DIV	P	Division (S1)÷(S2) → (D), (D+1)	○	○	○	131
	24	D	INC	P	Increment (D)+1 → (D)	○	○	○	132
	25	D	DEC	P	Decrement (D)−1 → (D)	○	○	○	132
	26	D	WAND	P	Logic word AND (S1) ∧ (S2) → (D)	○	○	○	133
	27	D	WOR	P	Logic word OR (S1) ∨ (S2) → (D)	○	○	○	133
	28	D	WXOR	P	Logic word exclusive OR (S1) ⊻ (S2) → (D)	○	○	○	133
	29	D	NEG	P	Negation ($\overline{D}$)+1 → (D)	○	○		134
Rotary and Shift	30	D	ROR	P	Rotation Right	○	○	○	136
	31	D	ROL	P	Rotation Left	○	○	○	136
	32	D	RCR	P	Rotation Right with carry	○	○	○	137
	33	D	RCL	P	Rotation Left with carry	○	○	○	137
	34		SFTR	P	Bit shift Right	○	○	○	138
	35		SFTL	P	Bit shift Left	○	○	○	138
	36		WSFR	P	Word shift Right	○	○		139
	37		WSFL	P	Word shift Left	○	○		140
	38		SFWR	P	Shift register write (FIFO Write)	○	○	○	141
	39		SFRD	P	Shift register read (FIFO Read)	○	○	○	142
Data Operation	40		ZRST	P	Zone reset	○	○	○	144
	41		DECO	P	Decode	○	○	○	145
	42		ENCO	P	Encode	○	○	○	146
	43	D	SUM	P	The sum of active bits	○	○		147
	44	D	BON	P	Check specified bit status	○	○		148
	45	D	MEAN	P	Mean	○	○		149
	46		ANS		Timed annunciator set	○	○		150
	47		ANR	P	Annunciator reset	○	○		150
	48	D	SQR	P	Square root	○	○		152
	49	D	FLT	P	BIN integer → Binary floating point format	○	○		153

∗ D ~ A 32 bit mode instruction option. ∗ P ~ Pulse (signal) operation option.

∗ ○ ~ The applicable PLC type

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
High-speed Processing	50		REF	P	I/O refresh	○	○	○	156
	51		REFF	P	I/O refresh and filter adjust	○	○		157
	52		MTR		Input matrix	○	○		158
	53	D	HSCS		High Speed Counter set	○	○	○	159
	54	D	HSCR		High Speed Counter reset	○	○	○	161
	55	D	HSZ		High Speed Counter zone compare	○	○		162
	56		SPD		Speed detection	○	○	○	167
	57	D	PLSY		Pulse Y output	○	○	○	168
	58		PWM		Pulse width modulation	○	○	○	169
	59	D	PLSR		Variable speed of Pulse output		○	○	170
Handy Instruction									
	61	D	SER	P	Search	○	○		174
	62	D	ABSD		Absolute Drum sequencer	○	○	○	175
	63		INCD		Incremental Drum sequencer	○	○	○	177
	64		TTMR		Teaching Timer	○	○		178
	65		STMR		Special Timer	○	○		179
	66		ALT	P	Alternate state	○	○	○	180
	67		RAMP		Ramp variable value	○	○	○	181
	69		SORT		Sort data	○	○		183
External Setting and Display	70	D	TKY		Ten Key input	○	○		186
	71	D	HKY		Hexadecimal Key input	○	○		187
	72		DSW		Digital Switch (Thumbwheel input)	○	○		189
	73		SEGD	P	Seven Segment Decoder	○	○	○	190
	74		SEGL		Seven Segment with Latch	○	○		191
	76		ASC		ASCII code Convert	○	○		193
	77		PR		Print	○	○		194
	78	D	FROM	P	Read from a special function block	○	○		195
	79	D	TO	P	Write to a special function block	○	○		195
External Serial Comm-unications	80		RS		Serial communication instruction	○	○	○	198
	81	D	PRUN	P	Parallel Run	○	○		202
	82		ASCI	P	Converts HEX → ASCII	○	○	○	203
	83		HEX	P	Converts ASCII → HEX	○	○	○	204
	84		CCD	P	Check Code	○	○	○	205
	85		VRRD	P	VR volume read	○	○	○	206
	86		VRSC	P	VR volume scale	○	○	○	207
	88		PID		PID control loop		○		352
	89		LINK		Easy Link communication	○	○		208
	149		MBUS		MODBUS communication		○	○	370
Floating Point	110	D	ECMP	P	Compares two BIN floating point values		○		214
	111	D	EZCP	P	Compares a BIN float range with a BIN float value		○		215
	118	D	EBCD	P	Converts BIN floating point format to DEC format		○		216
	119	D	EBIN	P	Converts DEC format to BIN floating point format		○		216
	120	D	EADD	P	Adds up two BIN floating point numbers		○		217
	121	D	ESUB	P	Subtracts one BIN floating point number from another		○		218
	122	D	EMUL	P	Multiplies two BIN floating point numbers		○		219
	123	D	EDIV	P	Divides one BIN floating point number from another		○		220
	127	D	ESQR	P	Square root of a BIN floating point value		○		221
	129	D	INT	P	BIN floating point → BIN integer format		○		222
	130	D	SIN	P	Calculates the sine of a BIN floating point value		○		223
	131	D	COS	P	Calculates the cosine of a BIN floating point value		○		224
	132	D	TAN	P	Calculates the tangent of a BIN floating point value		○		225

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
Others	90		DBRD	P	Reads data from the data bank	○	○		228
	91		DBWR	P	Writes data into the data bank	○	○		229
	147	D	SWAP	P	Swaps high/low byte	○	○		230
	169	D	HOUR		Operational Hour meter		○		376
	176		TFT		Timer (10 ms)	○	○	○	231
	177		TFH		Timer (100 ms)	○	○	○	232
	178		TFK		Timer (1 sec.)	○	○	○	233
Position Control	155	D	ABS		Absolute current value read		VB1		253
	156	D	ZRN		Zero position return		VB1		254
	157	D	PLSV		Pulse variable output		VB1		255
	158	D	DRVI		Drive to increment		VB1		256
	159	D	DRVA		Drive to absolute		VB1		257
Time & Convert	160		TCMP	P	Compare two times	○	○		236
	161		TZCP	P	Compare a time to a specified time range	○	○		237
	162		TADD	P	Adds ups two time values to get a new time	○	○		238
	163		TSUB	P	Subtracts one time value from another to get a new time	○	○		239
	166		TRD	P	Reads the RTC current value to a group of registers	○	○		240
	167		TWR	P	Sets the RTC to the value stored in a group of registers	○	○	○	241
	170	D	GRY	P	Converts BIN → Gray code	○	○		242
	171	D	GBIN	P	Converts Gray code → BIN	○	○		243
In-line Comparisons	224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○	246
	225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○	246
	226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○	246
	228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○	246
	229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○	246
	230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○	246
	232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○	246
	233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○	246
	234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○	246
	236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○	246
	237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○	246
	238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○	246
	240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○	246
	241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○	246
	242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○	246
	244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○	246
	245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○	246
	246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○	246
Newly Added Instructions	92		TPID		Temperature PID Control		V1.70		363
	250	D	SCL	P	Scaling (Translated by Coordinate)		V1.70		377
	251	D	SCL2	P	Scaling II (Translated by Coordinate)		V1.70		377
	151	D	DVIT		One-speed Interrupt Constant Quantity Feed		VB1		379
	153	D	LIR		Relatively Linear Interpolation		VB1		381
	154	D	LIA		Absolutely Linear Interpolation		VB1		384
	188		CRC	P	Cyclic Redundancy Check - 16		V1.72		387

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
A	20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○	128
	46		ANS		Timed annunciator set	○	○		150
	47		ANR	P	Annunciator reset	○	○		150
	62	D	ABSD		Absolute Drum sequencer	○	○	○	175
	66		ALT	P	Alternate state	○	○	○	180
	76		ASC		ASCII code Convert	○	○		193
	82		ASCI	P	Converts HEX → ASCII	○	○	○	203
	155	D	ABS		Absolute current value read		VB1		253
	232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○	246
	233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○	246
	234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○	246
	236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○	246
	237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○	246
	238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○	246
B	15		BMOV	P	Block move	○	○	○	123
	18	D	BCD	P	Converts BIN → BCD	○	○	○	126
	19	D	BIN	P	Converts BCD → BIN	○	○	○	126
	44	D	BON	P	Check specified bit status	○	○		148
C	00		CJ	P	Conditional jump	○	○	○	110
	01		CALL	P	Call subroutine	○	○	○	111
	10	D	CMP	P	Compare	○	○	○	118
	14	D	CML	P	Compliment	○	○	○	122
	84		CCD	P	Check Code	○	○	○	205
	131	D	COS	P	Calculates the cosine of a BIN floating point value		○		224
	188		CRC	P	Cyclic Redundancy Check - 16		V1.72		387
D	05		DI		Disable interrupt	○	○	○	112
	23	D	DIV	P	Division (S1)÷(S2) → (D), (D+1)	○	○	○	131
	25	D	DEC	P	Decrement (D)−1 → (D)	○	○	○	132
	41		DECO	P	Decode	○	○	○	145
	72		DSW		Digital Switch (Thumbwheel input)	○	○		189
	90		DBRD	P	Reads data from the data bank	○	○		228
	91		DBWR	P	Writes data into the data bank	○	○		229
	151	D	DVIT		One-speed Interrupt Constant Quantity Feed		VB1		379
	158	D	DRVI		Drive to increment		VB1		256
	159	D	DRVA		Drive to absolute		VB1		257
E	04		EI		Enable interrupt	○	○	○	112
	42		ENCO	P	Encode	○	○	○	146
	110	D	ECMP	P	Compares two BIN floating point values		○		214
	111	D	EZCP	P	Compares a BIN float range with a BIN float value		○		215
	118	D	EBCD	P	Converts BIN floating point format to DEC format		○		216
	119	D	EBIN	P	Converts DEC format to BIN floating point format		○		216
	120	D	EADD	P	Adds up two BIN floating point numbers		○		217
	121	D	ESUB	P	Subtracts one BIN floating point number from another		○		218
	122	D	EMUL	P	Multiplies two BIN floating point numbers		○		219
	123	D	EDIV	P	Divides one BIN floating point number from another		○		220
	127	D	ESQR	P	Square root of a BIN floating point value		○		221
F	06		FEND		First end	○	○	○	113
	08		FOR		Start of a FOR-NEXT loop	○	○	○	115
	16	D	FMOV	P	Fill move	○	○	○	124
	49	D	FLT	P	BIN integer → Binary floating point format	○	○		153
	78	D	FROM	P	Read from a special function block	○	○		195
G	170	D	GRY	P	Converts BIN → Gray code	○	○		242
	171	D	GBIN	P	Converts Gray code → BIN	○	○		243
H	53	D	HSCS		High Speed Counter set	○	○	○	159
	54	D	HSCR		High Speed Counter reset	○	○	○	161
	55	D	HSZ		High Speed Counter zone compare	○	○		162
	71	D	HKY		Hexadecimal Key input	○	○		187
	83		HEX	P	Converts ASCII → HEX	○	○	○	204
	169	D	HOUR		Operational Hour meter		○		376

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
I	03		IRET		Interrupt return	○	○	○	112
	24	D	INC	P	Increment (D)+1 → (D)	○	○	○	132
	63		INCD		Incremental Drum sequencer	○	○	○	177
	129	D	INT	P	BIN floating point → BIN integer format		○		222
L	89		LINK		Easy Link communication	○	○		208
	153	D	LIR		Relatively Linear Interpolation		VB1		381
	154	D	LIA		Absolutely Linear Interpolation		VB1		384
	224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○	246
	225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○	246
	226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○	246
	228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○	246
	229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○	246
	230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○	246
M	12	D	MOV	P	Move	○	○	○	120
	22	D	MUL	P	Multiplication (S1)×(S2) → (D+1.D)	○	○	○	130
	45	D	MEAN	P	Mean	○	○		149
	52		MTR		Input matrix	○	○		158
	149		MBUS		MODBUS communication		○	○	370
N	09		NEXT		End of a FOR-NEXT loop	○	○	○	115
	29	D	NEG	P	Negation $(\overline{D})+1 \rightarrow (D)$	○	○		134
O	240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○	246
	241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○	246
	242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○	246
	244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○	246
	245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○	246
	246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○	246
P	57	D	PLSY		Pulse Y output	○	○	○	168
	58		PWM		Pulse width modulation	○	○	○	169
	59	D	PLSR		Variable speed of Pulse output		○	○	170
	77		PR		Print	○	○		194
	81	D	PRUN	P	Parallel Run	○	○		202
	88		PID		PID control loop		○		352
	157	D	PLSV		Pulse variable output		VB1		255
R	30	D	ROR	P	Rotation Right	○	○	○	136
	31	D	ROL	P	Rotation Left	○	○	○	136
	32	D	RCR	P	Rotation Right with carry	○	○	○	137
	33	D	RCL	P	Rotation Left with carry	○	○	○	137
	50		REF	P	I/O refresh	○	○	○	156
	51		REFF	P	I/O refresh and filter adjust	○	○		157
	67		RAMP		Ramp variable value	○	○	○	181
	80		RS		Serial communication instruction	○	○	○	198
S	02		SRET		Subroutine return	○	○	○	111
	13		SMOV	P	Shift move	○	○	○	121
	21	D	SUB	P	Subtraction (S1) – (S2) → (D)	○	○	○	129
	34		SFTR	P	Bit shift Right	○	○	○	138
	35		SFTL	P	Bit shift Left	○	○	○	138
	38		SFWR	P	Shift register write (FIFO Write)	○	○	○	141
	39		SFRD	P	Shift register read (FIFO Read)	○	○	○	142
	43	D	SUM	P	The sum of active bits	○	○		147
	48	D	SQR	P	Square root	○	○		152
	56		SPD		Speed detection	○	○	○	167
	61	D	SER	P	Search	○	○		174
	65		STMR		Special Timer	○	○		179
	69		SORT		Sort data	○	○		183
	73		SEGD	P	Seven Segment Decoder	○	○	○	190
	74		SEGL		Seven Segment with Latch	○	○		191
	130	D	SIN	P	Calculates the sine of a BIN floating point value		○		223
	147	D	SWAP	P	Swaps high/low byte	○	○		230
	250	D	SCL	P	Scaling (Translated by Coordinate)		V1.70		377
	251	D	SCL2	P	Scaling II (Translated by Coordinate)		V1.70		377

Type	FNC No.	Instruction Title			Function	Applicable PLC Type			Ref. Page
		D		P		M	VB	VH	
T	64		TTMR		Teaching Timer	○	○		178
	70	D	TKY		Ten Key input	○	○		186
	79	D	TO	P	Write to a special function block	○	○		195
	92		TPID		Temperature PID Control		○		363
	132	D	TAN	P	Calculates the tangent of a BIN floating point value		○		225
	160		TCMP	P	Compare two times	○	○		236
	161		TZCP	P	Compare a time to a specified time range	○	○		237
	162		TADD	P	Adds ups two time values to get a new time	○	○		238
	163		TSUB	P	Subtracts one time value from another to get a new time	○	○		239
	166		TRD	P	Reads the RTC current value to a group of registers	○	○		240
	167		TWR	P	Sets the RTC to the value stored in a group of registers	○	○	○	241
	176		TFT		Timer (10 ms)	○	○	○	231
	177		TFH		Timer (100 ms)	○	○	○	232
	178		TFK		Timer (1 sec.)	○	○	○	233
V	85		VRRD	P	VR volume read	○	○	○	206
	86		VRSC	P	VR volume scale	○	○	○	207
W	07		WDT	P	Watch Dog Timer refresh	○	○	○	114
	26	D	WAND	P	Logic word AND (S1) ∧(S2) → (D)	○	○	○	133
	27	D	WOR	P	Logic word OR (S1) ∨(S2) → (D)	○	○	○	133
	28	D	WXOR	P	Logic word exclusive OR (S1)⊻(S2) → (D)	○	○	○	133
	36		WSFR	P	Word shift Right	○	○		139
	37		WSFL	P	Word shift Left	○	○		140
X	17	D	XCH	P	Exchange	○	○	○	125
Z	11	D	ZCP	P	Zone compare	○	○	○	119
	40		ZRST	P	Zone reset	○	○	○	144
	156	D	ZRN		Zero position return		VB1		254

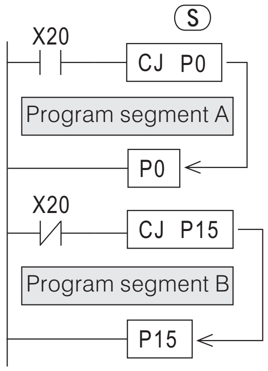
6-2 Program Flow Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
00		CJ	P	Conditional Jump	○	○	○
01		CALL	P	Call Subroutine	○	○	○
02		SRET		Subroutine Return	○	○	○
03		IRET		Interrupt Return	○	○	○
04		EI		Enable Interrupt	○	○	○
05		DI		Disable Interrupt	○	○	○
06		FEND		First End	○	○	○
07		WDT	P	Watch Dog Timer Refresh	○	○	○
08		FOR		Start of a FOR-NEXT Loop	○	○	○
09		NEXT		End of a FOR-NEXT Loop	○	○	○

	FNC 0 CJ	P	CJ P S	Conditional Jump	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S													○			○

- M and VB series, S=P0 ~ P255
- VH series, S=P0 ~ P63



S : Destination Pointer of Conditional Jump

- When the conditional contact for the CJ instruction becomes "OFF"(CJ is not active), the program will keep running. When the conditional contact for the CJ instruction becomes "ON"(CJ is active), program will execute Jump actions and jump to the destination of CJ, and then keeps on running.
- When X20= "OFF", the CJ P15 instruction will execute Jump actions, and Program B will not be executed.
- When X20="ON", the CJ P0 instruction will execute Jump actions, and Program A will not be executed.
- If the CJ instruction is not executed, the program segment enclosed will be executed as normal programs.

When the CJ instruction executes Jump actions, every device of the skipped program segment will change as follows:

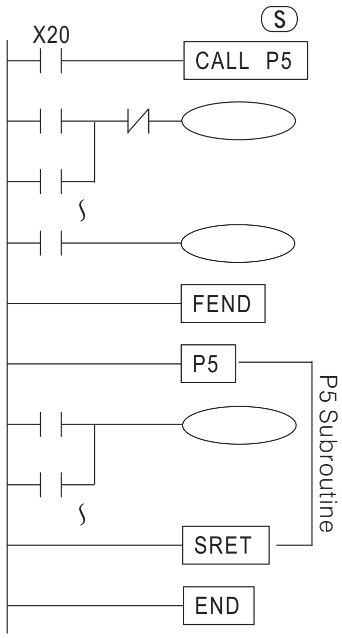
During Jump execution, the actions of every device in the program segment

- Y, M and S stay unchanged as before the Jump action.
- 10ms and 100ms Timers will stop counting time.
- 1ms Timer will continue to count time, but the output coil will not normally activate until the Jump stops.
- T192 ~ T199 will continue to count time and the output coil will also activate.
- High Speed Counter will continue to count and the output coil will also activate.
- Counter will stop counting.
- If the Reset instruction of Retentive timers and counters is driven before Jump, the device will still be reset during the Jump.
- Applied instructions will not be executed.

- Using the CJ instruction can skip unnecessary programs directly, so the program scan time can be saved.
- The CJ instruction can be used to solve the problem of double coil outputs.
- A Pointer numbered P can only appear once in a program; If the Pointer is specified more than once, errors will be incurred .
- As Pointer P255 is equal to the END address in a M or VB series program, CJ P255 is equal to jump to the END of a program.
- As Pointer P63 is equal to the END address in a VH series program, CJ P63 is equal to jump to the END of a program.

					M	VB	VH
	FNC 1 CALL	P	CALLP (S)	Call Subroutine	○	○	○
					M	VB	VH
	FNC 2 SRET		SRET	Subroutine Return	○	○	○

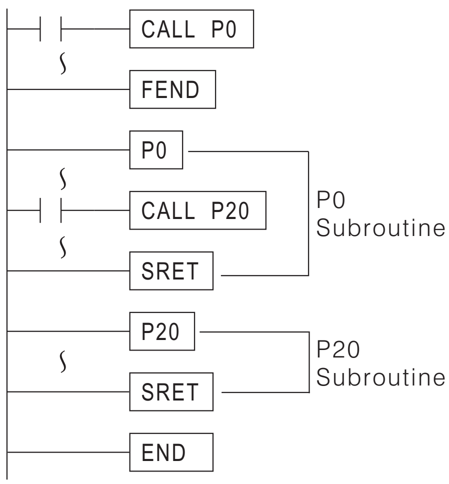
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S													○			○
• M and VB series, S = P0 ~ P254							• VH series, S = P0 ~ P62									



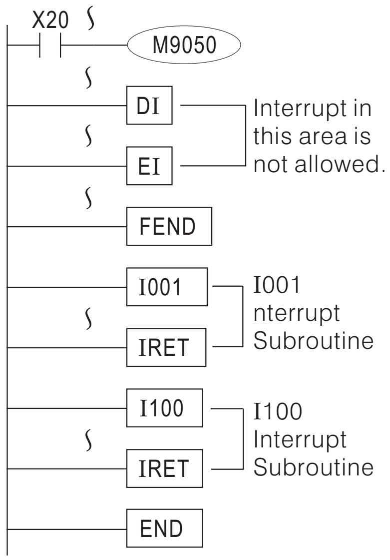
S : Subroutine Pointer

- When X20 = "ON", the CALL instruction will make the program flow jump to Pointer P5 to run subroutines, until an SRET instruction is encountered, where the program flow jumps back to the line of ladder logic immediately following the original CALL instruction and then keeps running.
- Subroutines should be written after the FEND instruction.
- If the CJ instruction and the CALL instruction are used in a program, the same Pointer number is not allowed.
- A same subroutine can be called in a program as many times as required.
- In a subroutine, a CALL instruction is available for calling other subroutines, while subroutines can be nested for 5 levels at most.
- The Timers used in the subroutine must be selected from the range T192 ~ T199 and T246 ~ T249. (VH series is not available).

- 2-Level Nest Subroutine Call (5 level at most)



					M	VB	VH
	FNC 3 IRET		IRET	Interrupt Return	○	○	○
	FNC 4 EI		EI	Enable Interrupt	○	○	○
	FNC 5 DI		DI	Disable Interrupt	○	○	○



- Generally a program is under Enable Interrupt status, but except the program flow is during the area between DI and EI, where the program is under Disable Interrupt.
- Assume that programs are under Enable Interrupt status: When X0= "OFF" → "ON", I001 Interrupt Subroutine will be executed until when the IRET instruction is encountered, then the flow returns to the main program and keep running. When X1= "ON" → "OFF ", I100 Interrupt Subroutine will be executed until when the IRET instruction is encountered, then the flow returns to the main program and keep running.
- When X20= "ON", the Interrupt Disable Special Coil M9050 is active and then I00□ is driven to disable Interrupt, the interrupt from the input terminal X0 is blocked.
- Please write Interrupt Pointer I after the FEND instruction.
- Generally, when the program flow executing an interrupt subroutine, all other interrupts are not allowed; But the EI and DI instructions interrupt subroutine can accept, this means that an interrupt subroutine may be interrupted during its operation, however at most 2 nested levels interrupt are accepted.

- The Timers used in general subroutines and interrupt subroutines must be selected from the range T192 ~ T199 and T246 ~ T249 (VH series is not available).
- When the program flow is worked between DI and EI, an interrupt demand cannot be executed immediately. The demand will be memorized, until the interrupt function is allowed, the interrupt subroutines will be executed.
- The pulse of the interrupt signal should be 200μs or longer.
- If the interrupt subroutine's I/O needs processed instantly, please use FNC53 immediate I/O refresh instruction.

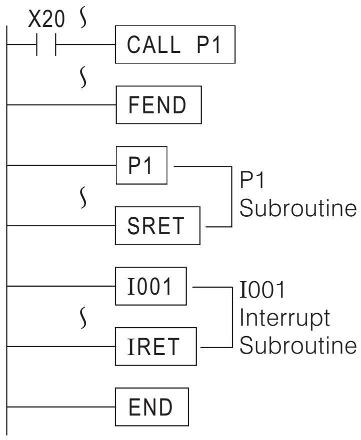
The assigned numbers for the Interrupt Pointer (I):

Input Interrupt		Timer Interrupt	High Speed Counter Interrupt
External Input Terminal	Interrupt Pointer	Interrupt Pointer	Interrupt Pointer
X0	I00 □	3 points: I6 □□ I7 □□ I8 □□	6 Points: I010 I020 I030 I040 I050 I060
X1	I10 □		
X2	I20 □		
X3	I30 □		
X4	I40 □		
X5	I50 □		
□=1, indicates the interrupt during rising □=0, indicates the interrupt during falling		□□=01 ~ 99, indicate Timer Interrupt interval length, where the time interval will be 1 ~ 99 ms	With the instruction FNC53 (DHSCS) to make a interrupt signal

Interrupt Control Special Coils:

Coil ID No.	Instruction of Function
M9050	Input interrupt I00□ is prevented.
M9051	Input interrupt I10□ is prevented.
M9052	Input interrupt I20□ is prevented.
M9053	Input interrupt I30□ is prevented.
M9054	Input interrupt I40□ is prevented.
M9055	Input interrupt I50□ is prevented.
M9056	Timer interrupt I6□□ is prevented.
M9057	Timer interrupt I7□□ is prevented.
M9058	Timer interrupt I8□□ is prevented.
M9059	High Speed Counter interrupt I010 ~ I060 is prevented.

	FNC 6 FEND		FEND	First End	M	VB	VH
					○	○	○



- An FEND instruction indicates the first end of a main program block.
- An FEND instruction placed before CALL instruction or after SRET instruction will be deemed as an error.
- Pointer P and Interrupt Pointer I which is specified by CALL instruction should be written behind the FEND instruction.
- If two or more FEND instructions are used, the subroutine should be placed between the last FEND instruction and END instruction.

					M	VB	VH
	FNC 7 WDT	P	WDTP	Watch Dog Timer Refresh	○	○	○

PLC is provided with a WTC (Watch Dog Timer), which is used to monitor operation condition of the PLC system. If any trouble occurs to PLC's CPU, through the WDT's monitoring, will command PLC to stop operation and turn all external output "OFF" to achieve the protection purpose.

The WDT (Watch Dog Timer) action statement:

WDT is a hardware timer (a 200ms timer, because when PLC= "STOP" → "RUN", the value of WDT will reload from the content value of Special Register D9000, while the setting value of D9000 is "200") counting time downward by a timing unit of 1ms. If the value reaches "0", WDT will determine that there is a system trouble, it forces the PLC to stop operation and turn all external output "OFF" to achieve the protection purpose. When the system operates normally, PLC will revert its WDT timer before it executes the beginning of program (STEP 0).

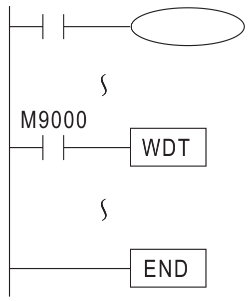
There are two reasons to activate WTD (Watch Dog Timer) function:

(1) Any trouble is happened in the PLC system and WDT performs the protection function.

(2) If the time of program execution is too long, the program's scan time more than the content value of D9000, it will triggers the protection function of WDT. Below are two approaches to improve the foregoing situation and make the system operate normally.

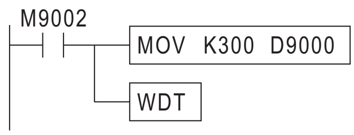
triggers the protection function of WDT. Below are two approaches to improve the foregoing situation and make the system operate normally.

① Insert WDT instruction into the program, because WDT instruction will revert the timing value of WDT.



② Use MOV instruction to change the content value of D9000.

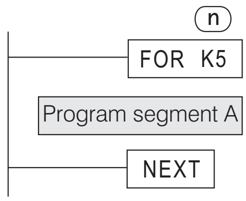
To adopt this approach, it should be noted that on the first scan time of PLC="STOP" → "RUN" , the value of WDT timer is still 200ms. The program below can be used for the solution where necessary.



					M	VB	VH
	FNC 8 FOR		FOR (n)	Start of a FOR-NEXT Loop	○	○	○
	FNC 9 NEXT		NEXT	End of a FOR-NEXT Loop	○	○	○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
n					○	○	○	○	○	○	○	○		○	○	○

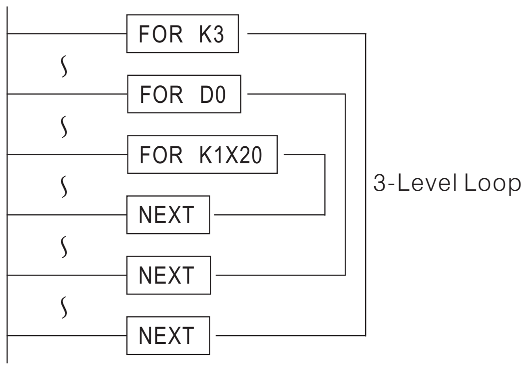
- n = 1 ~ 32,767 (Otherwise, n = 1, if the setting value exceeds beyond the range)



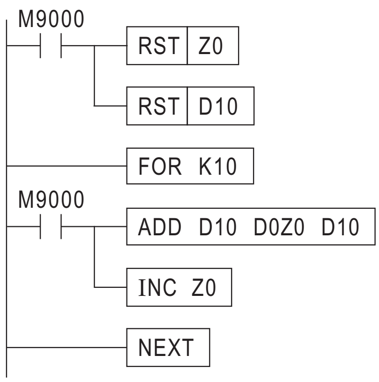
n : The number of times to be repeated in FOR-NEXT loop.

- The program in the FOR-NEXT loop will be executed “n” times.
- As in the left diagram, Program segment A is executed 5 times sequentially.

- In a For-Next loop, CJ instruction can be used to jump out of the loop.
- At most 5 levels can be used for a next FOR-NEXT loop. Be sure to note that the loop should be taken not to exceed WDT’s default value, otherwise an error will occur.
- Errors will occur under the following circumstances:
  NEXT instruction is placed in front of FOR instruction.
  NEXT instruction is placed behind FEND or END instruction.
  FOR instruction and NEXT instruction are not programmed as a pair.
- Multiple-level Loop Program



- Using FOR-NEXT Loop instructions jointly with Pointer Register V, Z will make programs more flexible. The program below will add up the content value of D0 ~ D9 and store the result in D10.



MEMO

6-3 Compare and Transfer Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
10	D	CMP	P	Compare	○	○	○
11	D	ZCP	P	Zone Compare	○	○	○
12	D	MOV	P	Move	○	○	○
13		SMOV	P	Shift Move	○	○	○
14	D	CML	P	Compliment	○	○	○
15		BMOV	P	n→n Block Move	○	○	○
16	D	FMOV	P	1→n Fill Move	○	○	○
17	D	XCH	P	Exchange	○	○	○
18	D	BCD	P	Converts BIN to BCD	○	○	○
19	D	BIN	P	Converts BCD to BIN	○	○	○

D	FNC 10 CMP	P	DCMPP (S1) (S2) (D)	Compare	M	VB	VH
					○	○	○

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D		○	○	○												○

- D occupies 3 consecutive devices

```
            (S1)  (S2)  (D)
X20
|-| |------[CMP  K100  D100  M100]
```

S_1 : Compare Value 1

S_2 : Compare Value 2

D : Compare Result; occupying 3 consecutive points

- Compare the data of (S1) with the data of (S2) and save the result in (D) (Compare Result).
- The CMP instruction will be enabled when X20= "ON".
  If K100>D100, then M100= "ON" ;
  If K100=D100, then M101= "ON" ;
  If K100<D100, then M102= "ON" .
- When X20= "OFF", the instruction is disabled, the status ("ON"/ "OFF") of M100, M101 and M102 remains as the status before X20= "OFF".
- Please use serial or parallel links of M100 ~ M102 to generate the result as "≥", "≤" or "≠".

D	FNC 11 ZCP	P	DZCPP S1 S2 S D	Zone Compare	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
S					○	○	○	○	○	○	○	○		○	○	○
D		○	○	○												○

- D occupies 3 consecutive devices
- $S_1 \leq S_2$

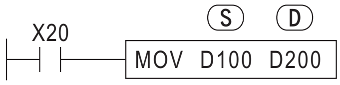
```
                 S1    S2    S     D
  X20
 ─┤ ├────┤ZCP  K100  K200  D100  M100 │
```

S_1 : Lower limit of zone compare
S_2 : Upper limit of zone compare
S : Compare Value
D : Compare Result; occupying 3 consecutive points

- Compare the data of S with the data of S_1, the data of S_2, and save the result in D (Compare Result).
- The CMP instruction will be enabled when X20= "ON".
  If K100>D100 (Lower Limit>Compare Value), then M100= "ON";
  If K100≤D100≤K200 (Compare Value is located between Upper Limit and Lower Limit), then M101= "ON";
  If K200<D100 (Compare Value>Upper Limit), then M102= "ON".
- The instructions is disabled when X20= "OFF". The status ("ON"/ "OFF") of M100, M101 and M102 remains as the status before X20= "OFF".
- When S_1 > S_2, the value of S_1 will become both of the Upper/Lower Limits to compare with S.

D	FNC 12 MOV	P	DMOVP (S) (D)	Move	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○



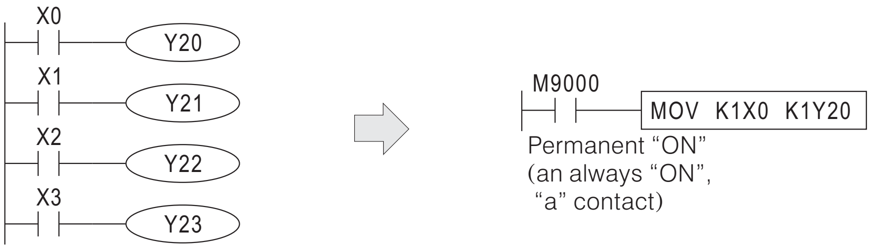
S : Source Device of Transfer

D : Destination Device

- To copy the designated value from S to D.
- The content value of D100 will be copied to D200 when X20= "ON".
- The instruction is disabled and D200 remains invariable when X20= "OFF".

Bit Transfer

To perform the program of left diagram, which can be changed as the right side.



32-bit Data Transfer

The instruction should be headed with a "D" when a 32-bit instruction is used.

When X20= "ON", move the content value (D1, D0) to (D101, D100).

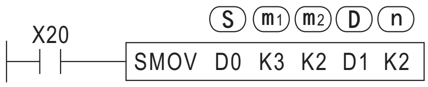
If the transfer target is a 32-bit counter, the instruction should be headed with a "D".

When X21= "ON", move the current value of C200 (32 bits) to (D11, D10).

FNC 13 SMOV	P	SMOVP (S) (m1) (m2) (D) (n)	Shift Move	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○		○
m1															○	
m2															○	
D									○	○	○	○		○		○
n															○	

- m_1=1~4; m_2=1~m_1; n=m_2~4



S : Source Device of Transfer

m_1 : The source position of the first digit to be moved

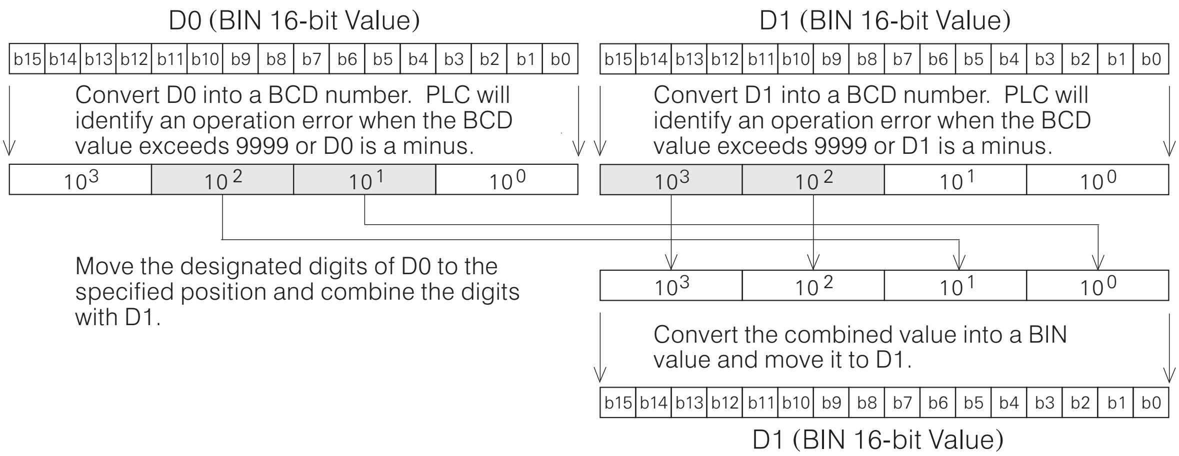
m_2 : The number of source digits to be moved

D : Destination Device

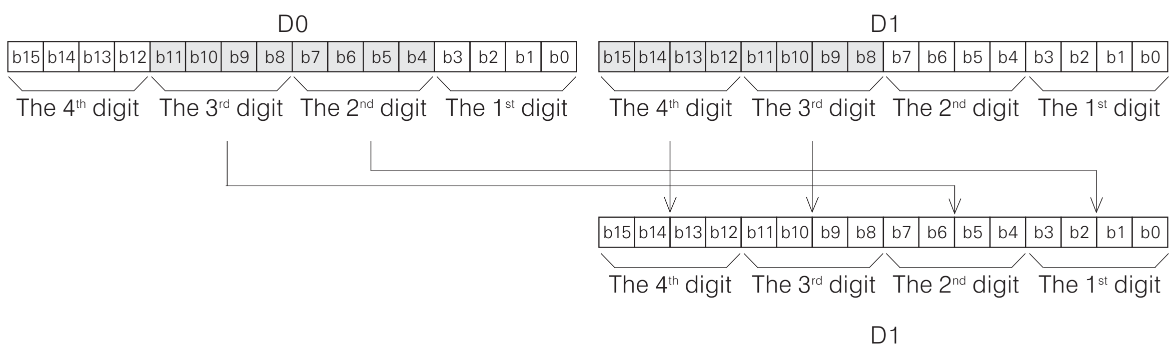
n : Destination position for the first digit

- This instruction can be used for data reorganization.
- The instruction can select different operation modes, it is based on the status of Special Coil M9168.

When M9168= "OFF"

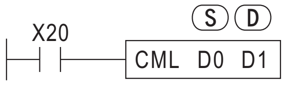


When M9168= "ON"



D	FNC 14 CML	P	DCMLP (S) (D)	Compliment	M	VB	VH
					○	○	○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○



S : Source Device of Transfer

D : Destination Device

- Invert all contents of (S) (i.e. “0” is inverted as”1” and “1”, inverted as “0”, for each digit) and copy the contents to (D) .
- When X20= “ON”, all of contents of D0 are inverted and copied to D1.
- When X20= “OFF”, the instruction is disabled and the contents of D1 remains invariable.

b15															b0	
1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	D0

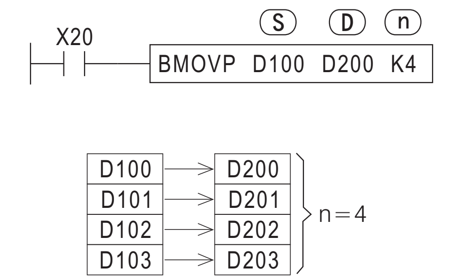
X20=ON

b15															b0	
0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	D1

FNC 15 BMOV	P	BMOVP (S) (D) (n)	n→n Block Move	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○					○
n															○	

- n=1～512



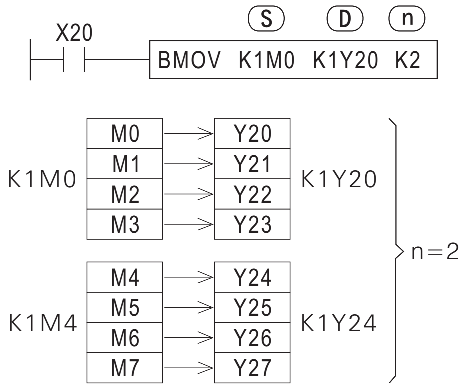
S : The head ID number of Source Device

D : The head ID number of Destination Device

n : Length of the block to be moved

- BMOV executes (S) → (D) "n" consecutive points of a data transfer.
- When X20= "OFF" → "ON", the content value of D100～D103 will be moved to D200～D203 orderly.

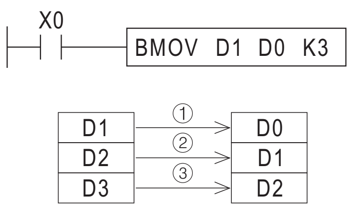
- When a block transfer of bit devices is executed, the data ranges of (S) and (D) should coincide.



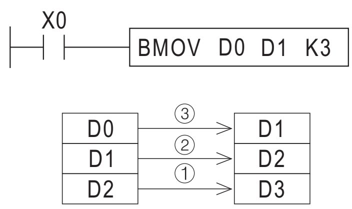
- When X20= "ON", K1M0 and K1M4 (equal to M0～M7) will be copied to K1Y20 and K1Y24 (equal to Y20～Y27).

- To prevent data writing errors during the transfer, the transfer will be processed in different orders when (S)>(D) or (S)<(D).

The transfer order when (S)>(D)



The transfer order when (S)<(D)



- Read and Write of File Register are required to be completed with the BMOV instruction. Please refer to Section 2-9 "File Register" for details.

D	FNC 16 FMOV	P	DFMOVP (S) (D) (n)	1→n Fill Move	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○					○
n															○	

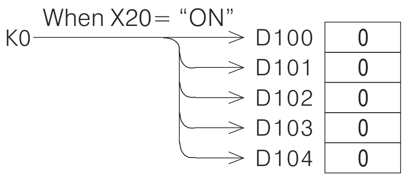
- n=1～512

S : Source Device of Transfer

D : The head ID number of Destination Device

n : Length of the block to be moved

- Move the content value of (S) to (n) registers which headed with (D).
- When X20= "ON", K0 will be copied to 5 continuous registers headed with D100 (D100～D104).
- If the range designated by (n) which is exceed the available devices space at the destination location, then only the available destination devices will be copied to.



D	FNC 17 XCH	P	DXCHP (D1) (D2)	Exchange	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D1						○	○	○	○	○	○	○		○		○
D2						○	○	○	○	○	○	○		○		○

X20 — XCHP D100 D200 [(D1) (D2)]

D1: Data 1 to be exchanged
D2: Data 2 to be exchanged

- Exchange (swap) contents values of the devices (D1) and (D2).
- When X20="OFF" → "ON", content values of (D100) and (D200) will be exchanged.

Before Execution		When X20= "OFF" → "ON"	After Execution	
123	D100		60	D100
60	D200		123	D200

X21 — DXCHP D0 D100 [(D1) (D2)]

- When X20="OFF" → "ON", content values of (D100) and (D200) will be exchanged.

Before Execution		When X21= "OFF" → "ON"	After Execution	
5	D0		15	D0
10	D1		20	D1
15	D100		5	D100
20	D101		10	D101

					M	VB	VH
D	FNC 18 BCD	P	DBCDP (S) (D)	Converts BIN to BCD	○	○	○
D	FNC 19 BIN	P	DBINP (S) (D)	Converts BCD to BIN	○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○		○
D						○	○	○	○	○	○	○		○		○

X20 ── BCD D0 K4Y20 ((S) D0, (D) K4Y20)

S : Converted Source (BIN)

D : Converted Result Destination (BCD)

- When X20= "ON", the BIN value in D0 will be converted into a BCD value. And then, moved to K4Y20 (Y20 ~ Y37).
- For a 16-bit instruction, PLC will identify an error when (S) exceeds the operational range (0 ~ 9,999).
- For a 32-bit instruction, PLC will identify an error when (S) exceeds the operational range (0 ~ 99,999,999).

X21 ── BIN K4X20 D1 ((S) K4X20, (D) D1)

S : Converted Source (BCD)

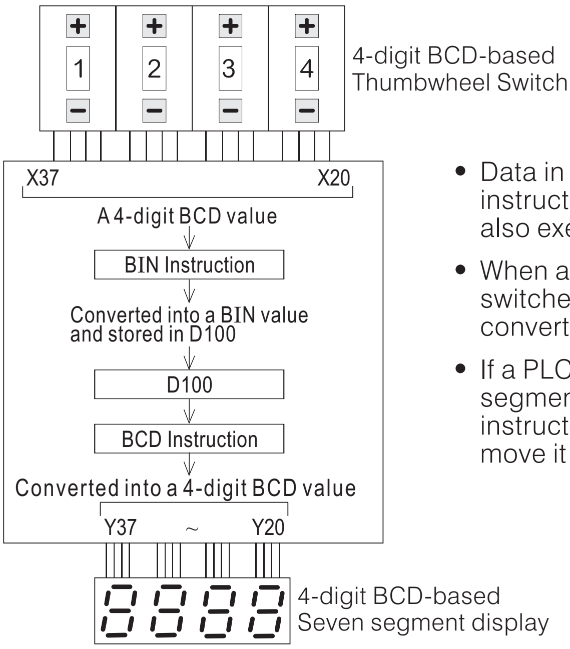
D : Converted Result Destination (BIN)

- When X21= "ON", the BCD value in K4X20 (X20 ~ X37) will be converted into a BIN value. And then, moved to D1.
- If the Source data is not provided in a BCD format, PLC will identify an operation error.

Application of BCD and BIN Instructions

M9000 ── BIN K4X20 D100 — Convert the BCD value of K4X20 into an equivalent BIN value and move it to D100.

── BCD D100 K4Y20 — Convert the BIN value of D100 into an equivalent BCD value and move it to K4X20.



- Data in PLC are all stored in a BIN format, and applied instructions (arithmetic and logical operations, ect.) are also executed based on BIN values.
- When a PLC is reading a BCD-based thumbwheel switches, which is required to use the BIN instruction to convert the data into a BIN value and store it in the PLC.
- If a PLC is used to output inner stored data to a seven segment display (BCD format), please use the BCD instruction to convert inner data into a BCD value and move it to the display.

6-4 Arithmetic and Logical Operations

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
20	D	ADD	P	Addition (S1)+(S2) → (D)	○	○	○
21	D	SUB	P	Subtraction (S1) – (S2) → (D)	○	○	○
22	D	MUL	P	Multiplication (S1)×(S2) → (D+1,D)	○	○	○
23	D	DIV	P	Division (S1)÷(S2) → (D),(D+1)	○	○	○
24	D	INC	P	Increment (D)+1 → (D)	○	○	○
25	D	DEC	P	Decrement (D) – 1 → (D)	○	○	○
26	D	WAND	P	Logic Word AND (S1)$\wedge$(S2) → (D)	○	○	○
27	D	WOR	P	Logic Word OR (S1)$\vee$(S2) → (D)	○	○	○
28	D	WXOR	P	Logic Word exclusive OR (S1)$\veebar$(S2) → (D)	○	○	○
29	D	NEG	P	Negation $(\overline{D})+1$ → (D)	○	○	

D	FNC 20 ADD	P	DADDP (S1) (S2) (D)	Addition (S1)+(S2) → (D)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

S1 : Summand
S2 : Addend
D : Total

- When X20= "OFF" → "ON" , the summand (D0) will be added to the addend (D1), and the total will be stored at the specified destination device (D2).

	10	D0
+	5	D1
	15	D2

- 16-bit Operation
  When the result of an operation, (D) , is equal to "0", the zero flag M9020= "ON".
  When the result of an operation exceeds 32,767, the carry flag M9022= "ON".
  When the result of an operation is less than -32,768, the borrow flag M9021= "ON".

- When X20= "ON", add (D1, D0) and (D3, D2) together and store the total in (D5, D4).

	100,000	(D1,D0)
+	– 100	(D3,D2)
	99,900	(D5,D4)

- 32-bit Operation
  When the result of an operation, (D), is equal to "0", the zero flag M9020= "ON".
  When the result of an operation exceeds 2,147,483,647, the carry flag M9022= "ON".
  When the result of an operation is less than -2,147,483,648, the borrow flag M9021= "ON".

D	FNC 21 SUB	P	DSUBP (S1) (S2) (D)	Subtraction (S1) – (S2) → (D)	M	VB	VH
					○	○	○

Operand	Devices																
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

```
          (S1)(S2)(D)
X20
─┤ ├───[SUBP  D0  D1  D2]
```

S_1 : Minuend

S_2 : Subtrahend

D : Remainder

- When X20= “OFF” → “ON” , the subtrahend (D1) will be subtracted from the minuend (D0), and the remainder will be stored at the destination device (D2).

	10	D0
–	5	D1
	5	D2

- 16-bit Operation
  When the result of an operation, (D), is equal to “0”, the zero flag M9020= “ON”.
  When the result of an operation exceeds 32,767, the carry flag M9022= “ON”.
  When the result of an operation is less than -32,768, the borrow flag M9021= “ON”.

```
          (S1)(S2)(D)
X20
─┤ ├───[DSUBP  D0  D2  D4]
```

- When X20="ON", subtract (D3, D2) from (D1, D0) and store the remainder in (D5, D4).

	100,000	(D1,D0)
–	– 100	(D3,D2)
	100,100	(D5,D4)

- 32-bit Operation
  When the result of an operation, (D), is equal to “0”, the zero flag M9020= “ON”.
  When the result of an operation exceeds 2,147,483,647, the carry flag M9022= “ON”.
  When the result of an operation is less than –2,147,483,648, the borrow flag M9021= “ON”.

D	FNC 22 MUL	P	DMULP (S1) (S2) (D)	Multiplication (S1)×(S2) → (D+1,D)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D											○					○

```
        (S1)(S2)(D)
X20
─┤ ├──[MUL D0 D1 D2]
```

S1 : Multiplicand

S2 : Multiplier

D : Product (of a multiplication)

- When X20= "ON", the multiplicand (D0) will be multiplied by the multiplier (D1), and the remainder will be stored at the destination device (D3, D2).

```
     10   D0
×     5   D1
─────────
     50   (D3,D2)
```

- Two 16-bit data sources multiplied together will create a 32-bit product.
- The Most Significant Bit (MSB) of a 32-bit product indicates a positive or negative ("0" represents a positive and "1" represents a negative).

```
        (S1)(S2)(D)
X20
─┤ ├──[DMULP D0 D2 D4]
```

- When X20= "ON", multiply (D1, D0) by (D3, D2) and store the product in (D7, D6, D5, D4).

```
      100,000   (D1,D0)
×         -10   (D3,D2)
─────────────
 -1,000,000   (D7,D6,D5,D4)
```

- A 32-bit multiplicand multiplied by a 32-bit multiplier will create a 64-bit product.
- The Most Significant Bit (MSB) of a 64-bit product indicates a positive or negative ("0" represents a positive and "1" represents a negative).

D	FNC 23 DIV	P	DDIVP S1 S2 D	Division (S1)÷(S2) → (D),(D+1)	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D											○					○

```
X20        S1 S2 D
─┤├──── DIVP D0 D1 D2
```

S_1 : Dividend

S_2 : Divisor

D : Quotient and Remainder

- When X20= "OFF" → "ON" , the dividend (D0) will be divided by the divisor (D1), and the quotient will be stored at the destination device (D2) while the remainder will be stored in (D3).

	10	D0
÷	−3	D1
Quotient	−3	D2
Remainder	1	D3

- In case a 16-bit quotient and a 16-bit remainder are created, the Most Significant Bit will indicate a positive or negative ("0" represents a positive and "1" represents a negative).

```
X20        S1 S2 D
─┤├──── DDIVP D0 D2 D4
```

- When X20= "OFF" → "ON" , divide (D1, D0) by (D3, D2) and store the quotient in (D5, D4), store the remainder in (D7, D6).

	−300	(D1,D0)
÷	−11	(D3,D2)
Quotient	27	(D5,D4)
Remainder	−3	(D7,D6)

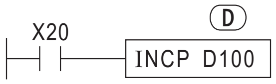
- In case a 32-bit quotient and a 32-bit remainder are yielded, the most significant bit will indicate a positive or negative ("0" represents a positive and "1" represents a negative).

Note:

- PLC will identify an operation error, if the divisor is equal to "0".
- The quotient of a positive dividend and a positive divisor (or a negative dividend and a negative divisor) will automatically be a positive; If either of a dividend or divisor is positive and the other is negative, the quotient will automatically be a negative.
- A positive dividend produces a positive remainder, while a negative dividend produces a negative remainder.

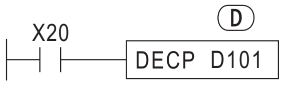
					M	VB	VH
D	FNC 24 INC	P	DINCP (D)	Increment (D)+1 → (D)	○	○	○
D	FNC 25 DEC	P	DDECP (D)	Decrement (D) – 1 → (D)	○	○	○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○



D : Destination Device

- When X20= “OFF” → “ON” , the current value of destination (D100) will have its value incremented (increased) by a value of “1”.
- If the instruction is not a pulse (P) instruction, (D100) will have its value incremented by a value of “1” in every scan cycle.
- In a 16-bit operation, when a value of “+32,767” is reached, the next increment of “1” will write a value of “-32,768” to the destination device.
- In a 32-bit operation, when a value of “+2,147,483,647” is reached, the next increment of “1” will write a value of “-2,147,483,648” to the destination device.
- The instruction operation result will never lead to any change of a flag.



D : Destination Device

- When X20= “OFF” → “ON” ,the current value of destination (D101) will have its value decremented (decreased) by a value of “1”.
- If the instruction is not a pulse (P) instruction, (D101) will have its value decremented by a value of “1” in every scan cycle.
- In a 16-bit operation, when a value of “-32,768” is reached, the next decrement of “1” will write a value of “+32,767” to the destination device.
- In a 32-bit operation, when a value of “-2,147,483,648” is reached, the next increment of “1” will write a value of “+2,147,483,647” to the destination device.
- The instruction operation result will never lead to any change of a flag.

D	FNC 26 WAND	P	DWANDP S1 S2 D	Logic Word AND $(S1) \wedge (S2) \rightarrow (D)$	M ○	VB ○	VH ○
D	FNC 27 WOR	P	DWORP S1 S2 D	Logic Word OR $(S1) \vee (S2) \rightarrow (D)$	M ○	VB ○	VH ○
D	FNC 28 WXOR	P	DWXORP S1 S2 D	Logic Word exclusive OR $(S1) \veebar (S2) \rightarrow (D)$	M ○	VB ○	VH ○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

```
         S1 S2 D
X20
─┤ ├───[WAND D0 D1 D2]
```

S1 : Source Device 1
S2 : Source Device 2
D : Operation Result

- When X20= "ON", 16 bits of (D0) and (D1) execute the logic AND operation and restore the result in (D2).
- The logic AND operation rules are: $0 \wedge 0=0$, $0 \wedge 1=0$, $1 \wedge 0=0$ and $1 \wedge 1=1$; any "0" will cause a result of "0".

0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	D0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D1
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	D2

```
         S1 S2 D
X21
─┤ ├───[WOR D3 D4 D5]
```

S1 : Source Device 1
S2 : Source Device 2
D : Operation Result

- When X20= "ON", 16 bits of (D3) and (D4) execute the logic OR operation, and restore the result in (D5).
- The logic OR operation rules are: $0 \vee 0=0$, $0 \vee 1=0$, $1 \vee 0=0$ and $1 \vee 1=1$; any "1" will cause a result of "1".

0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	D3
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D4
0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	D5

```
         S1 S2 D
X22
─┤ ├───[WXOR D6 D7 D8]
```

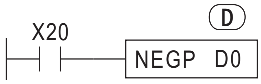
S1 : Source Device 1
S2 : Source Device 2
D : Operation Result

- When X20= "ON", 16 bits of (D6) and (D7) execute the logic XOR operation, and restore the result in (D8).
- The logic XOR operation rules are: $0 \veebar 0=0$, $0 \veebar 1=1$, $1 \veebar 0=1$ and $1 \veebar 1=0$; same values will cause a result of "0", otherwise, "1".

0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	D6
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D7
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	D8

D	FNC 29 NEG	P	DNEGP D	Negation $(\overline{D})+1 \rightarrow (D)$	M	VB	VH
					○	○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○



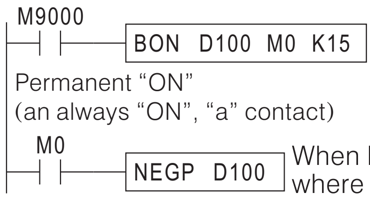
D : the selected device to be inverted

- When X20= “OFF” → “ON” , each single bit pattern of (D0) will be inverted (“0” inverted into “1” and vice versa) and then added with “1”. The result will be stored in (D0). The instruction select the complement of “2” for the value of . D The operation changes the positive or negative symbol of a value. For example,

Before execution 10 D0
↓ X20=OFF → ON
After execution −10 D0

Before execution −100 D0
↓ X20=OFF → ON
After execution 100 D0

- The absolute value of D100 can be generated with the following program.



The 15th bit (MSB) of D100 equaling to “1” indicates a negative value of D100, where M0= “ON”; otherwise, M0= “OFF”.

When M0= “OFF” → “ON”, we will select a negative value for D100, where the value of D100 will be inverted into a positive from a negative.

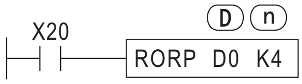
6-5 Rotary and Shift Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
30	D	ROR	P	Rotation Right	○	○	○
31	D	ROL	P	Rotation Left	○	○	○
32	D	RCR	P	Rotation Right with Carry	○	○	○
33	D	RCL	P	Rotation Left with Carry	○	○	○
34		SFTR	P	Bit Shift Right	○	○	○
35		SFTL	P	Bit Shift Left	○	○	○
36		WSFR	P	Word Shift Right	○	○	
37		WSFL	P	Word Shift Left	○	○	
38		SFWR	P	Shift Register Write (FIFO Write)	○	○	○
39		SFRD	P	Shift Register Read (FIFO Read)	○	○	○

D	FNC 30 ROR	P	⊣⊢ DRORP D n	Rotation Right	M ○	VB ○	VH ○
D	FNC 31 ROL	P	⊣⊢ DROLP D n	Rotation Left	M ○	VB ○	VH ○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○
n															○	

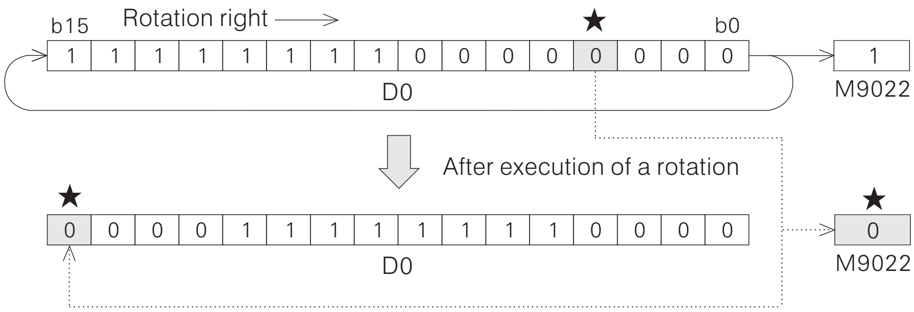
- The 16-bit instruction n=1 ~ 16
- The 32-bit instruction n=1 ~ 32
- When D is designated as KnY, KnM and KnS, the 16-bit instruction can only designate K4Y, K4M and K4S, while the 32-bit instruction can only designate K8Y, K8M and K8S

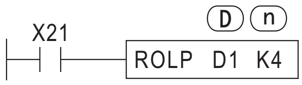


D : the selected device to be rotated

n : number of the bits to be rotated

- The bit pattern of the device designated by D is rotated n bit places to the right.
- When X20= "OFF" → "ON", the 16-bit data of (D0) will be rotated 4 bits to the right, and the status of the rotated data will be copied to the carry flag M9022.

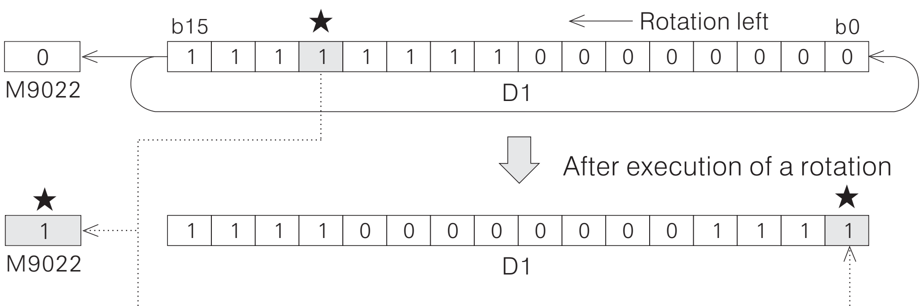




D : the selected device to be rotated

n : number of the bits to be rotated

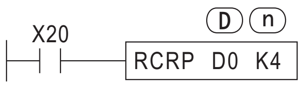
- The bit pattern of the device designated by D is rotated n bit places to the left.
- When X21= "OFF" → "ON", the 16-bit data of (D1) will be rotated 4 bits to the left, and the status of the rotated data will be copied to the carry flag M9022.



D	FNC 32 RCR	P	DRCRP (D) (n)	Rotation Right with Carry	M	VB	VH
					○	○	○
D	FNC 33 RCL	P	DRCLP (D) (n)	Rotation Left with Carry	M	VB	VH
					○	○	○

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○
n															○	

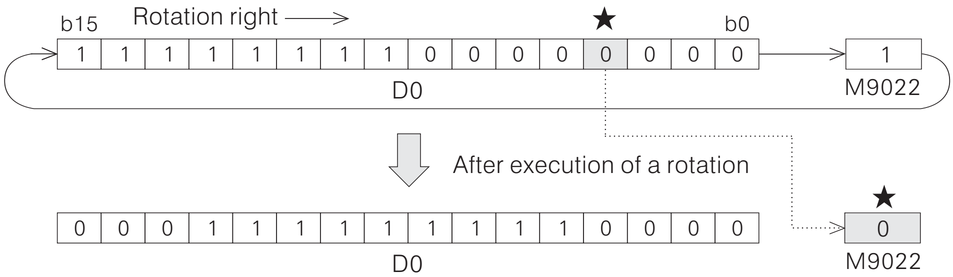
- The 16-bit instruction n=1 ~ 16
- The 32-bit instruction n=1 ~ 32
- When D is designated as K*n*Y, K*n*M and K*n*S, the 16-bit instruction can only designate K4Y, K4M and K4S, while the 32-bit instruction can only designate K8Y, K8M and K8S

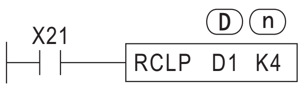


D : the selected device to be rotated

n : number of the bits to be rotated

- The contents of the device designated by (D) are rotated "n" bit places to the right with (n) bits extracted from the carry flag M9022.
- When X20= "OFF" → "ON", the 16-bit data of (D0) will be rotated 4 bits to the right with 4 bits extracted from the carry flag M9022.

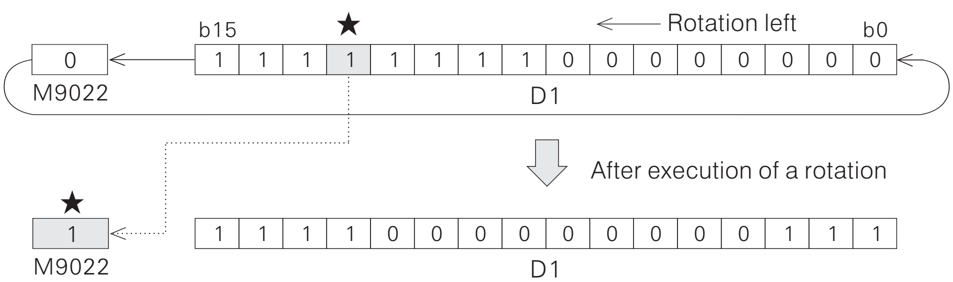




D : the selected device to be rotated

n : number of the bits to be rotated

- The contents of the device designated by (D) are rotated "n" bit places to the left with (n) bits extracted from the carry flag M9022.
- When X21= "OFF" → "ON", the 16-bit data of (D1) will be rotated 4 bits to the left with 4 bits extracted from the carry flag M9022.



FNC 34 SFTR	P	SFTRP (S) (D) (n1) (n2)	Bit Shift Right	M ○	VB ○	VH ○
FNC 35 SFTL	P	SFTLP (S) (D) (n1) (n2)	Bit Shift Left	M ○	VB ○	VH ○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○												○
D		○	○	○												○
n1															○	
n2															○	

- n1=1 ~ 1024
- n2=1 ~ n1

X20 — SFTRP X0 M0 K16 K4 (S) (D) (n1) (n2)

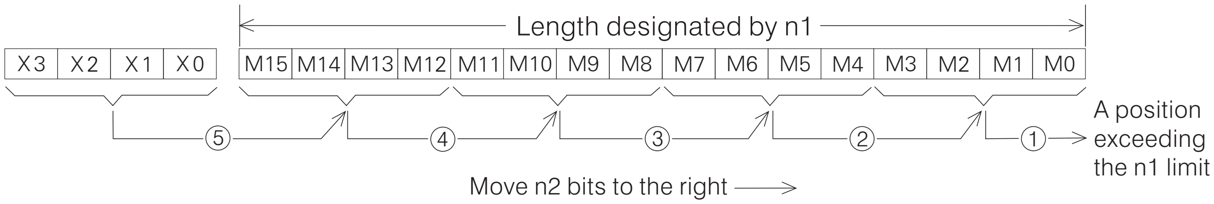
S : The head of source device ID number to be Moved in

D : The head of destination device ID number to be shifted

n1: data length to be shifted

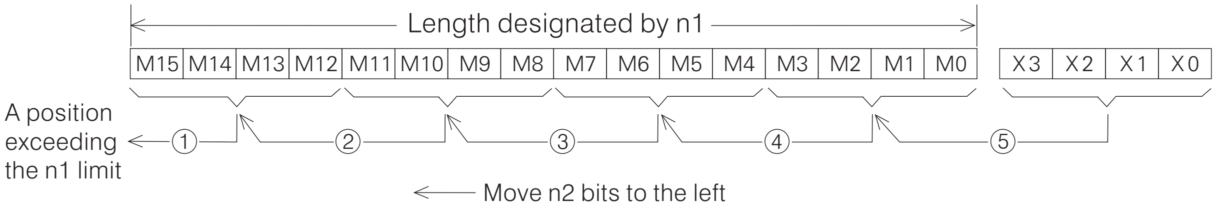
n2: number of the bits in a shift

- Move the length of (n1) bits of a device, headed with (D), (n2) bits to the right. A device headed with (S) will be used as the output complementary bit during the shift.
- When X20= "OFF" → "ON", the device composed of M0 ~ M15 (16 bits) will be moved 4 bits to the right; X0 ~ X3 will be moved in M12 ~ M15 for use of the output complementary bits.



X21 — SFTLP X0 M0 K16 K4 (S) (D) (n1) (n2)

- Move the length of (n1) bits of a device, headed with (D), (n2) bits to the left. A device headed with (S) will be used as the output complementary bit during the shift.
- When X21= "OFF" → "ON", the device composed of M0 ~ M15 (16 bits) will be moved 4 bits to the left; X0 ~ X3 will be moved in M0 ~ M3 for use of the output complementary bits.



FNC 36 WSFR	P	WSFRP (S) (D) (n1) (n2)	Word Shift Right	M	VB	VH
				○	○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○					○
n1															○	
n2															○	

• n1=1～512 • n2=1～n1

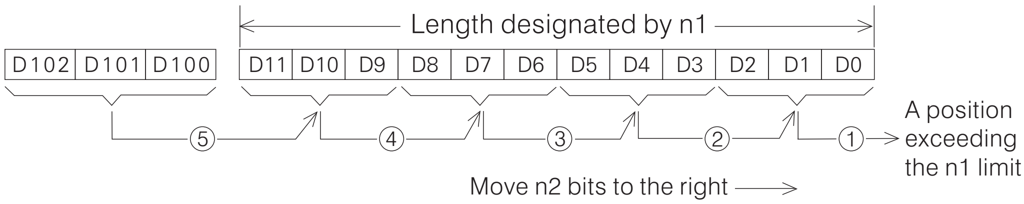
S : The head of source device ID number to be Moved in

D : The head of destination device ID number to be shifted

n1: data length to be shifted

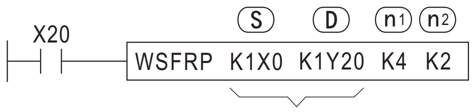
n2: number of the word in a shift

- Move a word stack with the length of (n1) words of a device, headed with (D), (n2) words to the right. A device headed with (S) will be used as the output complementary word during the shift.
- When X20= “OFF” → “ON”,the word stack composed of D0～D11 (12 words) will be moved 3 words to the right; D100～D102 will be moved in D9～D11 for use of the output complementary words.

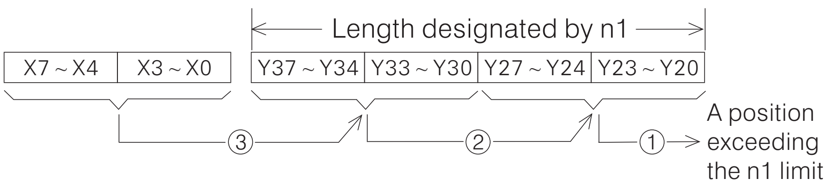


Note:

- The device properties designated by (S) and (D) must be the same (both are word devices or bit devices).
- When (S) and (D) designate bit devices, the digits of K*n* must be the same.



The digits designated by (S) and (D) must be the same.



	FNC 37 WSFL	P	WSFLP (S) (D) (n1) (n2)	Word Shift Left	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○					○
n1															○	
n2															○	

● n1=1 ~ 512 ● n2=1 ~ n1

X21 — WSFLP D100 D0 K12 K3 [(S) (D) (n1) (n2)]

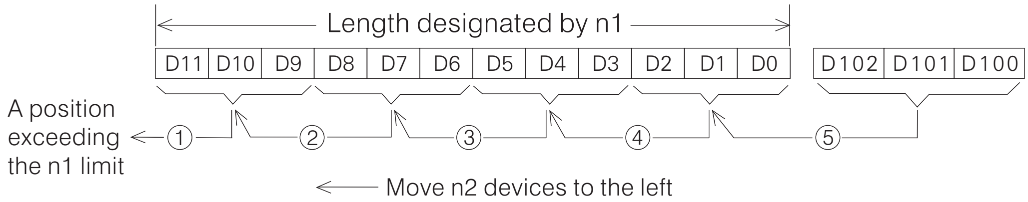
S : The head of source device ID number to be Moved in

D : The head of destination device ID number to be shifted

n1: data length to be shifted

n2: number of the word in a shift

- Move a word stack with the length of (n1) words of a device, headed with (D), (n2) words to the left. A device headed with (S) will be used as the output complementary word during the shift.
- When X21 = "OFF" → "ON", the word stack composed of D0 ~ D11 (12 words) will be moved 3 words to the left; D100 ~ D102 will be moved in D0 ~ D2 for use of the output complementary words.

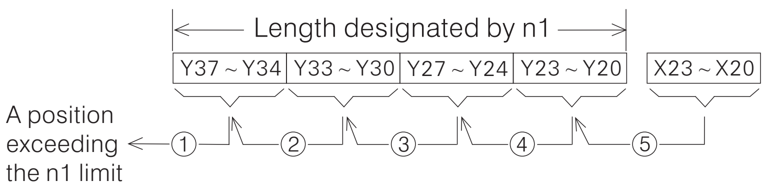


Note:

- The device properties designated by (S) and (D) must be the same (both are word devices or bit devices).
- When (S) and (D) designate bit devices, the digits of Kn must be the same.

X0 — WSFLP K1X20 K1Y20 K4 K1 [(S) (D) (n1) (n2)]

The digits designated by (S) and (D) must be the same.



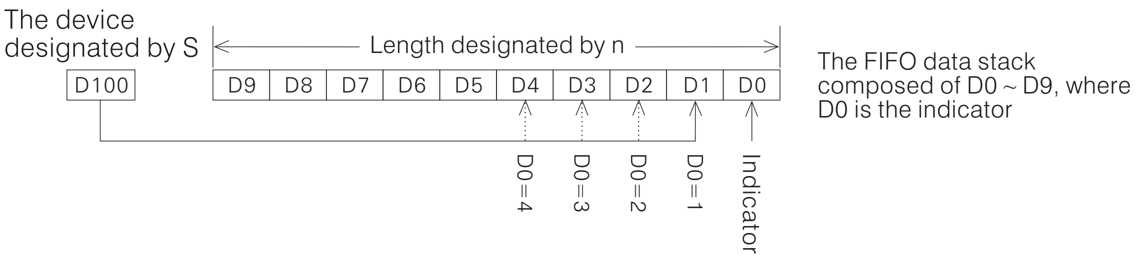
	FNC 38 SFWR	P	SFWRP (S) (D) (n)	Shift Register Write (FIFO Write)	M	VB	VH
					○	○	○

Operand	Devices																
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D									○	○	○					○
n															○	

- n=2 ~ 512

S : the device to be written to a FIFO data stack
D : source digit number of the FIFO data stack
n : Length of the FIFO data stack

- The data stack of (n) words, headed with (D), is defined as the FIFO data stack. The first device of the FIFO data stack is designated as the indicator. When the instruction is enabled, the content value of the indicator will be added with "1" firstly, and then, the content value of the device designated by will be moved to the position, designated by (S) the indicator, in the FIFO data stack.



- Suppose (D0)=0. When X20= "OFF" → "ON", the content value of (D0) will become "1" and the content value of (D100) will be moved to (D1). If, again, X20= "OFF" → "ON", the content value of (D0) will become "2" and the content value of (D100) will be moved to (D2), and so forth.
- (D0) records the position where it is written to the FIFO data stack. When the content value of (D0)≥(n-1), the instruction, if enabled again, will not allow data to be written any more, the value of (D0) will remain invariable and the carry flag M9022= "ON".
- This instruction (SFWR) is usually used jointly with the SFRD instruction, specified in the next page, to achieve the write/read control of the FIFO data stack.

FNC 39 SFRD	P	SFRDP (S) (D) (n)	Shift Register Read (FIFO Read)	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S									○	○	○					○
D						○	○	○	○	○	○	○		○		○
n															○	

- n=2 ~ 512

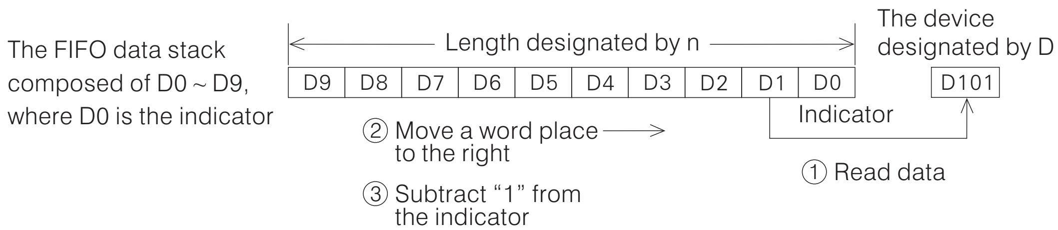
```
X21           (S)  (D)   (n)
|-| |-----[SFRDP  D0  D101  K10]
```

S : source digit number of the FIFO data stack

D : the device to be read from a FIFO data stack

n : Length of the FIFO data stack

- The data stack of (n) words, headed with (S), is defined as the FIFO data stack. The first device of the FIFO data stack is designated as the indicator. When the instruction is enabled, move the content value of the second device to the device designated by (D), and then, all of the FIFO data stack will be moved a word place to the right, and subtract "1" from the indicator's content value.



- Suppose (D0)=5. When X21= "OFF" → "ON",the content value of (D1) will be moved to (D100), D1 ~ D9 will be moved one word place to the right and the content value of (D0) will become 4, after subtracted by "1". If, again, X21= "OFF" → "ON", the content value of (D1) will be moved to (D101), D1 ~ D9 will be moved one word place to the right and the content value of (D0) will become 3, after subtracted by "1", and so forth.

- When the content value of (D0) equals to "0", the instruction, if enabled again, will not allow read data to be processed any more, the carry flag M9022= "ON" and the value of (D101) will remain invariable.

- This instruction (SFRD) is usually used jointly with the SFWR instruction, specified in the previous page, to achieve the write/read control of the FIFO data stack.

6-6 Data Operation Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
40		ZRST	P	Zone Reset	○	○	○
41		DECO	P	Decode	○	○	○
42		ENCO	P	Encode	○	○	○
43	D	SUN	P	The Sum of active bits	○	○	
44	D	BON	P	Check specified bit status	○	○	
45	D	MEAN	P	Mean	○	○	
46		ANS		Timed Annunciator set	○	○	
47		ANR	P	Annunciator Reset	○	○	
48	D	SQR	P	Square Root	○	○	
49	D	FLT	P	BIN integer → Binary floating point format	○	○	

	FNC 40 ZRST	P	ZRSTP (D1) (D2)	Zone Reset	M	VB	VH
					○	○	○

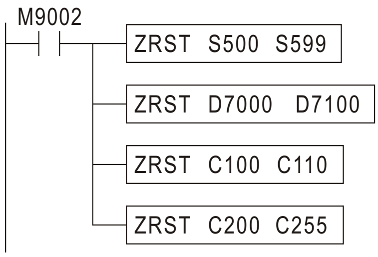
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D1		○	○	○					○	○	○					○
D2		○	○	○					○	○	○					○

- The ID number of device D1 must be less than or equal to (≤) the device D2.
- D1 and D2 have to designate the device of the same type.

Permanent “ON”
(Initial Pulse, “a” contact)

D1 : the device starting the Range Reset.
D2 : the device terminating the Range Reset.

- When the PLC is under “STOP” → “RUN”, M9002 will be “ON” for a Scan Time; All status of coils (M2000 ~ M2499) will be reset to “OFF”.



- The devices to be reset by the ZRST instruction consist of various bit devices and word devices.

- D1 and D2 have to designate the device of the same type, and the D1 device’s ID number must be less than or equal to (≤) the D2 device’s ID number. Only the device designated by D2 will be reset if the D1 ’s ID number is greater than the D2 ’s.
- This instruction can reset a 32-bit counter. It’s prohibited that D1 designates a 16-bit counter while D2 designates a 32-bit counter.

	FNC 41 DECO	P	DECOP (S) (D) (n)	Decode	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○					○	○	○	○		○	○	○
D		○	○	○					○	○	○	○		○		○
n															○	

- n=1 ~ 8, if D is a bit device.
- n=1 ~ 4, if D is a word device.

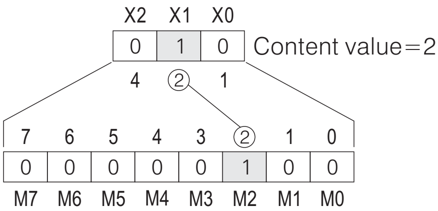
X20 —| |— DECOP X0 M0 K3 (S) (D) (n)

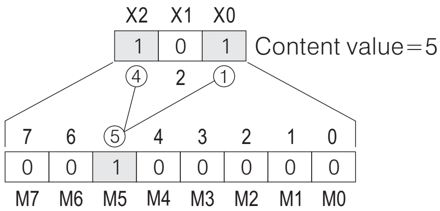
S : Decode source device

D : Destination device where decode results are stored

n : Length of decode bits

- In this example, (D) designates bit devices, which will occupy 2^n consecutive bit of devices headed with (D) to store decode results.
- Suppose the content value of X0 ~ X2 equals 2. When X20= "OFF" → "ON", the instruction DECO will decode the content value of X0 ~ X2 and move the results to M0 ~ M7, where M2= "ON".
- Suppose the content value of X0 ~ X2 equals 5. When X20= "OFF" → "ON", the instruction DECO will decode the content value of X0 ~ X2 and move the results to M0 ~ M7, where M5= "ON".

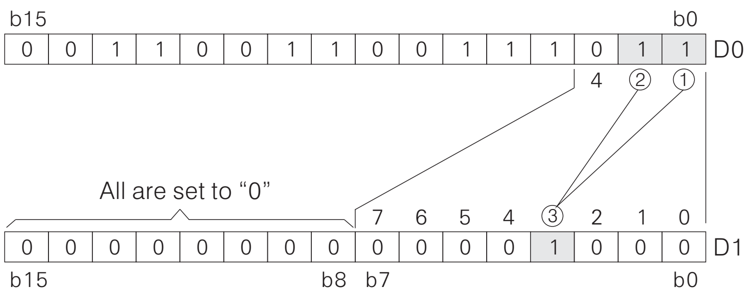




- In this example, (D) is a bit device, therefore n=1 ~ 8. When n=8, (D) will occupy 256 bit devices.

X20 —| |— DECOP D0 D1 K3 (S) (D) (n)

- In this example, (D) designates a bit device, therefore the range of (n) =1 ~ 4.
- When X20= "OFF" → "ON", the instruction DECO will decode the content value of (b0 ~ b2) of D0 and move the results to (b0 ~ b7) of D1. All unused data bits (b8 ~ b15) will be set to "0".



FNC 42 ENCO	P	ENCOP S D n	Encode	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○					○	○	○	○		○		○
D									○	○	○	○		○		○
n															○	

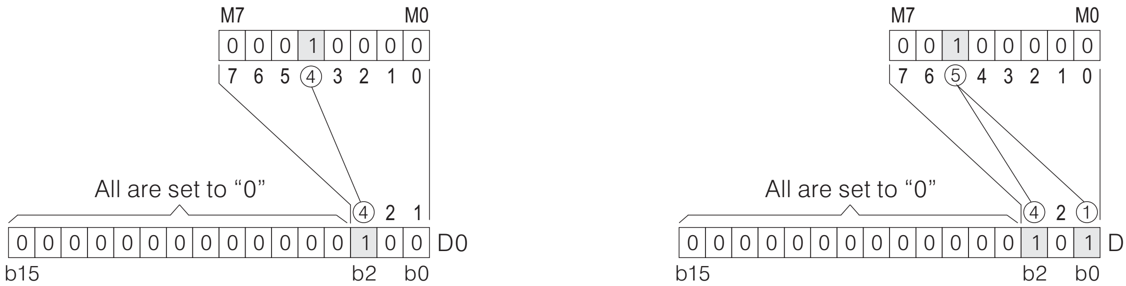
- n=1 ~ 8, if S is a bit device.
- n=1 ~ 4, if S is a word device.

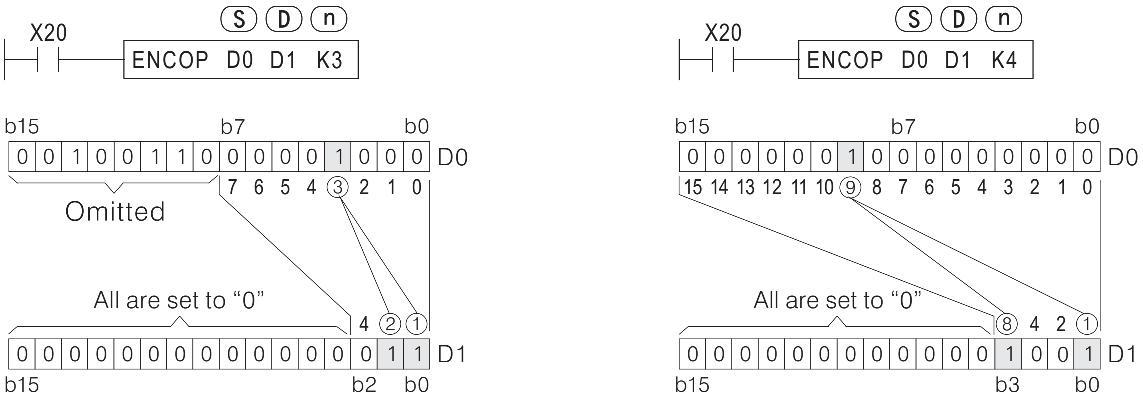
S : Encode source device

D : Destination device where encode results are stored

n : Length of encode bits

- When X20= "OFF" → "ON", the instruction ENCO will encode the contents of M0 ~ M7 and move the results to (b0 ~ b2) of D0. All unused data bits (b3 ~ b15) of D0 will be set to "0".
- In this example, S is a bit devices, therefore n=1 ~ 8. The effective range of S covers 256 bit devices when n=8.





Note:

- If there are more than a bit of the content of S equaling "1", the encode will be conducted on the basis of the largest number.
- PLC will identify an operation error if the content of S equals "0".
- When the conditional contact turns "OFF", the encode results (status of D) will remain.

D	FNC 43 SUM	P	DSUMP (S) (D)	The sum of active (“ON”) bits	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

X20 — SUM D0 D10 (S) (D)

S : Source device

D : Destination device where data are stored

- When X20= “ON”, the number of “1” (active) status within the 16 bits D0 are counted, and the among will be stored in D10. If all of the 16 bits of D0 equal “0”, then the zero flag M9020= “ON”.

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

D0 $\xrightarrow{X20=ON}$ 8 (D10)

- When a 32-bit instruction DSUM is used, (D) will still occupy 2 registers.

D	FNC 44 BON	P	DBONP (S) (D) (n)	Check specified bit active ("ON") status	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D		○	○	○												○
n															○	

- n=0 ~ 15, for a 16-bit instruction.
- n=0 ~ 31, for a 32-bit instruction.

```
        (S)(D)(n)
 X20
─┤ ├───[BON D0 M0 K5]
```

S : Source device.

D : Destination device where specified results are stored.

n : the designated position bit to be specified.

- Copy the status of the (n) th bit of the designated source device (S) to the destination device (D).
- When X20= "ON", b5 of D0 will be copied to M0.
- When X20= "OFF", the status of M0 will remain.

b15										b5					b0		
0	0	1	1	0	0	1	1	0	0	0	0	1	1	1	1	D0	$\xrightarrow{X20=ON}$ M0=OFF

b15										b5					b0		
0	0	0	0	1	1	1	1	0	0	1	1	1	1	0	0	D0	$\xrightarrow{X20=ON}$ M0=ON

D	FNC 45 MEAN	P	DMEANP (S) (D) (n)	Mean	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D						○	○	○	○	○	○	○		○		○
n															○	

- n = 1～64

```
X20          (S)  (D)  (n)
─┤ ├────┤MEAN  D0  D10  K5 ├
```

S : Head ID number of source devices to be generated a mean.

D : Destination device where the mean is stored.

n : Number of consecutive devices to be generated a mean.

- To sum up the content values of (n) consecutive devices which headed with (S), then generate a mean value and store it in a device designated by (D).
- When X20= "ON", generate a mean of the content values of consecutive 5 registers (D0 ~ D4) and store it in D10.

$$\frac{(D0) + (D1) + (D2) + (D3) + (D4)}{5} \xrightarrow{X20=ON} (D10)$$

100	D0
150	D1
200	D2
88	D3
100	D4

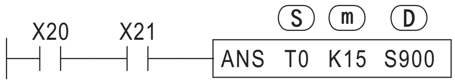
$\xrightarrow{X20=ON}$ 127 D10

The remainder of the calculated mean is ignored. (Remainder=3)

- Ignore the remainder, if any remainder comes out during the operation process.
- If the designated device's ID number exceeds the range, the device will only be processed within to prescribed range.

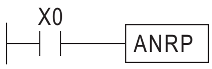
FNC 46 ANS		ANS (S) (m) (D)	Timed Annunciator Set	M	VB	VH
				○	○	
FNC 47 ANR	P	ANRP	Annunciator Reset	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S									○							○
m															○	
D				○												○
• S=T0~T199 • m=1~32767 • D=S900~S999																



S : Detect alarm timer
m: Timer configuration
D : Annunciator

- The instruction ANS is used exclusively to drive the instruction of annunciator outputs.
- When X20 and X21 turn "ON" for more than 1.5 seconds simultaneously, the annunciator S900= "ON" (to be driven). After S900= "ON", X20 or X21 turns "OFF", the contact of T0 becomes "OFF" and the current value is returned as "0" , but S900 will remain "ON".
- When both X20 and X21 turn "ON" simultaneously but less than 1.5 seconds, then either one of them turns "OFF", the current value of T0 will be returned as "0".
- Do not use a timer which has been assigned to this instruction.



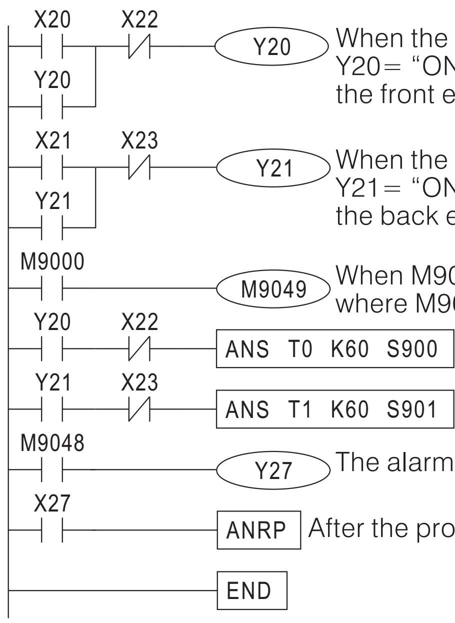
- The instruction ANR is used exclusively to reset the instruction of annunciator. When each time the ANR instruction is operated, annunciators which have been activated are sequentially reset one-by-one.
- When X0= "OFF" → "ON", the instruction ANR will be executed and the active annunciator will be reset to "OFF".
- If the instruction ANR is executed and if there are more than one active annunciator, the smallest active annunciator ID number will be reset. When the instruction ANR is executed once again, in this moment the smallest (which was the second smallest) active annunciator ID number will be reset. And so forth to reset other active annunciators.

Application Examples of Timed Annunciator Set

- When the special auxiliary coil M9049= "ON" and any assigned annunciator of S900 ~ S999 is activated, then M9048= "ON" and D9049 will display the annunciator number. If there are more than one annunciator being activated simultaneously, D9049 will display the smallest active annunciator ID number.

- The following chart is a Timed Annunciator Set loop

X20: Forward Switch	Y20: Forward Device	S900: Forward Annunciator
X21: Backward Switch	Y21: Backward Device	S901: Backward Annunciator
X22: Front End Position Switch	Y27: Alarm Indicator	
X23: Back End Position Switch		
X27: Annunciator Reset Button		



When the forward switch contact X20= "ON", the forward device Y20= "ON" and remain. Y20 will become "OFF" when the object reaches the front end position (X22= "ON").

When the backward switch contact X21= "ON", the backward device Y21= "ON" and remain. Y21 will become "OFF" when the object reaches the back end position (X23= "ON").

When M9049= "ON", the alarm monitor will be activated, where M9048 and D9049 are effective.

If Y20= "ON" for more than 6 seconds and does not reach the front end position, then S900= "ON".

If Y21= "ON" for more than 6 seconds and does not reach the back end position, then S901= "ON".

The alarm indicator Y27= "ON" if any one of annunciator is "ON".

After the problem is cleared, press once X27 to reset the annunciator.

D	FNC 48 SQR	P	DSQRP (S) (D)	Square Root	M	VB	VH
					○	○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○				○	○
D											○					○

```
              (S)(D)
   X20
|--| |-------[SQR  D0  D1]
```

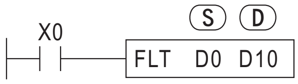
S : Source Device for performing mathematical square roots.

D : Destination device where the result is stored.

- This instruction performs a square root operation on the content value of device (S) and stores the result to the destination device (D).
- We perform a square root operation on the content value of D0 and stores the result at D1 when X20= "ON".
- In the result, only the integer part will remain, while the decimal part will be ignored; If any decimal is ignored, then M9021= "ON".
- Zero Flag M9020= "ON" when the operation result is equal to "0".
- (S) must be a positive; a negative will be determined an error operation by PLC and M9067 will be set "ON".

D	FNC 49 FLT	P	DFLTP S D	BIN integer → Binary floating point format	M	VB	VH
					○	○	

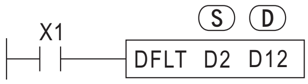
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source data

D : Destination device to store the equivalent float format value

- When X0= "ON",performs the convert operation from the content value of 16 bits register D0 (which is a BIN integer) to a binary floating point number, and copies the converted result to the destination devices (D11,D10).

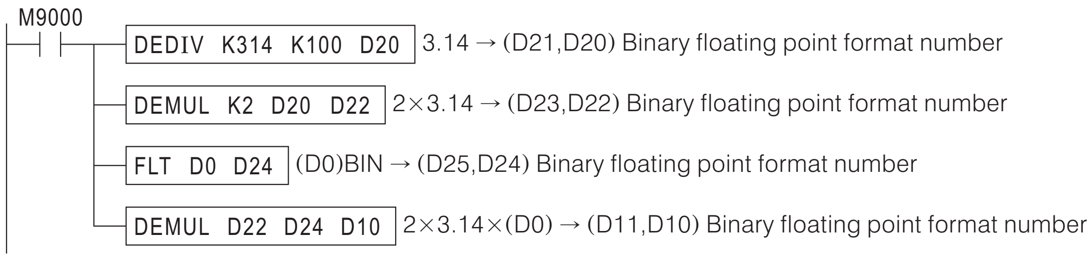


- When X1= "ON", performs the convert operation from the content value of 32 bits registers (D2,D3) (which is a BIN integer) to a binary floating point number, and copies the converted result to the destination devices (D13,D12).

- It is not necessary to use this instruction for constant K or H at floating calculation, because the constant will convert to binary floating point format automatically when the floating calculation is operation.
- A floating point number will occupy two consecutive registers, the format of a floating point number storage in registers, please refer to Section 2-12 "Numerical System".
- Floating point calculation example:
  Use a PLC and FLT instruction to do calculate

  2×3.14×(D0) ⟶ (D11,D10)

  BIN integer — Binary floating point format number



MEMO

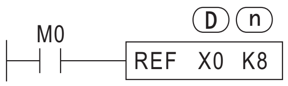
6-7 High Speed Processing Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
50		REF	P	I/O Refresh	○	○	○
51		REFF	P	I/O Refresh and Filter Adjust	○	○	
52		MTR		Input Matrix	○	○	
53	D	HSCS		High Speed Counter Set	○	○	○
54	D	HSCR		High Speed Counter Reset	○	○	○
55	D	HSZ		High Speed Counter Zone compare	○	○	
56		SPD		Speed Detection	○	○	○
57	D	PLSY		Pulse Y output	○	○	○
58		PWM		Pulse Width Modulation	○	○	○
59	D	PLSR		Pulse ramp		○	○

FNC 50 REF	P	REFP (D) (n)	I/O Refresh	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D	○	○														
n															○	

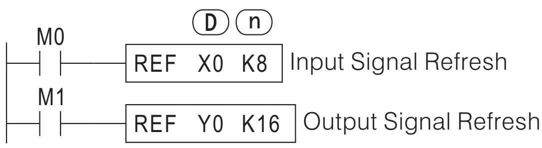
- D should always designate the device with its last digit of "0" (zero), e.g. X20, X30, Y20, Y30, etc.
- For M series, n=8 ~ 512 and n should always be a multiple of "8".
- For VB2 series, n=8 ~ 256 and n should always be a multiple of "8".
- For VB1 series, n=8 ~ 128 and n should always be a multiple of "8".
- For VB0 series, n=8 ~ 64 and n should always be a multiple of "8".
- For VH series, n=8 or 16.



D : The head address of I/O refresh device

n : The number of I/O refresh devices

- Before PLC performs STEP0 instructions, CPU will read "ON"/ "OFF" status of all input ends once and store them in the data memory. Until the END instruction is executed, all "ON"/ "OFF" status of output signals will be sent to output ends to drive external loadings. This instruction is necessary when we desire to read "ON"/ "OFF" status of the input (X) during the execution of the program or deliver the operation result to the output (Y) immediate.



- When M0="ON", the input signal status of X0 ~ X7 will be reloaded to PLC's status data memory. PLC can immediately read the status of X0 ~ X7 while performing this instruction, but the delay (approximately 10ms) on the input contact still remains.
- When M1="ON", the output signal status of Y0 ~ Y17 will be resent to output end contacts from PLC's status data memory. PLC can immediately send the status of Y0 ~ Y17 while performing this instruction, but the delay (by the relay, approximately 10ms) on the output contact still remains.
- D should always designates its last digit as "0" (zero), ex. X0, X10, X20, Y0, Y10, etc. n Should always be a multiple of "8". Any default value exceeding this range will be regarded as an error.

Series	**The range of X for D**	**The range of Y for D**	**The range of n**
M	X0 ~ X777, total 512 points	Y0 ~ Y777, total 512 points	n=8 ~ 512
VB2	X0 ~ X377, total 256 points	Y0 ~ Y377, total 256 points	n=8 ~ 256
VB1	X0 ~ X177, total 128 points	Y0 ~ Y177, total 128 points	n=8 ~ 128
VB0	X0 ~ X77, total 64 points	Y0 ~ Y77, total 64 points	n=8 ~ 64
VH	X0 ~ X17, total 16 points	Y0 ~ Y17, total 16 points	n=8 or 16

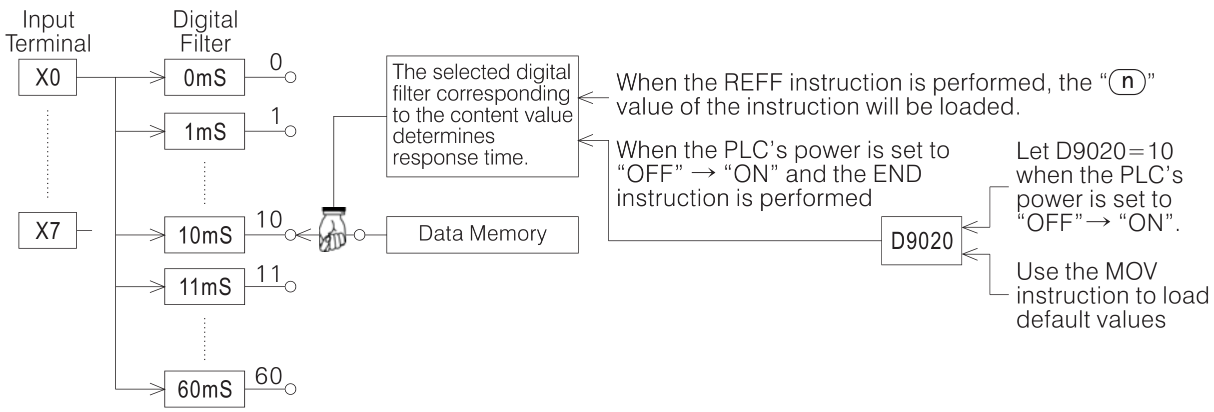
- Use the REF instruction in interrupt subroutines frequently to acquire real-time input/output status.

	FNC 51 REFF	P	REFFP n	I/O Refresh and Filter Adjust	M	VB	VH
					○	○	

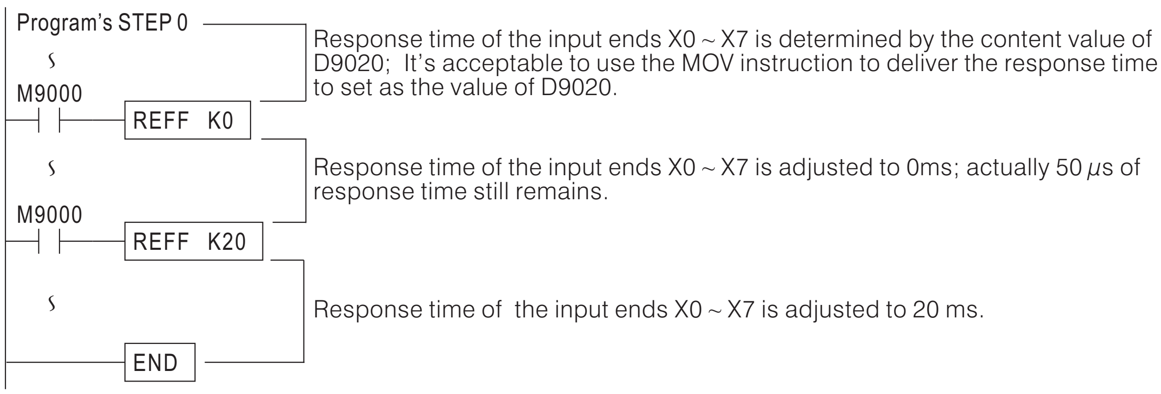
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
n															○	
● n=0 ~ 60																

n : the setting for response time (unit = ms)

- When X20="ON", response time for external input end X0 ~ X7 will be changed into 1ms and the "ON" / "OFF" status of X0 ~ X7 will be reloaded into data memory.
- To avoid noise intervention, there will always be a filter with response time approximately 10ms on the PLC's input end to filter out noise; Therefore, if to capture a input signal which with its pulse width less than 10ms, then it will be failed.
- Input contacts of X0 ~ X7 have been equipped with filters on which we can use the REFF instruction to adjust response time. The following figure shows the input configuration of X0 ~ X7:



- As shown in the figure above, the input terminals X0 ~ X7 have built-in digital filters with 0 ~ 60 ms. The rules determining response time of the input contacts X0 ~ X7 are described as follows:
  ①When the PLC's power is set to "OFF"→ "ON", the content value of D9020 will be set to 10 and response time will be set to 10ms.
  ②It's acceptable to use the MOV instruction to load the default value to D9020 and to adjust response time.
  ③Use the REFF instruction to adjust response time during the program execution.

- 

- When the interrupt function, the high speed counter or the SPD (FNC56) instruction is used in the program, response time of the corresponding input terminal will automatically adjusted to 50 μs.

	FNC 52 MTR		MTR Ⓢ Ⓓ1 Ⓓ2 ⓝ	Input Matrix	M	VB	VH
					○	○	

Operand	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○															
D1		○														
D2	○	○	○													
n															○	

- S should always designate an X with its last digit of "0" (occupies consecutive 8 points).
- D_1 should always designate a Y with its last digit of "0".
- D_2 should always designate a Y, M or S with its last digit of "0".
- n=2 ~ 8

```
       Ⓢ   Ⓓ1  Ⓓ2  ⓝ
X0
─┤├── MTR X20 Y20 M0 K2
```

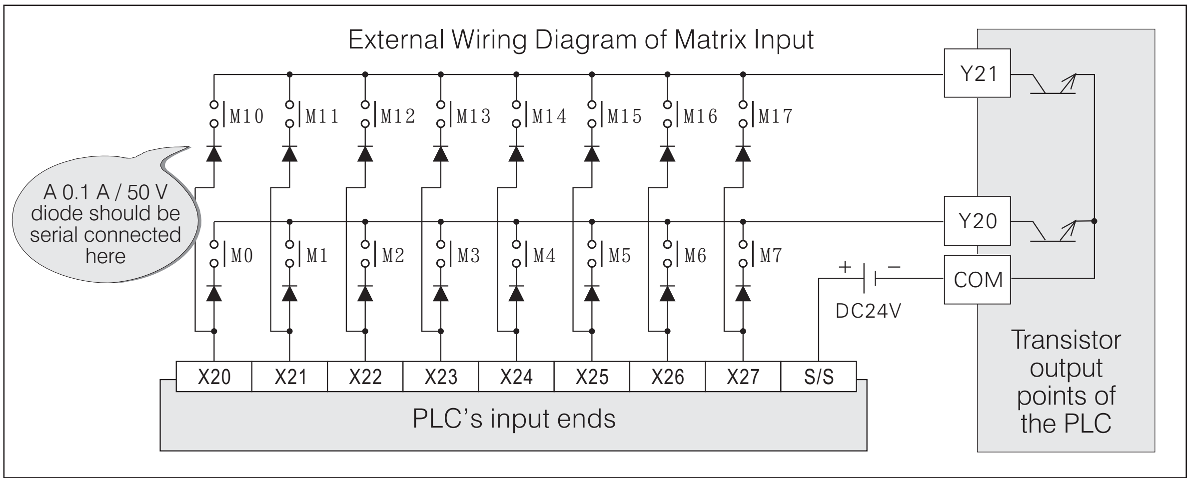
S : the head point for the matrix scan input

D_1: the head point for the matrix scan output

D_2: the head point of the matrix-table (the scan Storage internal coils)

n : number of array rows of the matrix scan

- This instruction reads status through the matrix scan: 8×ⓝ of external "ON"/ "OFF" status from 8 consecutive input ends which are headed with Ⓢ and ⓝ output ends are headed with Ⓓ1. This matrix scan reads the "ON"/ "OFF" status and reflects on the internal coils headed with Ⓓ2.

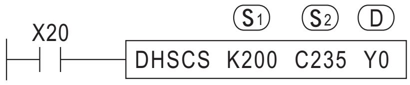


- From the diagram above, X20 ~ X27 and Y20 ~ Y21 constitute two rows array of the matrix input circuit. When X0="ON", the instruction is ready for execution and 16 "ON"/ "OFF" status of (8×2 matrix) array will be read and stored in internal coils of M0 ~ M7 and M10 ~ M17.
- When X0= "OFF", the instruction disables and the status of M0 ~ M7 and M10 ~ M17 remains.
- Using the MTR instruction to read one row of external switches array will takes two scan times. If a scan time is less than 10ms, then reads the status in one row of the array which will takes 20ms to read the status of external "ON"/ "OFF". Maximally, this instruction can connect 8 rows of external switches array. Reading 64 (8×8=64) external switches once will take 16 scan times or 160ms. Therefore, the coordination between external switches response rate and the loading time of the instruction should be considered when this instruction is used.
- The instruction's conditional contacts use M9000 (permanently "ON", "a" contacts) frequently.
- When this instruction performs a scan cycle each time, it will let the Execution Completed Flag M9029= "ON" for one scan time.
- The MTR instruction can be used once during the program.
- This instruction is only recommended for use with transistor output modules.

D	FNC 53 HSCS		DHSCS (S1) (S2) (D)	High Speed Counter Set	M	VB	VH
					○	○	○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2										○						○
D		○	○	○												○

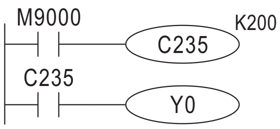
- S_2=C235 ~ C255
- D can also designate I0□0,□=1 ~ 6



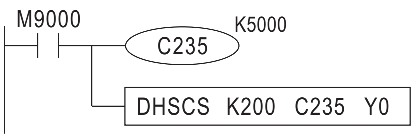
S1 : Compare value
S2 : No. of the selected high speed counter
D : Compare result

- The DHSCS instruction is used to give immediate outputs of High Speed Counter (HSC). HSC receives its high speed counter/pulse of corresponding terminals by using the interrupt input function (for more detailed instruction on HSC, please refer to Section 2-7 "High Speed Counter"). When a HSC is selected by the DHSCS instruction, the current value of HSC(S2) changes (increased/ decreased by "1"), the DHSCS instruction will immediately perform the "Compare" operation. When the current value of the HSC is equal to the Compare value (which is selected by(S1)), the device status of(S2)will turn "ON" and then remain the same status ("ON") even if the Compare result becomes unequal. Generally,(D) in this instruction is designated to an output coil Y. When an output coil Y has been designated by(D), the status "ON" will be carry out immediately to the output terminals.
- When this HSCS instruction is used in the VH series PLC and designated(D)to output coils Y, only the output points Y0 ~ Y17 are allowed.
- As the example above, the DHSCS will be enabled when X20="ON". When the current value of C235 changes from 199 to 200 or from 201 to 200, the status of Y0="ON". At the time the status will be sent to output end, and also the status "ON" remains.

Common Output V.S. DHSCS Instruction Output



The timing when the external output end of Y0 is driven, which is affected by the PLC scan time.

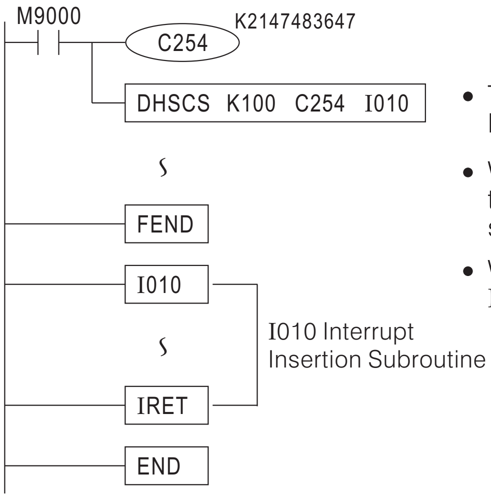


By the function of interrupt, the status of Y0 is immediately output to external output end, irrespective of the PLC scan time. Please notice that: There is output delay of the relay/transistor at the output end.

Note:

- This instruction is a 32-bit instruction; DHSCS should be always entered when the instruction is input.
- There's no limitation on the times used of these instructions DHSCS, DHSCR and DHSZ; However, the total of these instructions performed at the same time should not exceed "6".
- Both the output contacts of High speed counter and Compare output of DHSCS, DHSCR or DHSZ are performed when there is a counter input. The Compare action shall not be performed if the current value of High speed counter is changed by transferring instructions, because there is no counter input signals. Therefore, no Compare output occurs.

High Speed Counter Interrupt



- The device Ⓓ of DHSCS instruction can also designate the Pointer of High speed counter, I0□0, □=1 ~ 6.
- When the current value of C254=100, CPU will jump to the interrupt Pointer I010 to perform the interrupt subroutine.
- When M9059="ON", the High speed counter interrupt of I010 ~ I060 will be blocked.

D	FNC 54 HSCR		DHSCR (S1) (S2) (D)	High Speed Counter Reset	M ○	VB ○	VH ○

Operand	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2										○						○
D		○	○	○												○

- S_2=C235 ~ C255
- D also available to designate the same High speed counter ID number as S_2 but only when D=S_2.

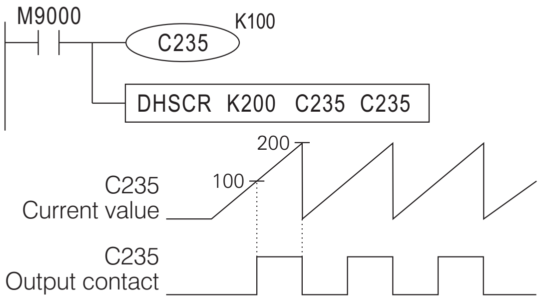
S_1 : Selected compare value

S_2 : Selected high speed counter ID number

D : Use compare result to reset the destination

- The DHSCR instruction is used to give immediate outputs of High speed counter (HSC).
- When this HSCR instruction is used in the VH series PLC and designated (D) to output coils Y, only the output points Y0 ~ Y17 are allowed.
- When X20="ON" and the current value of C235 changes from 199 to 200 or from 201 to 200, the status of Y0="OFF". At the time the status will be sent to output end, and also the status "OFF" remains.

Automatic Reset



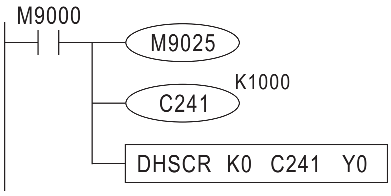
- When (S2) and (D) of the DHSCR instruction designate the same ID number of High speed counter, the counter will perform the self Reset action.
- When the current value of C235=200, C235 will be reset, its current value will be reset to "0", and the output contact will become "OFF".

Note:

- This instruction is a 32-bit instruction; DHSCR should be always entered when the instruction is input.
- There's no limitation on the times used of these instructions DHSCS, DHSCR and DHSZ; However, the total of these instructions performed at the same time should not exceed "6".
- Both the output contacts of High speed counter and Compare output of DHSCS, DHSCR or DHSZ are performed when there is a counter input. The Compare action shall not be performed if the current value of High speed counter is changed by transferring instructions, because there is no counter input signals. Therefore, no Compare output occurs.

About Special Coil M9025

- Some high speed counters have external reset terminals. When the external reset terminal is "ON", the current value of the corresponding high speed counter will be reset to "0" and the output contact will become "OFF". Let M9025="ON", if you desire the reset action to drive external outputs immediately. The following is a sample program.



- X1 is the external reset input terminal of C241.
- When X1="ON", the current value of C241 will be reset to "0", the output contact will become "OFF", and the DHSCR instruction will be performed and Y0 will be reset to "OFF".
- If M9025= "OFF" and X1="ON", the current value of C241 will be reset to "0", the output contact will become "OFF", and the DHSCR instruction will not be performed and the status of Y0 will remain the same.

D	FNC 55 HSZ		DHSZ (S1) (S2) (S) (D)	High Speed Counter Zone compare	M	VB	VH
					○	○	

Operation 1: High Speed Counter Current Value Againsts To a Specified Range

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○
S										○						○
D		○	○	○												○
• S=C235 ~ C255																
• D occupies 3 consecutive points, if D is designated to a Y, then D shall be Y□□0 ~ Y□□5																

```
                 (S1)  (S2)  (S)  (D)
 M9000
─┤ ├──────┤ DHSZ  K100  K200  C255  Y20 │
```

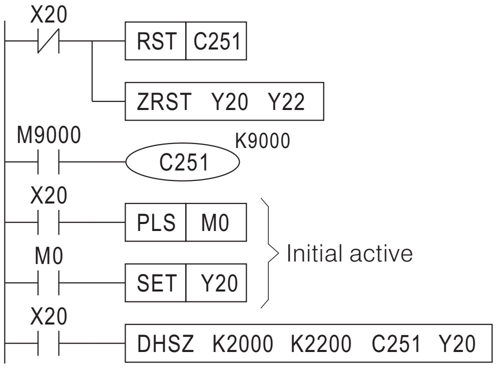
S1 : Lower limit of Zone Compare
S2 : Upper limit of Zone Compare
S : High speed counter ID number
D : Compare Result

- All of the counting value and result outputs of this instruction are processed with interrupt insertion, Y20 ~ Y22 will immediately output irrespective of scan time. Results of Zone comparison are shown as follows:
  When K100 > the current value of C255, then Y20 = "ON".
  When K100 ≤ the current value of C255 ≤ K200, then Y21 = "ON".
  When K200 < the current value of C255, then Y22 = "ON".
- If (S1) > (S2), then the value of (S1) will become both of the Upper/Lower limits to compare with (S).

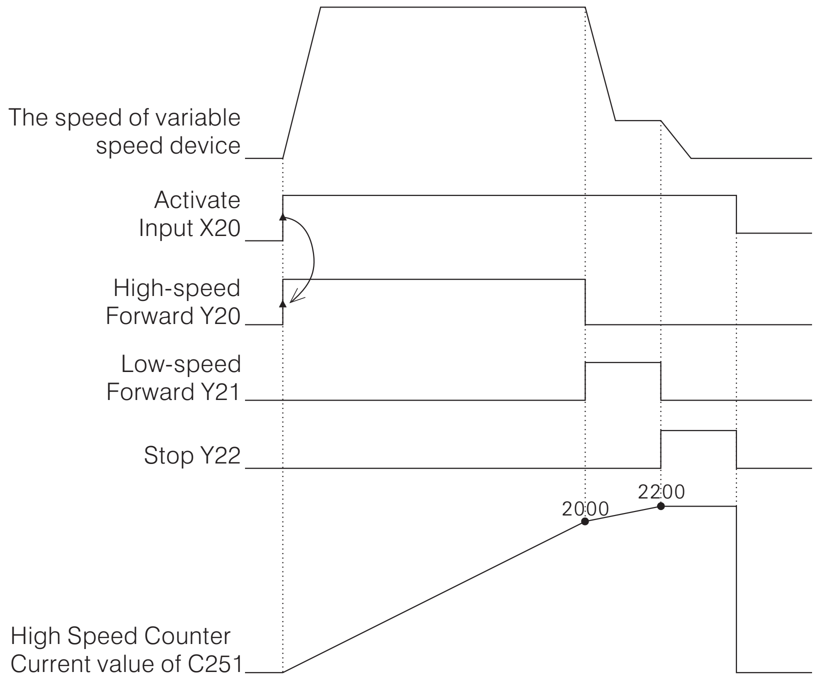
Notes for all modes (operation 1~3):

- This instruction is a 32-bit instruction; DHSZ should be always entered when the instruction is input.
- There's no limitation on the times used of these instructions DHSCS, DHSCR and DHSZ; However, the total of these instructions performed at the same time should not exceed "6".
- Both the output contacts of High speed counter and Compare output of DHSCS, DHSCR or DHSZ are performed when there is a counter input. The Compare action shall not be performed if the current value of High speed counter is changed by transferring instructions, because there is no counter input signals. Therefore, no Compare output occurs.
- If (D) of the DHSZ instruction is designated to a Y, the assigned ID number should be Y□□0 ~Y□□5 , rather than Y□□6 or Y□□7 (e.g. Y20, Y25 are acceptable while Y26, Y27 are not).

Use the HSZ instruction to perform high/low speed stop control



- C251 is an A/B phase high speed counter, X0 is an A-phase pulse input, and X1 is a B-phase pulse input.
- X20 is a signal for activation.
- The DHSZ will have compare outputs only when there is counting pulse entering into C251. So when X20="OFF"→"ON" (the initial active signal of the left-side program), it will activates the motor operation (let Y20= "ON"). At the very beginning, the motor operation will produces a counting pulse and feedback it to the High speed counter. And then, performs the corresponding Compare results of Y20 ~ Y22.

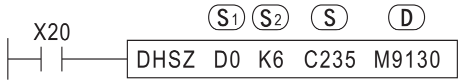


Operation 2: The HSZ Instruction's Multiple Point Compare Mode (When D=M9130)

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	Z	K,H	VZ index
S1											○					○
S2															○	○
S										○						○
D			○													

- S1 occupies 4×S2 consecutive Registers; S2=K1 ~ K128; S=C235 ~ C255;

When the DHSZ instruction's D is designated to M9130, the instruction will perform Compare outputs between the current value of High speed counter and the setting values of comparison data table. In this mode, devices of each operand are shown as follows:



S1 : Head device ID number of the Compare table, designates Data Register D only

S2 : Number of Compare data groups, designates K1 ~ K128 only

S : High speed counter ID No., designates C235 ~ C255 only

D : Mode designation, designates M9130 only

Comparison Data Table

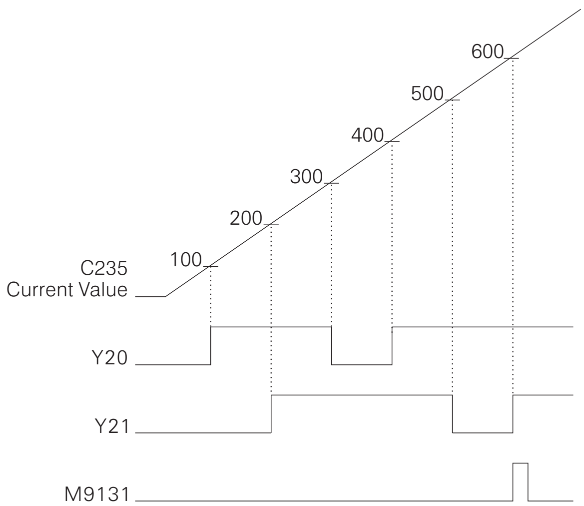
D9130

Comparison Data		Designate "Y" Output Device	SET/RST Designated "Y" Device	Table of Record Number
Lower Reiister	Upper Register			
D0 K100	D1 K0	D2 H20	D3 K1	0
D4 K200	D5 K0	D6 H21	D7 K1	1
D8 K300	D9 K0	D10 H20	D11 K0	2
D12 K400	D13 K0	D14 H20	D15 K1	3
D16 K500	D17 K0	D18 H21	D19 K0	4
D20 K600	D21 K0	D22 H21	D23 K1	5

S1 S2

H20 represents Y20
H21 represents Y21

K0: Reset to "OFF"
K1: Set to "ON"

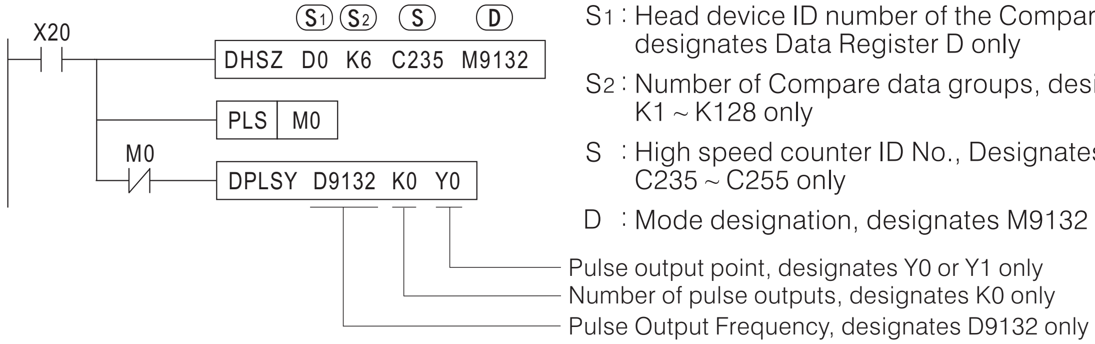


- When X20="ON", the instruction begins to be performed. The Comparison Data Table is processed by one "Record number" at a time. A comparison between the current value of High speed counter C235 (which is designated by) and the content value of Comparison data (D1, D0) in the first group (Record 0) is started. If the comparison is equal, Y20 will be set to "ON" and output immediately. And also, the content value of Record Number D9130 will be increased by "1" (turn into "1").
Then, the current value of C235 begins to be compared to the content value of Comparison data (D5, D4) of second Group (Record 1). If the comparison is also equal, Y21 will be set to "ON" and output immediately. And also, the content value of Record Number D9130 will be increased by "1" (turn into "2"). Then, the subsequent Compare will be proceeded accordingly, until the data compare of the last group is equal while Execution Completed Flag M9131="ON" for a scan time. Later D9130 will be reset to "0" and the data Comparison of first group will be performed again.
- When X20="ON"→ "OFF", the instruction will be disabled, the content of Record Number D9130 will be cleared as "0", but while the output coilis "ON"/ "OFF" status will remain.
- The instruction's Compare operation and output actions are processed by interrupt function.
- The instruction can only be used once in a program.

Operation 3: The Frequency Control Mode Combining HSZ and PLSY（When D=M9132）

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	Z	K,H	VZ index
S1											○					○
S2															○	○
S										○						○
D			○													
● S1 occupies 4×S2 consecutive Registers; S2=K1 ~ K128; S=C235 ~ C255																

When the DHSZ instruction's D is designated to M9132, and assemble with the DPLSY instruction as follows, which performs the function that using the current value of High speed counter to control the PLSY pulse output frequencies.



S1：Head device ID number of the Compare table, designates Data Register D only

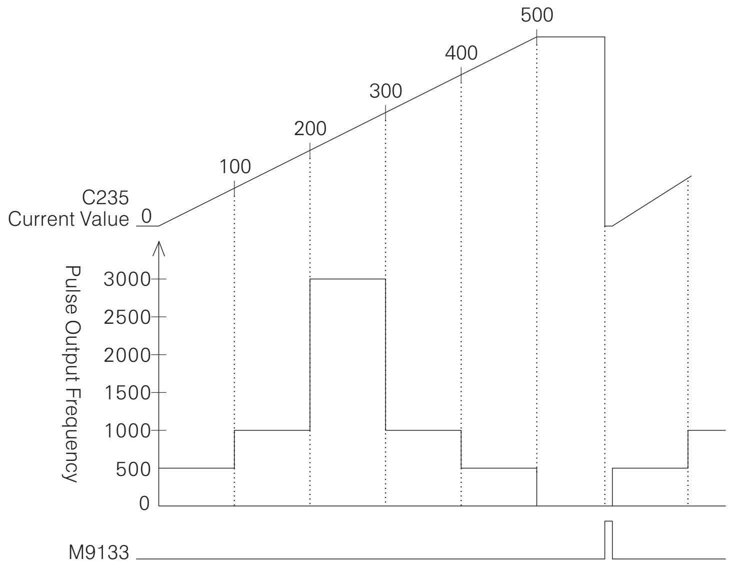
S2：Number of Compare data groups, designates K1 ~ K128 only

S ：High speed counter ID No., Designates C235 ~ C255 only

D ：Mode designation, designates M9132 only

Comparison Data Table

Comparison Data: Lower Reiister	Comparison Data: Upper Register	Pulse Output Frequency 0 ~ 20KHz/ 0 ~ 7KHz	D9131 Table of Record Number
D0 K100	D1 K0	(D3,D2) K500	0
D4 K200	D5 K0	(D7,D6) K1000	1
D8 K300	D9 K0	(D11,D10) K3000	2
D12 K400	D13 K0	(D15,D14) K1000	3
D16 K500	D17 K0	(D19,D18) K500	4
D20 K0	D21 K0	(D23,D22) K0	5



- When X20="ON", the instruction begins to be performed. The Comparison Data Table is processed by one "Record number" at a time. In the beginning, the content value of Table D9131= "0". According to the content value (D3, D2) of the Comparison Data Table, Y0 is assigned to output 500Hz pulses.
  Besides, a comparison between the current value of High speed counter C235 (which is designated by S) and the content value of Comparison data (D1, D0) in the first group (Record 0) is started. If an equal comparison is given, the content value of Record Number D9131 will be increased by "1" (turn into "1"). And then, Y0 outputs 1000Hz pulses according to (D7, D6) of the Comparison Data Table and the current value of C235 begins to be compared to the content value of Comparison data (D5,D4) in the second Group (Record 1). If the comparison is also equal, the content value of D9131 will be increased by "1" (turn into "2"). Then, the subsequent Compare will be proceeded accordingly,until the data compare of the last group is equal while Execution Completed Flag M9133="ON" for a scan time. Later D9131 will be reset to "0" and the data Comparison of first group will be performed again.

- When X20 = “ON”→“OFF”, the instruction will be disabled, the content of Record Number D9131 will be cleared as “0”.
- The instruction can only be used once in a program.
- When this instruction is performed, the PLSY instruction will be not performed until the first scan is finished, and the preparation of the data in the Comparison Data Table must be completed before the first scan to the END instruction.
- D9131: Record Number Counter for the Comparison Data Table

  D9132: In this frequency control mode, it will using the content value of D9131 to select frequency which is the corresponding pulse output frequencies in the Comparison Data Table, and put the selected frequency into (D9133, D9132) registers.

  D9134: In this frequency control mode, it will using the content value of D9131 to select corresponding Comparison Datas in the Comparison Data Table, and put the selected datas into (D9135, D9134) registers.

	FNC 56 SPD		SPD S1 S2 D	Speed Detection	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1	○															○
S_2					○	○	○	○	○	○	○	○		○	○	○
D									○	○	○					○

- S_1=X0 ~ X5
- D occupies 3 consecutive points

S_1 : Exterior pulse input end

S_2 : Time frame of receiving pulses (unit:ms)

D : Detection result

- Within the time frame (which is designate by S2 and unit=ms), calculate the number of pulses coming from the exterior input end (which is designate by S1) and store the result in the register (which is designate by D).
- When X20="ON", D1 begins to accumulate the number of pulses input from the exterior input end X0. After 1000ms of the time frame, store the accumulated results to D0, then clear the current count value of D1 as "0". And then, once again, re-calculate the number of input pulses from X0.
- D2 displaies the Timer's remaining time (unit=ms).
- The main purpose of the instruction is to get the rotation rate of the rotation facility. The rotation rate can get easily from using the content value of D0:

$$N=\frac{60\times(\text{Content value of D0})}{n\times t}\times 10^3\ (\text{rpm})$$

N : Rotation rate

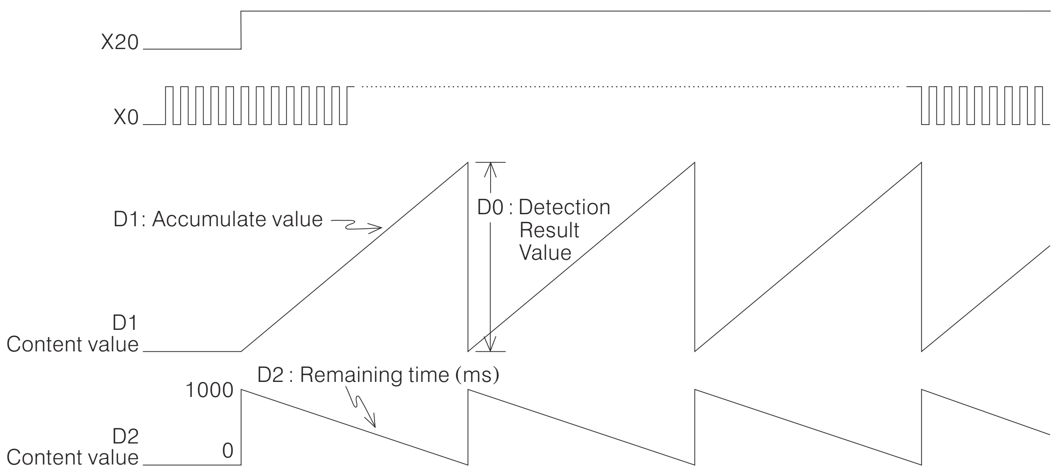
n : Number of pulses generated from a rotation of the rotation facility

t : Content value designated by S2

As in the equation referred above, let n=100, (D0)=3,000, then we will have

$$N=\frac{60\times 3000}{100\times 1000(\text{ms})}\times 10^3=1800\ (\text{rpm})$$

- The exterior input end designated by the instruction's S1 cannot be used as the pulse input terminal or the exterior interrupt insertion signal for High speed counter.
- The max. frequencies of input pulses for the instruction's exterior input end X0 ~ X5 will be 10KHz. But, all the SPD instruction's and High speed counter's total counting frequencies should be no faster than 20KHz.



D	FNC 57 PLSY		⊣⊢ —[D PLSY (S1) (S2) (D)]	Pulse output	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D		○														○
M series	● $S_1=2\sim20000$ ● 16-bit instruction $S_2=0\sim32767$ ● 32-bit instruction $S_2=0\sim2147483647$ ● D=Y0 or Y1															
VB series	● $S_1=2\sim7000$ ● 16-bit instruction $S_2=0\sim32767$ ● 32-bit instruction $S_2=0\sim2147483647$ ● D=Y0 or Y1															
VH series	● $S_1=2\sim7000$ ● 16-bit instruction $S_2=0\sim32767$ ● 32-bit instruction $S_2=0\sim2147483647$ ● D=Y0															

```
                  (S1)  (S2)  (D)
   X20
 ─┤ ├───[PLSY  K500  D100  Y0]
```

S_1 : Pulse output frequency
S_2 : Number of pulse outputs
D : Pulse output point

- When X20="ON", Y0 outputs the specified quantity (D100's content value) of pulses at the 500Hz frequency rate (500 pulses per second).

  (S1) designates the output pulse frequency range. (M series from 2 to 20,000Hz; VB and VH series from 2 to 7,000Hz)

  (S2) designates the number of output pulses

  For a 16-bit instruction, the specified range will be 1 ~ 32,767 pulses.

  A 32-bit instruction, the specified range will be 1 ~ 2,147,483,647 pulses.

  If (S2) is set to "0", the quantity of pulses is unlimited for continuous outputs.

  (D) designates the pulse output point (M and VB series can use Y0 or Y1 only; VH series can use Y0 only).

- The signal pulse is described as having a 50% duty cycle (it is "ON" for 50% of the pulse and consequently "OFF" for the remaining 50%). CPU transfers pulses to output ends immediately by the interrupt mode.
- When the quantity of pulse outputs (which designated by (S2)) are completed, then M9029="ON" for a scan time.
- Special Register D9137 (Upper 16 bits), D9136 (Lower 16 bits) will display the total output pulses of the PLSY instruction.

  Special Register D9141 (Upper 16 bits), D9140 (Lower 16 bits) will display the PLSY instruction's output pulses to Y0.

  Special Register D9143 (Upper 16 bits), D9142 (Lower 16 bits) will display the PLSY instruction's output pulses to Y1.
- When the conditional contact X20 becomes "OFF" during the pulse output, pulse outputs will be stopped and the pulse outpoint (Y0 or Y1) will also turns "OFF"; When X20 becomes "ON" again, the pulse generating will be restored from the first pulse.
- During the instruction execution, it's possible for the instruction to change the content value of (S1) through the program; However, changing (S2) will not become effective until the current operation has been completed.
- The instruction can only be used once in a program.

FNC 58 PWM	PWM (S1) (S2) (D)	Pulse Width Modulation	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D		○														○

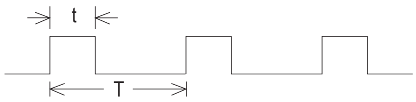
- S_1=0 ~ 32767
- S_2=1 ~ 32767
- D=Y0 or Y1 (VH series D=Y0)

S_1 : Output Pulse “ON” width, t=0 ~ 32,767 ms

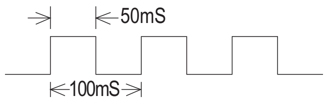
S_2 : Output Pulse cycle distance, T=1 ~ 32767ms

D : Pulse output point

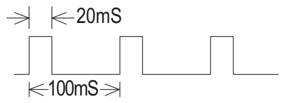
- The PWM instruction is operated as an instruction generating t/T pulse width modulation characteristics of the sequence diagram shown in the right.



- When the conditional contact is “ON”, a pulse with a cycle distance of “T” (designated by (S2)) and the “ON” pulse width of “t” (designated by (S1)) will be output at the output point which designated by (D).
- When X20=“ON” and suppose D0=50, then Y0 will output the following pulses



- When X20=“ON” and suppose D0=20, then Y0 will output the following pulses



- If X20 becomes “OFF”, Y0 will also become “OFF”.
- If “t” is larger than “T”, an operation error will occur.
- The PWM instruction will be operated only once in the program.
- The pulse output point specified by the instruction cannot overlap the output point which specified by the PLSY or PLSR instruction.

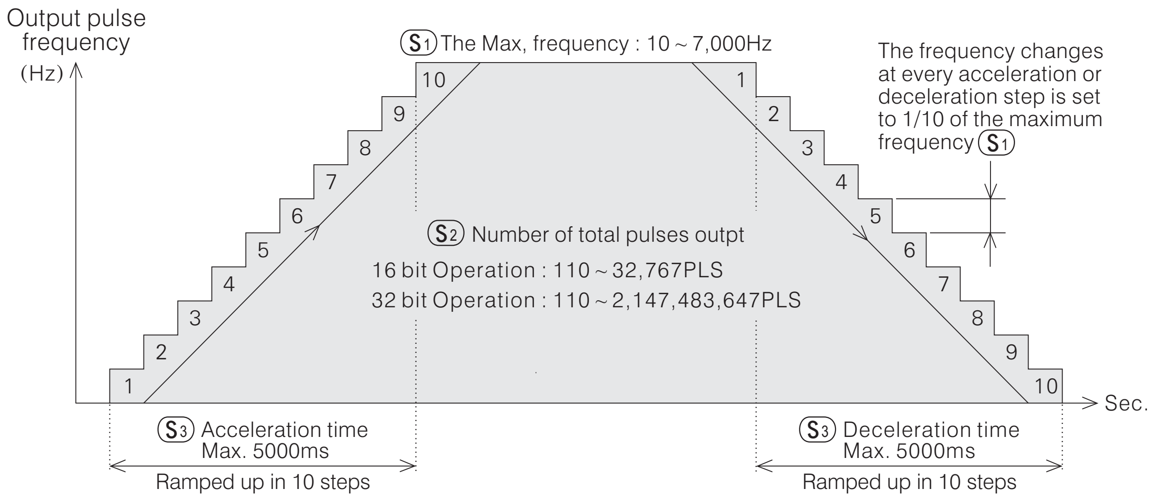
D	FNC 59 PLSR		DPLSR (S1) (S2) (S3) (D)	Pulse ramp	M	VB	VH
						○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
S_3					○	○	○	○	○	○	○	○		○	○	○
D		○														○

● S_1=10 ~ 7000 ● 16-bit instruction S_2=110 ~ 32767 ● 32-bit instruction S_2=110 ~ 2147483647 ● D=Y0 or Y1

```
                    (S1)    (S2)   (S3)  (D)
 X20
─┤ ├───────── PLSR  K1000  D100  K1000  Y0
```

S_1 : Maximun pulse output frequency
S_2 : Number of total pulses output
S_3 : Ramp time for acceleration or deceleration
D : Pulse output point



- When X20= "ON", Y0 outputs the specified quantity (D100's content value) of pulses, the previous diagram is for showing the development of output frequency.

  (S1) designates the maximun output pulse frequency.
  It may use the frequency range from 10 to 7,000Hz, and also the frequency should be set to a multiple of 10.

  (S2) designates the number of output pulses.
  For a 16-bit instruction, the specified range will be 110 ~ 32,767 pulses;
  For a 32-bit instruction, the specified range will be 110 ~ 2,147,483,647 pulses.

  (S3) designates the ramp time for acceleration or deceleration. (unit=ms)

  The available range is: $\frac{100{,}000}{S_1} \leq S_3 \leq 5{,}000$

  If set the value of S_3 is less than $\frac{100{,}000}{S_1}$, the error range of the acceleration/deceleration steps' timing become larger.

  And also, please set the value of (S3) more than 10 times of the maximum program scan time (the content value of D9012). If the setting is less than this, then the timing of the acceleration/deceleration steps become uneven.

  (D) designates the pulse output device is limited to Y0 or Y1 only and the output point should be transistor type.

- This instruction may use the range of output frequency is from 10 to 7,000Hz. When the frequencies of the maximum output pulse or the acceleration/deceleration steps are exceeded the range, it will automatically adjust the frequencies to this range.
- When the quantity of pulse outputs (which designated by(S2)) are completed, then M9029="ON" for a scan time.
- Special Registers D9137 (Upper 16 bits) and D9136 (Lower 16 bits) will display the total output pulses of the PLSY and PLSR instructions.

  Special Registers D91341 (Upper 16 bits) and D9140 (Lower 16 bits) will display the PLSY and PLSR instructions output pulses to Y0.

  Special Registers D9143 (Upper 16 bits) and D9142 (Lower 16 bits) will display the PLSY and PLSR instructions output pulses to Y1.

  The content value of Special Registers above can use the instruction DMOV K0 D91□□to reset it.
- When the conditional contact X20 becomes "OFF" during the pulse output, pulse outputs will be stopped and the pulse output point (Y0 or Y1) will also turns "OFF"; When X20 becomes "ON" again, the pulse generating will be restored from the first pulse.
- During the instruction execution, to change any parameter in this instruction is useless.
- The instruction can only be used once in a program.
- The Y0 and Y1 output points which are driven by PLSY or PLSR instruction can not output pulse at the same time.

MEMO

6-8 Handy Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
61	D	SER	P	Search	○	○	
62	D	ABSD		Absolute Drum Sequencer	○	○	○
63		INCD		Incremental Drum Sequencer	○	○	○
64		TTMR		Teaching Timer	○	○	
65		STMR		Special Timer	○	○	
66		ALT	P	Alternate state	○	○	○
67		RAMP		Ramp variable value	○	○	○
69		SORT		Sort tabulated data	○	○	

D	FNC 61 SER	P	DSERP S1 S2 D n	Search	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○					○
S2					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○					○
n											○				○	

- For a 16-bit instruction, n=1 ~ 256
- For a 32-bit instruction, n=1 ~ 128
- D occupies 5 consecutive devices

X20 — SER D0 D10 D20 K10 (S1 S2 D n)

S1 : Head device ID number of a defined data stack to be searched

S2 : Parameter data to be searched

D : Searched result's storage head device ID number

n : The stack length of the searched data

- The data stack is assigned by "n" consecutive devices which headed with S1. Compare the content value of the device specified by S2 to each device in the data stack, and store the comparison result into 5 consecutive devices headed with D.
- For a search data stack formed by D0 ~ D9. When X20= "ON", compare D10 with D0~D9 and store the result into D20 ~ D24. (Assune the content value of parameter D10=100.)

The result of the search

Data Position Number	Data Stack for Searching	Content Value of D0 ~ D9	Compared Data	Comparison Data
0	S1 D0	100	S2 D10 100	Equal value
1	D1	120		
2	D2	100		Equal Value
3	D3	85		
4	D4	125		
5	D5	60		Min. Value
6	D6	100		Equal Value
7	D7	95		
8	D8	100		Equal Value
9	D9	210		Max. Value

(n = data positions 0 ~ 9)

Result Storage Device	Content Value	Description
D D20	4	Total number of the equal comparison result
D21	0	Data position number of the first equal value
D22	8	Data position number of the last equal value
D23	5	The Min. value data position number
D24	9	The Max. value data position number

- D will record the larger data position number when there's more than one minimum or maximum value in the data stack.
- All the content values of D20 ~ D22 will be "0" when there's no equal value.
- For a 32-bit instruction, S1 , S2 and D will designate a 32-bit register while n will designate a 16-bit register.

D	FNC 62 ABSD		DABSD S1 S2 D n	Absolute Drum Sequencer	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○					○
S2										○						○
D		○	○	○												○
n															○	

- When S1 designates K*n*X, K*n*Y, K*n*M and K*n*S, where n of K*n* should be "4" for a 16-bit instruction and should be "8" for a 32-bit instruction; the ID number of X, Y, M and S should be a multiple of "8"
- A 16-bit instruction S2=C0 ~ C199, a 32-bit instruction S2=C200 ~ C255
- n=1~ 64

```
                    S1  S2  D   n
 X20
 |  |----------[ABSD D0  C0  M0  K4]
```

S1 : Head device ID number of the comparison table
S2 : The ID number of the counter
D : Head device ID number of the comparison result
n : Number of comparison section groups

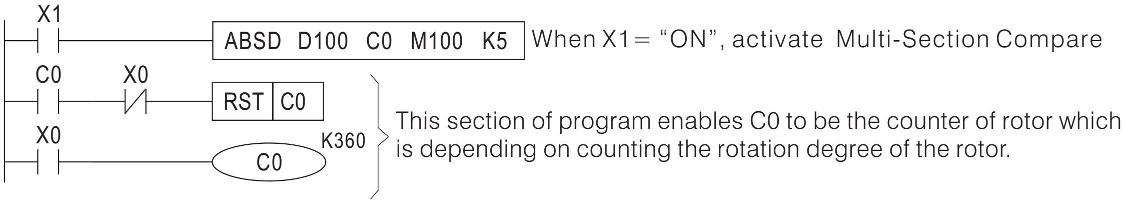
- The instruction is a Multi-Section Compare instruction and generally is operated for multi-section absolute drum sequencer.

	Lower Limit	Upper Limit	Comparison Value	Comparison Result
n	S1 D0=50	D1=200	S2 C0=100	D M0=1
	D2=0	D3=50		M1=0
	D4=80	D5=120		M2=1
	D6=120	D7=300		M3=0

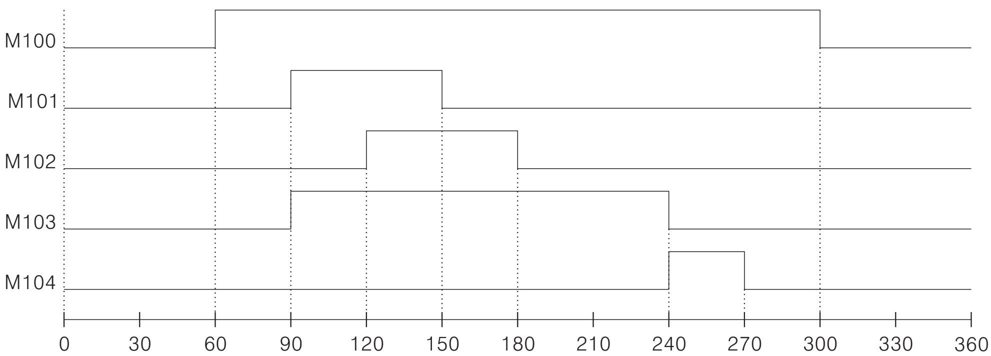
- When X20= "ON", the current value of the selected counter C0 is compared against a user defined data table [(D0, D1), (D2, D3), (D4, D5) and (D6, D7) 4 groups of upper/lower limit], and the results are stored on M0 ~ M3 respectively.
  If [Lower Limit ≤ Comparison Value ≤ Upper Limit], the corresponding output point will be turned "ON"; Otherwise, the comparison value is not placed between Upper Limit and Lower Limit, the corresponding output point will be turned "OFF".
- When X20= "OFF", the status, "ON"/ "OFF", of M0 ~ M3 will remain.

A Program Example

Suppose that a drum-controlled rotor sends a pulse to the input terminal X0 when it rotates by one degree, then the following program will perform the checkup and control actions of the drum degree.



Lower Limit	Upper Limit	Comparison Value	Comparison Result
D100=60	D101=300		M100
D102=90	D103=150		M101
D104=120	D105=180	C0	M102
D106=90	D107=240		M103
D108=240	D109=270		M104



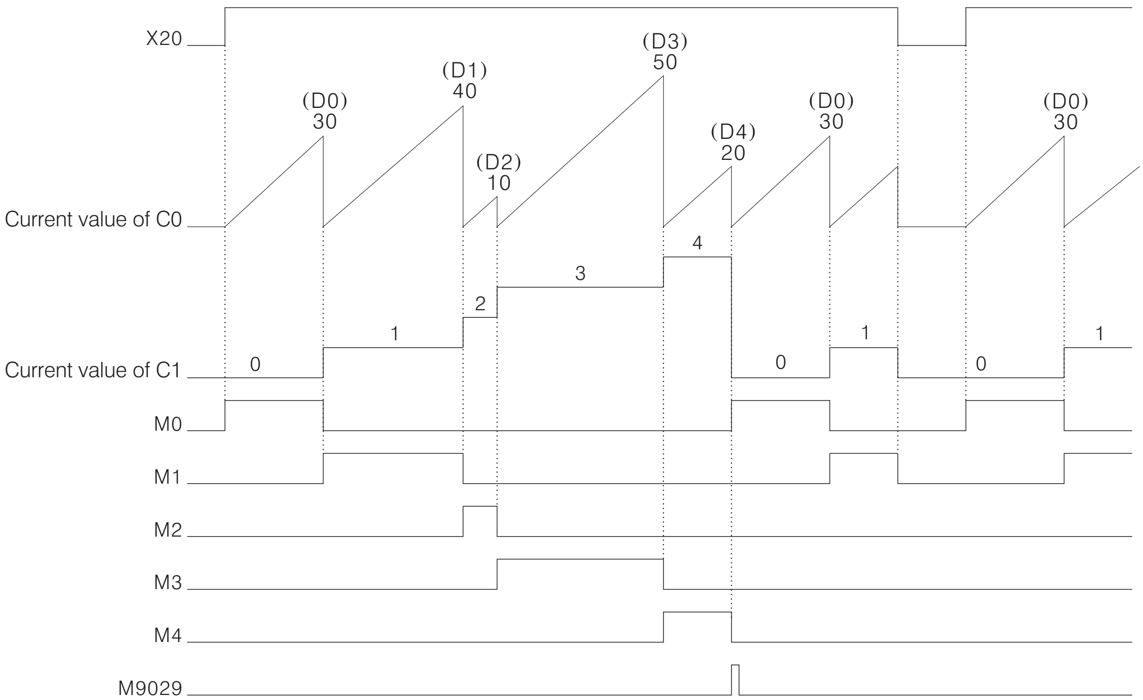
	FNC 63 INCD		INCD (S1) (S2) (D) (n)	Incremental Drum Sequencer	M	VB	VH
					○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○					○
S2										○						○
D		○	○	○												○
n															○	

- When S1 designates K*n*X, K*n*Y, K*n*M and K*n*S, where *n* of K*n* should be "4" and the number of X, Y, M and S should be a multiple of "8"
- S2=C0 ~ C198 • n=1~ 64

S1 : Head device ID number of the comparison table
S2 : ID number of the counter
D : Head device ID number of the comparison result
n : Number of comparison section groups

- The INCD instruction is a multi-section incremental drum-controlled instruction.
- If (S1) designates D0, n=5 and (D) designates M0, then the comparison values are stored in D0 ~ D4, while M0 ~ M4 serve as the outputs. Suppose that (D0)=30, (D1)=40, (D2)=10 and (D4)=20.
- For detailed actions of the INCD instruction, please refer to the following sequence diagram.

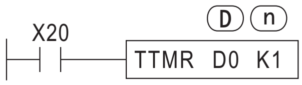


- (S2) will occupies two consecutive ID number counters.
- For a multi-section incremental comparison output, Execution Completed Flag M9029 will turn "ON" for a scan time while a circulation is completed.
- When X20= "ON" → "OFF", the current values of C0 and C1 will be reset to "0" and M0 ~ M4 will be turned "OFF".

	FNC 64 TTMR		TTMR D n	Teaching Timer	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D											○					○
n															○	

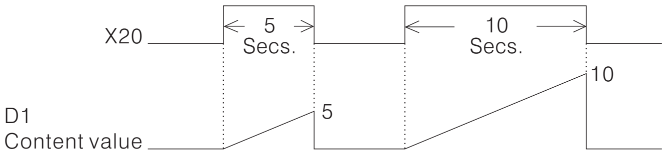
• n=0 ~ 2 • D occupies 2 consecutive registers



D : The ID number of the register which can store the timed data of "ON" duration (ex. From a push button switch)

n : Setting for multiplier

- Suppose X20 is the external push button switch.
- When X20= "ON" (is pressed), the content value of D1 will respond (in secs) to the duration of X20= "ON".
  If X20="ON" for 5 seconds, then D1=5. Because n=1, so D0=50.
  If X20="ON" for 10 seconds, then D1=5. Because n=1, so D0=100.



- The content value of D0 is determined by the content value of D1 and "n"; their correlation is:

n	Content value of D1	Content value of D0	
0	10 (Supposed Value)	10×1=10	When n=0, (D0)=(D1)×1
1		10×10=100	When n=1, (D0)=(D1)×10
2		10×100=1000	When n=2, (D0)=(D1)×100

Accordingly, it is easily to use the content value of D0 become the setting value of a Timer T via a proper setting value of "n".

n=1 can be applied to a 100 ms unit Timer

n=2 can be applied to a 10 ms unit Timer

- When X20= "ON" → "OFF", the current value of D1 will be reset to "0" but the content value of D0 will remain.

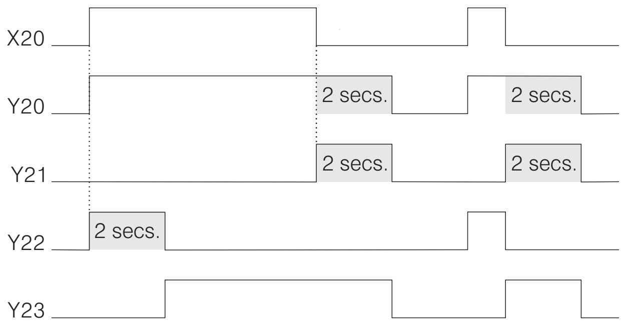
FNC 65 STMR	STMR (S) (m) (D)	Special Timer	M	VB	VH
			○	○	

Operand	Devices																
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S									○							○
m															○	
D		○	○	○												○

- S=T0~T199 • m= 1~32767 • D occupies 4 consecutive devices

S : ID number of designated Timer
m : Setting value of the Timer (unit=100ms)
D : Head ID number of the output device

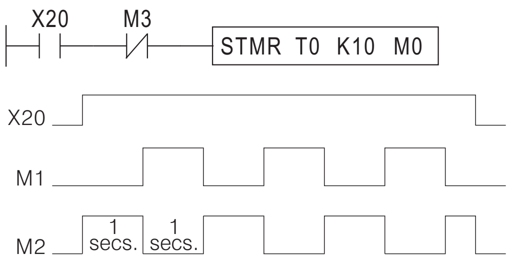
- The STMR instruction is operated exclusively to produce an Off-delay, a trigger and a flashing circuit.
- When X20= "ON", the STMR instruction starts to be performed. As (m) =20, the T0 become a 2 seconds setting value Timer.



- Y20 is an Off-delay output.
- When Y21 is an input signal turned from "ON" to "OFF", a trigger for one shot timer output will be enabled.
- Y22 and Y23 are designed for output signals exclusively composing the flashing circuit. The following example is a practical approach for the flashing circuit.

- In the program, do not reuse the Timer ID number which has been used by this instruction before.

Flashing Circuit



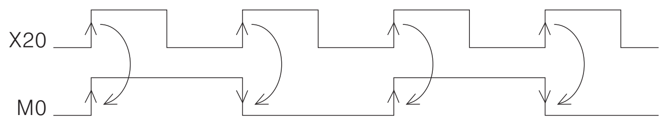
- Perform a serial link "b" Contact of M3 after X20, then M1 and M2 will perform the flashing circuit.

	FNC 66 ALT	P	ALTP D	Alternate State	M	VB	VH
					○	○	○

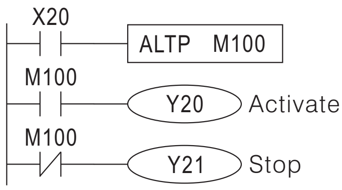
Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D		○	○	○												○

D : Destination Device

- When X20= "OFF " → "ON" for the first time, M0= "ON"; while X20= "OFF" → "ON" for the second time, M0= "OFF". Hence when X20= "OFF" → "ON" for an odd time, M0= "ON", and for an even time, M0= "OFF".

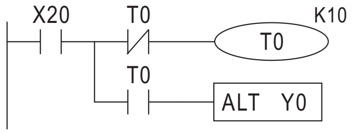


Activate and stop by operating the same push button control

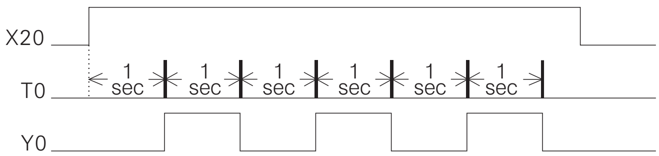


- When X20= "OFF" → "ON" for the first time, M100= "ON"; Y20= "ON", Y21= "OFF" and the operation will be activated.
- When X20= "OFF" → "ON" for the second time, M100= "OFF"; Y20= "OFF", Y21= "ON" and the operation will be stopped.

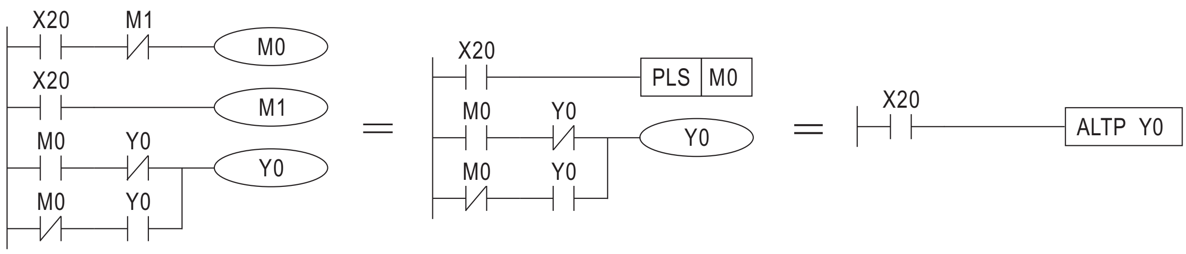
Produce the flashing state



- When X20= "ON", T0 will produce a pulse every other second.
- Every time when T0 produces a pulse, the state of Y0 will be changed once.

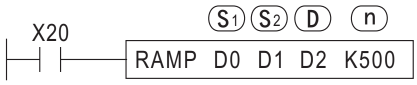


Traditional circuit : single-"ON"/double- "OFF"



FNC 67 RAMP	RAMP (S1) (S2) (D) (n)	Ramp Variable Value	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					○
D											○					○
n															○	
● n= 1 ~ 32767 ● D occupies 2 consecutive registers																



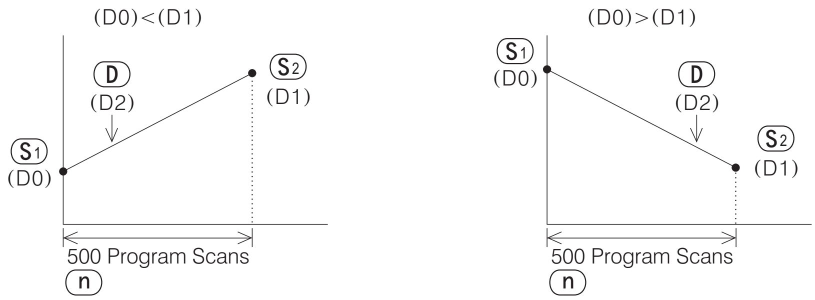
S1 : Initial value of the ramp signal

S2 : Destination value of the ramp signal

D : Value of the "journey" of the ramp signal

n : Specified number of Scan Times of the "journey" takes

- Write the initial point value of the ramp signal to D0 and write the destination value of the ramp signal to D1.
  When X20= "ON" and (D0) < (D1), then the current value of D2 will increase from the setting value of D0 to that of D1.
  When X20= "ON" and (D0) > (D1), then the current value of D2 will decrease from the setting value of D0 to that of D1.
  It takes 500 of PLC's Scan Times for the current value shifted from the setting value of D0 to that of D1.



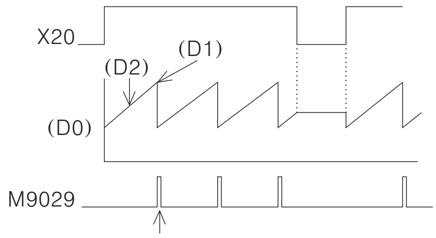
When the instruction is performed, the value of the "journey" of the ramp signal will be reflected on (D2) while the value of the "journey" of the scan times will be reflected on (D3).

- As shown in the diagram above, whether the pointing curve of D2, appears to be in Linear Gradient is closely correlated to the scan time of PLC. Generally PLC does not always take the same scan time. Thus, if in the occasion where the RAMP instruction is applied and it requires Linear Gradient, the interval that the RAMP instruction is performed must be equal each time. In terms of this purpose, it's acceptable to use the constant scan time setting function or the interrupt function. (Please reference to the program examples in next page.)
- When X20= "ON" → "OFF", the instruction will be disabled and D3 will be cleared as "0"; And if X20 is set "ON" again, the instruction will restore.
- When the execution of the instruction is completed, M9029= "ON" and the content value of D2 will be restored to the setting value of D0.
- The instruction can work with the analog output to incorporate the action of the buffered activation/stop.
- If X20= "ON" and PLC turns from STOP to RUN, please clear D3 as "0" (placed at the front end of the program).

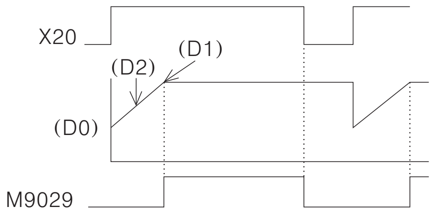
Operation Modes (swapped by Flag M9026)

When the RAMP instruction is performed, the operation mode will change depending on the status of Special Auxiliary Coil M9026.

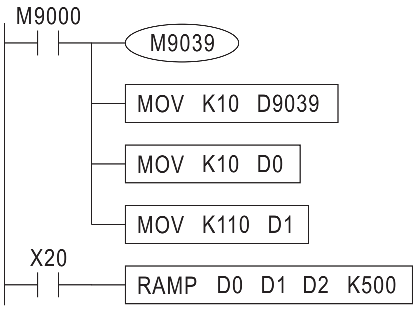
- If M9026 = "OFF", it will generate contiguous ramp signals.(Repeat Mode)



- If M9026 = "ON" , it will generate only one ramp signal. (Hold Mode)



A program model for usage of the constant scan time function



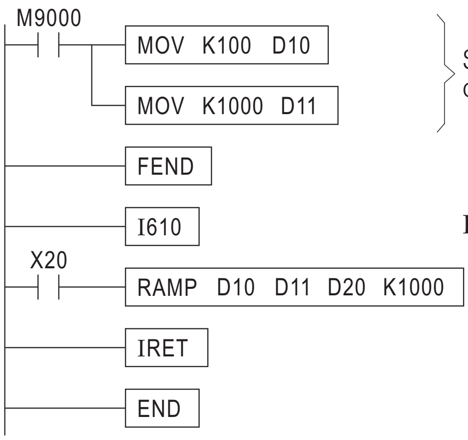
Settle the scan time at 10ms.

Set the initial value of the Ramp signal at 10 and the destination at 110.

Time of the entire journey of the RAMP instruction (the current value of D2 changes from the setting value of D0 to that of D1) will be 10ms × 500=5,000ms=5 Secs.

Since the scan time is settled at 10ms (each scan time is consistent) the ramp signal appears to be in Linear Gradient. However, in the program model referred above, it should be noted that the setting value of the constant scan time is required to be a little larger than the maximum value of the actual scan time. Otherwise, the constant scan time function would be useless. To oversee the D9012 register will get the maximum value of the actual scan time.

A program model for usage of the Time Interrupt function



Set the initial value of the Ramp signal at 100 and the destination at 1000.

I610 is the interrupt pointer of the time interrupt at 10ms.

Time of the entire journey of the RAMP instruction (the current value of D20 changes from the setting value of D10 to that of D11) will be 10ms × 1000=10,000ms=10 Secs.

FNC 69 SORT	SORT (S) (m1) (m2) (D) (n)	Sort Data	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					
m1															○	
m2															○	
D											○					
n											○				○	

● $m_1=1\sim32$ ● $m_2=1\sim6$ ● $n=1\sim m_2$

```
      S  m1 m2  D    n
X20
SORT D0 K5 K4 D100 D200
```

S : Head register ID number of the original data block

m1: Number of data records to be sorted

m2: Number of data fields of each set

D : Head register ID number of the data block where Sort results are stored

n : Reference value of Sort data

Original Data Table
(Start from destination register S)

Data Filed			
1	2	3	4
Student ID	Philology	Mathematics	History
(D0) 1	(D5) 80	(D10) 70	(D15) 75
(D1) 2	(D6) 65	(D11) 70	(D16) 90
(D2) 3	(D7) 90	(D12) 65	(D17) 80
(D3) 4	(D8) 75	(D13) 90	(D18) 65
(D4) 5	(D9) 80	(D14) 85	(D19) 95

m1 (rows) ; m2 (columns)

- The SORT instruction is used to sort several data records (designated by m1). Each may have some data fields (the number of data fields is designated by m2) while "n" is used to assign the n[th] field as the basis for Sort Data. S designates the head register ID number of the original data to be sorted and D designates the head register ID number of the data block where Sort results are stored.
- When X20= "ON", the Sort instruction is performed. This instruction completes the Sort action only after m1 scan cycle(s). When the Sort is completed, the Execution Completed Flag M9029="ON" for a can time and the Sort action will be stopped.
- Both S and D will occupy m1×m2 consecutive register(s)
- The SORT instruction can be used once only in the program.

Sort Data Result Table
(Start from destination register D)
when D200=2

Data Filed			
1	2	3	4
Student ID	Philology	Mathematics	History
(D100) 2	(D105) 65	(D110) 70	(D115) 90
(D101) 4	(D106) 75	(D111) 90	(D116) 65
(D102) 1	(D107) 80	(D112) 70	(D117) 75
(D103) 5	(D108) 80	(D113) 85	(D118) 95
(D104) 3	(D109) 90	(D114) 65	(D119) 80

m1 (rows) ; m2 (columns)

Sort Data Result Table
(Start from destination register D)
when D200=4

Data Filed			
1	2	3	4
Student ID	Philology	Mathematics	History
(D100) 4	(D105) 75	(D110) 90	(D115) 65
(D101) 1	(D106) 80	(D111) 70	(D116) 75
(D102) 3	(D107) 90	(D112) 65	(D117) 80
(D103) 2	(D108) 65	(D113) 70	(D118) 90
(D104) 5	(D109) 80	(D114) 85	(D119) 95

m1 (rows) ; m2 (columns)

MEMO

6-9 External Setting and Display Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
70	D	TKY		Ten Key input	○	○	
71	D	HKY		Hexadecimal Key input	○	○	
72		DSW		Digital Switch (thumbwheel input)	○	○	
73		SEGD	P	Seven Segment Decoder	○	○	○
74		SEGL		Seven Segment with Latch	○	○	
76		ASC		ASCII code Convert	○	○	
77		PR		Print	○	○	
78	D	FROM	P	Read from a special function block	○	○	
79	D	TO	P	Write to a special function block	○	○	

D	FNC 70 TKY		DTKY (S) (D1) (D2)	Ten Key Input	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○												○
D1						○	○	○	○	○	○					○
D2		○	○	○												○

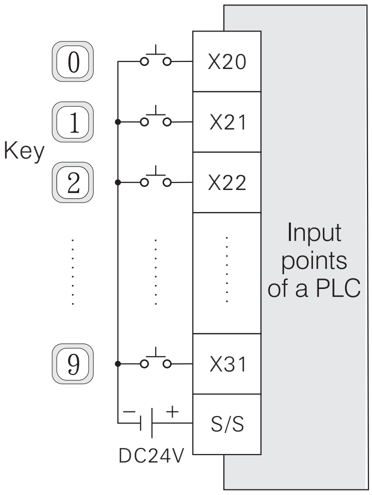
```
 X0         (S) (D1) (D2)
─┤ ├────── TKY  X20  D0  M0
```

S : Initial device of the key input

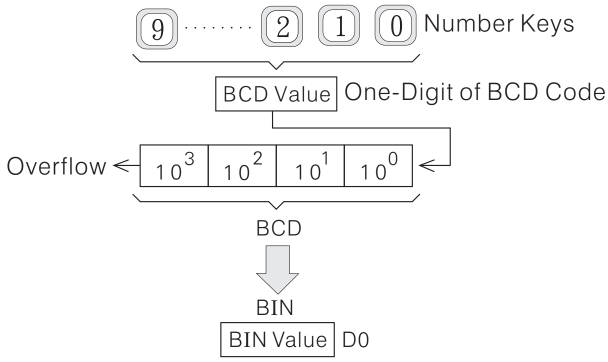
D1 : Place where the key input value is stored

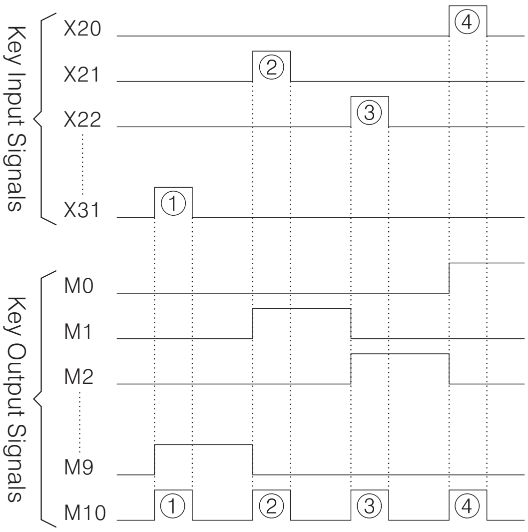
D2 : Initial destination device of the key output signal

- The instruction designates consecutive ten input devices, initiating from (S), which represent decimal numbers 0 ~ 9 in order. These 10 external input devices are connected to 10 keys. Based on the 10 keys pressed is sequence, a four-digit decimal number 0 ~ 9,999 (a 16-bit instruction) or an eight-digit decimal number 0 ~ 99,999,999 (a 32-bit instruction) can be input. And then, the input value will be placed in (D1). Also the instruction uses 11 consecutive devices which starting from (D2) to store the status of the keys.



- As in the left diagram, the input points X20 ~ X31 are received the signals from keys 0 ~ 9. When X0= "ON", the instruction started to be performed. The value input by the number keys is placed in D0 (by format of BIN) and the statuses of the keys are restored in M0 ~ M9 and M10.





- As shown in the left sequence diagram, the number keys and following X20 ~ X31 are input in order, then the result 9,120 is stored in D0.
- When X31 is connected (key #9 is pressed), M9 will turn "ON" and remain "ON", until the next key is pressed (X21= "ON" → M9= "OFF"). The same situation applies to other keys.
- If any of the keys X20 ~ X31 is pressed, M10= "ON" and the devices corresponding to M0 ~ M9 are "ON".
- When the status of X0 "ON" → "OFF", the value of D0 will stay unchanged and M0 ~ M10 will all turn "OFF".

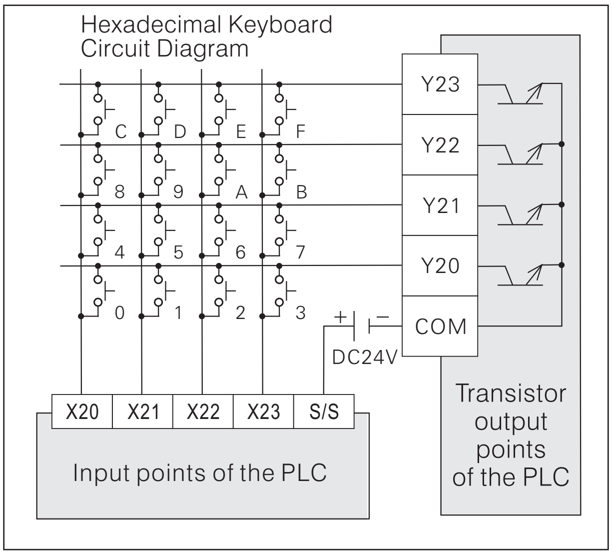
D	FNC 71 HKY		⊣⊢ **D**HKY (S) (D1) (D2) (D3)	Hexadecimal Key Input	M	VB	VH
					○	○	

<table>
<tr><th rowspan="2">Operand</th><th colspan="16">Devices</th></tr>
<tr><th>X</th><th>Y</th><th>M</th><th>S</th><th>KnX</th><th>KnY</th><th>KnM</th><th>KnS</th><th>T</th><th>C</th><th>D</th><th>SD</th><th>P</th><th>V,Z</th><th>K,H</th><th>VZ index</th></tr>
<tr><td>S</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><td></td><td></td><td></td><td>○</td></tr>
<tr><td>D1</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><td></td><td></td><td></td><td>○</td></tr>
<tr><td>D2</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><td></td><td></td><td></td><td>○</td></tr>
<tr><td>D3</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><td></td><td></td><td></td><td>○</td></tr>
</table>

```
        (S)  (D1)  (D2) (D3)
 X0
─┤├──┤ HKY  X20  Y20  D0  M0 │
```

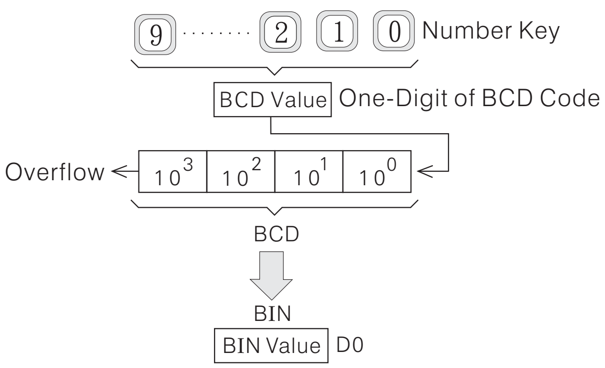
S : Multiplex scanning initial point of the key input
D1 : Multiplex scanning initial point of the key output
D2 : Place where the key input value is stored
D3 : Initial destination device of the key output signal

- The instruction creates the Hexadecimal Key Input by matrix scan of 4 consecutive external input points initiating from (S) and 4 consecutive external output points initiating from (D1). The value input by the number key is stored in (D2). And the instruction uses 8 consecutive devices which starting from (D3) to store the status of the keys.



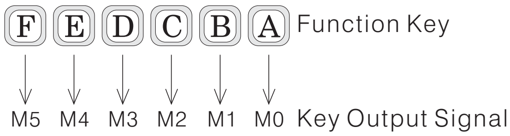
- In the left diagram, the Hexadecimal Keyboard is composed of X20 ~ X23 and Y20 ~ Y23. When X20 = "ON", the instruction started to be performed. The value input by any key is placed in D0 and the status of the key is restored in M0 ~ M7.
- M9029 will turn "ON" for a scan time when the instruction is performed for a scan.
- If there are several keys being pressed at the same time, only the key activated first is effective.
- If the special coil M9167 is already "ON", the HKY instruction can be used for input a hexadecimal value 0 ~ F.
- The HKY instruction can be used once only in the program.
- This instruction is only recommended for use with transistor output modules.

Number Input



- The value input by the number keys is placed in D0 (by format of BIN).
- For a 16-bit instruction, the available maximum input value will be 9,999. If an input value has more than 4 digits, the previous digit(s) will be overflowed.
- For a 32-bit instruction, the available maximum input value will be 99,999,999. If an input value has more than 8 digits, the previous digit(s) will be overflowed.

Number Input



- The A ~ F keys are defined as function keys.
- If a function key is pressed, the corresponding key output signal will turn "ON" and remain the same status, until other function key has been pressed the previous signal will be "ON" → "OFF".

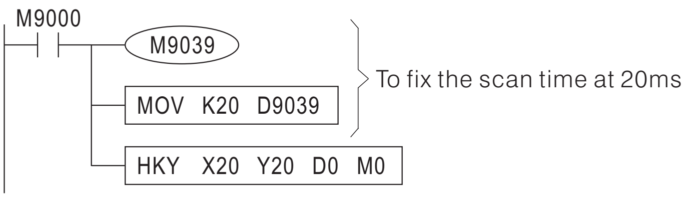
  For example, when A is pressed, M0 will turn and remain "ON". And if F is pressed then, M5 will turn and remain "ON" while M0= "OFF".

Key Output Signal

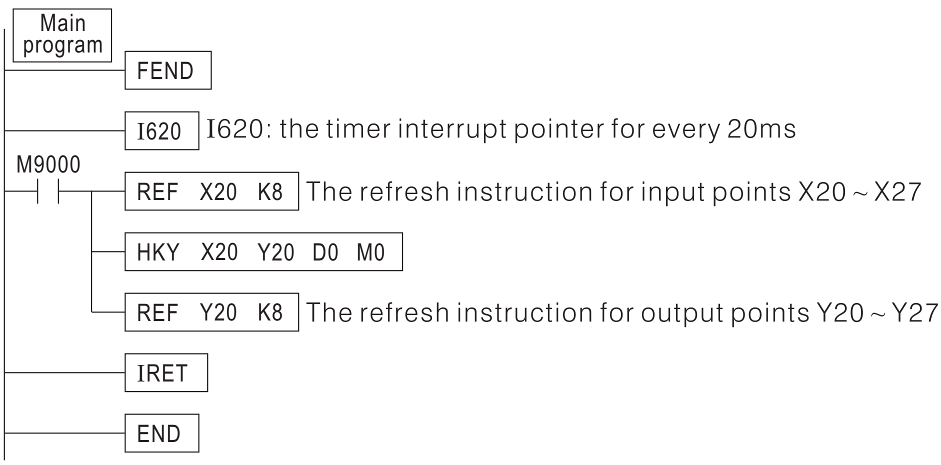
- If the keys A ~ F are pressed, the corresponding key output signals M0 ~ M5 will turn "ON".
- During the period when any one of the function keys A ~ F is pressed, M6= "ON"; And M6= "OFF" when the key is released.
- During the period when any one of the number keys 0 ~ 9 is pressed, M7= "ON"; And M7="OFF" when the key is released.
- When the conditional contact X0= "OFF", the input value will stay unchanged; However, M0 ~ M7 will all turn "OFF".

Notice

- When the instruction is performed, it should take 8 scan times to effectively capture a key. When the program's scan time is too long or too short, it may affect to read the key input signal incorrectly. The solution may be shown as follows:
- If the scan time is too short, this may possibly does not have enough time to take the I/O responses then it will cause to read the input keys incorrect. Please use the constant scan time function to fix the scan time at 20ms.



- If the scan time is too long, this will cause key responses to be slow. Please use the timer interrupt function to fix the scan time of keys at 20ms.



	FNC 72 DSW		DSW (S) (D1) (D2) (n)	Digital Switch Input (Thumbwheel Input) Input	M	VB	VH
					○	○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	○															○
D1		○														○
D2									○	○	○					○
n															○	

• n=1～2

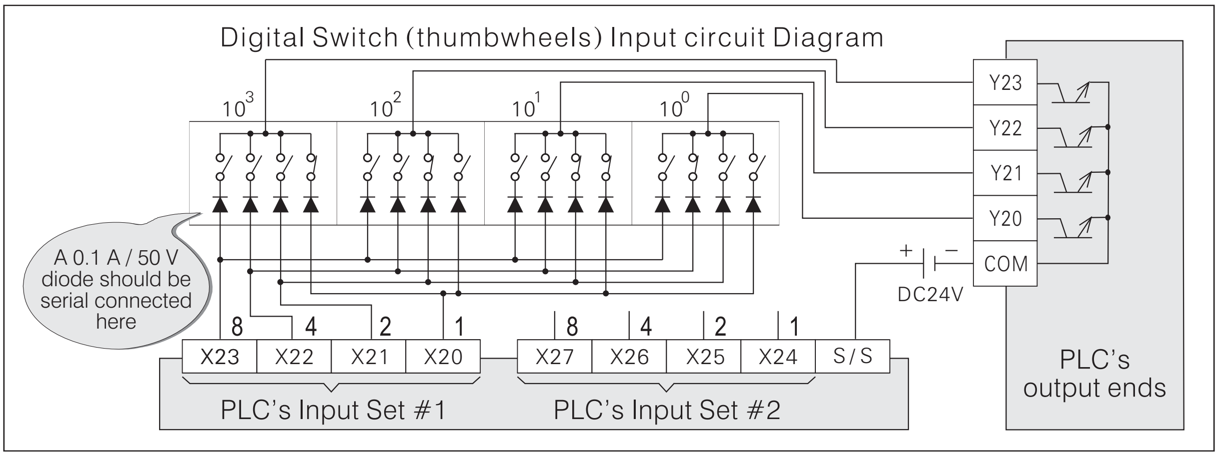
S : Multi-scan the digital switch input of initial point

D1: Multi-scan the digital switch output of initial point

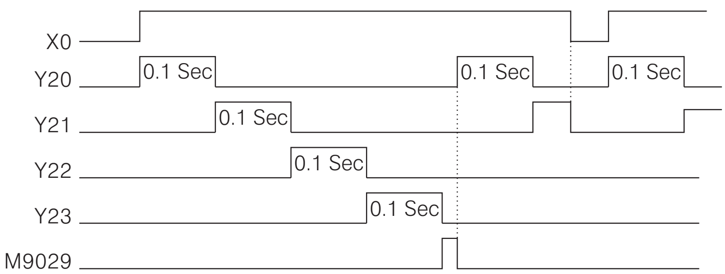
D2: Place the value of the digital switchs

n : Number of digital switch sets connecting

- The instruction scans and reads one set (or two sets) of four-digit thumbwheel digital switches by 4 (or 8) consecutive input points initiating from (S) and 4 consecutive output points initiating from (D1). The value of the digital switch is stored in (D2) to read 1 or 2 sets of four-digit switches is decided by (n).

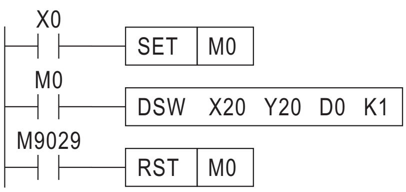


- The diagram shown above is a circuit of a multiplexed thumbwheels switch which is composed by X20～X23 and Y20～Y23. When X0= "ON", the instruction will start to be performed, the value of the thumbwheels will be read and converted into a BIN format then the value will be stored in D0. If (n)=k2 and input points X24~X27 are connected with another set of the thumbwheels then the value of the 2nd set of the thumbwheels will be stored in D1.



- The left diagram is a scan sequence diagram. When X0= "ON", Y20～Y23 will automatically cycle the scan. If each cycle is completed, Execution Completed Flag M9029 will be "ON" for a scan time.
- The instruction can be used only once in the program and recommended to use transistor output unit(s) for the multiplex scan output ends Y20～Y23.

The approach using the relay output unit(s) as the scan output end.



- X0 uses the push-button switch.
- The DSW instruction will read the value of the thumbwheel digital switch once when X0 is pressed once. In the remaining time, no DSW instruction would be performed, nor does any scan action proceed. Therefore, there's no problem even if the relay is used at the scan output end.

	FNC 73 SEGD	P	SEGDP (S) (D)	Seven Segment decoder	M	VB	VH
					○	○	○

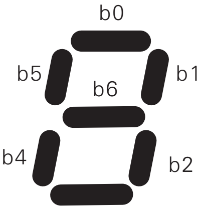
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

```
                (S)  (D)
 X20
─┤ ├───────[SEGD  D0  K2Y20]
```

S : Source device to be decoded

D : Output device after decoded

- When X20="ON", decode the content value (nibble format 0 ~ F) of D0's lower four bits (b3 ~ b0) into a code for a seven-segment display and output it through Y20 ~ Y27.
- The output structure of SEGD is shown in the following table.

(S) Hexadecimal Number	(S) Bit Format	Composition of the seven segment display	(D) b7	b6	b5	b4	b3	b2	b1	b0	Data Displayed
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

FNC 74 SEGL	SEGL (S) (D) (n)	Seven Segment with Latch	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D		○														○
n															○	

- n=0～7

```
X20           (S) (D) (n)
─┤├──────── SEGL  D0  Y20  K0
```

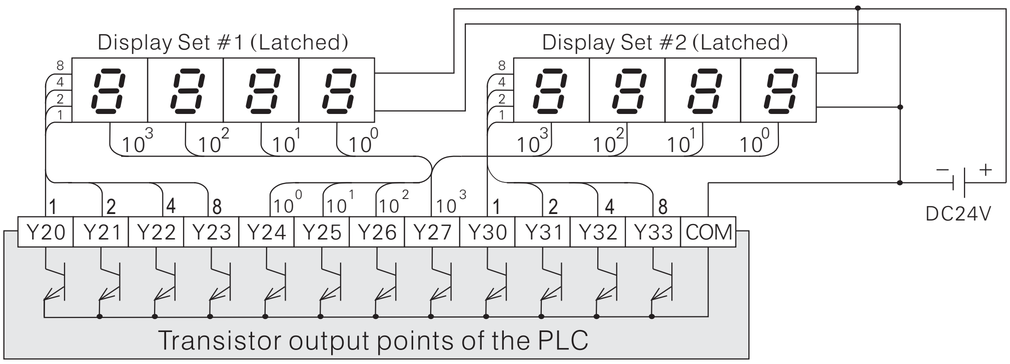
S : Source decimal value to be shown in the seven segment display

D : Initial point for the scan output of the seven segment display

n : Polarity designated for output signals and latch signals

- This instruction scan outputs to one (two) set(s) of four-digit seven segment display will occupy eight (Twelve) consecutive output points initiating from (D) and demonstrates the content value of (S) on the seven segment display. Whether there is one or two sets of four-digit display for the scan output is determined by "(n)", and also "(n)" is used to designate the polarity combination for the PLC output terminal and the display input terminal.

Circuit Diagram for Seven Segment Display Scan Output



- The diagram shown above is the circuits of a seven segment display composed of Y20～Y27. When X20= "ON", the instruction will start to be performed. The value of D0 will be converted into a BCD code then transferred and displayed in Set #1. If the value of D0 exceeds 9,999, an operation error will occur. If Display Set #2 is also connected with the circuit and the "(n)" value is set properly, the content value of D1 will be demonstrated on Display Set #2.

- when X20= "ON", Y24～Y27 will cycle the output scan automatically. It takes 12 program scan times for a display cycle and M9029 will turn "ON" for a program scan time when each cycle is completed.

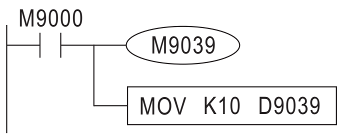
Setting value of "n"

A correct setting of "n" value is not only can be used to match the logic polarity of the PLC transistor output terminal with the input terminal of the seven segment display module but also to demonstrate there is one or two sets of display to be used.

Number of Display Sets	One set				Two sets			
Polarity of the PLC output terminal and the input terminal of the display data	Same		Different		Same		Different	
Polarity of the PLC output terminal and the input terminal of the display latched signal	Same	Different	Same	Different	Same	Different	Same	Different
n	0	1	2	3	4	5	6	7

The value of "n" is selected by referring to the table above, also it can use a number 0 ~ 3 or 4 ~ 7 to insert "n" orderly. And then test them one by one, until the value of the seven segment display is correctly demonstrate.

Notice



When the instruction is performed, at least it needs a 10ms of scan time. If the scan time is less than 10ms, please use the constant scan time function to fix the scan time at 10ms.

- The SEGL instruction can be used once only in the program.
- This instruction is only recommended for use with transistor output modules.

FNC 76 ASC	ASC (S) (D)	ASCII Code Convert	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S	Key-in eight English letters from computer															
D									○	○	○					○

```
X20              (S)       (D)
─┤ ├────[ASC  ABCDEFGH  D0]
```

S : The source of English letters will be converted to ASCII codes

D : The device where ASCII codes are stored

- When X20= "ON", English letters A ~ H will be converted into ASCII codes and stored in D0 ~ D3.

b15	b0	
42H (B)	41H (A)	D0
44H (D)	43H (C)	D1
46H (F)	45H (E)	D2
48H (H)	47H (G)	D3
Higher 8 bits	Lower 8 bits	

- If M9161= "ON", each English letter will take over a register position after conversion into an ASCII code, where lower 8 bits (b7 ~ b0) of the register will store ASCII codes and higher 8 bits (b15 ~ b8) will be filled with zero ("0").

b15	b0	
00H	41H (A)	D0
00H	42H (B)	D1
00H	43H (C)	D2
00H	44H (D)	D3
00H	45H (E)	D4
00H	46H (F)	D5
00H	47H (G)	D6
00H	48H (H)	D7
Higher 8 bits	Lower 8 bits	

- If the English letters contents in (S) is less than 8 characters, the difference is made up with "Space Key" Char (ASCII code 20H).

FNC 77 PR	PR (S) (D)	Output ASCII codes	M	VB	VH
			○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S									○	○	○					○
D		○														○

● D occupies 10 consecutive points

```
          (S) (D)
  X20
|-| |----[PR  D0  Y20]
```

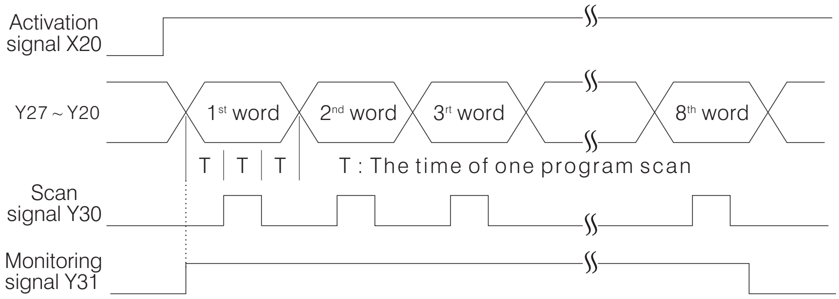
S : Source devices where ASCII codes are stored

D : Output points exporting ASCII codes

- The instruction will read ASCII codes of 4 (or 8) source registers (initiated from (S)) byte by byte. And then, orderly output the ASCII codes to the designated consecutive 8 output points (initiated from (D)).
- The process referred above designates the points from Y27 (the first bit) to Y20 (the last bit) are the data output points. It also designates Y30 as the scan signal and Y31 as the monitoring signal.
- There are two operation modes for the PR instruction, depending on the status "ON"/"OFF" of M9027.

M9027= "OFF"

- To generate the 8 words of sequence outputs. The operation sequence diagram is shown below:

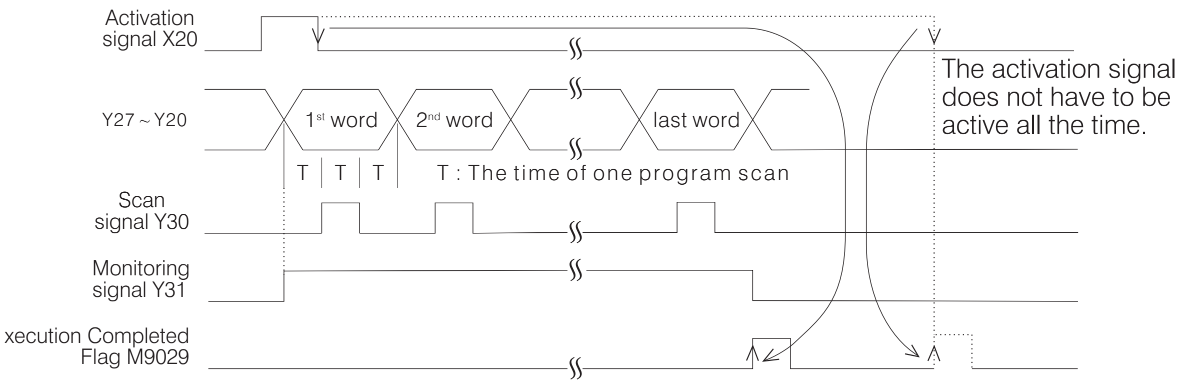


(S)	Up 8 bits	Down 8 bits
D0	2nd word	1st word
D1	4th word	3rd word
D2	6th word	5th word
D3	8th word	7th word

- If X20 turns "OFF" during the instruction is performed, the instruction is disabled then the data output will be discontinued. When X20 turns "ON" again, data will be transferred from the first letter.

M9027= "ON"

- To generate the 16 words of sequence outputs. The operation sequence diagram is shown below:



- The code "00H" (NUL) represents the end of the string and the following words will not be processed.
- If X20 always stays "ON", the output will be stopped automatically when all data are finished. Meanwhile M9029 will not be activated until X20 turns "OFF".

- Please use a transistor output unit for the output points designated by the instruction.
- The PR instruction can be used once only in the program.
- When performing the instruction, please use the constant scan time function to fix the scan time or place the instruction in a subroutine of the timer interrupt function, they will fix the time value of "T" which shown in the diagram above.

D	FNC 78 FROM	P	DFROMP (m1) (m2) (D) (n)	Read special module BFM	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
m2															○	
D						○	○	○	○	○	○					○
n															○	
● M Series: m1=1 ~ 31	● VB0 Series m1=1 ~ 4 ; VB1 Series m1=1 ~ 8 ; VB2 Series m1=1 ~ 16															
● m2=0 ~ 32767	● n=1 ~ 32767															

```
        (m1) (m2) (D) (n)
 X20
|-| |---[FROM K2 K5 D0 K4]
```

m1 : The position number of specified special module

m2 : Initial serial number of the BFM to be read

D : The initial device of storage for collect the picked up data

n : Number of data groups to be read

- The CPU module of M Series and the Main Unit of VB Series PLC use the instruction to read BFM data of the special module.
- When X20= "ON", 4 groups (they will be BFM #5 ~ BFM #8 because (n) =4 and (m2) =5) data in the specified special module (which is installed in the (m1) =2nd position) will be read and stored in D0 ~ D3.
- About the special module of the M Series and VB Series, the definitions of position are different. To assign the (m1) of M Series, please refer to the next page; For the (m1) of VB Series, each special module is consecutively assigned from K1 to K16, it begins with the closest one to the Main Unit.
- When X20= "OFF", the instruction will not be performed but the data (which was read previously) will still remain.

D	FNC 79 TO	P	DTOP (m1) (m2) (S) (n)	Special module BFM write in	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
m2															○	
S					○	○	○	○	○	○	○				○	○
n															○	
● M Series: m1=1 ~ 31	● VB0 Series m1=1 ~ 4 ; VB1 Series m1=1 ~ 8 ; VB2 Series m1=1 ~ 16															
● m2=0 ~ 32767	● n=1 ~ 32767															

```
        (m1) (m2) (S) (n)
 X20
|-| |---[TO K2 K0 D0 K1]
```

m1 : The position number of specified special module

m2 : Initial serial number of BFM which will be written

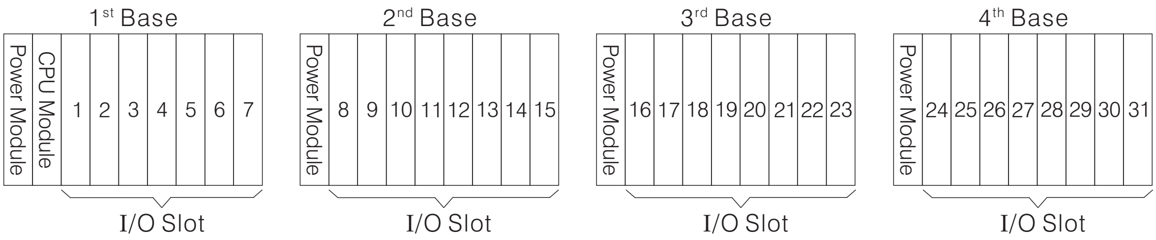
S : The initial source device, which stores the data is for the BFM

n : Number of data groups to be write

- The CPU module of M Series and the Main Unit of VB series PLC use the instruction to write BFM data to the special module.
- When X20= "ON", the content value of D0 will be written into BMF #0 of the special module which is installed in the 7th position. Because n=1, there is only one data group written in.
- About the special module of the M Series and VB Series, the definitions of position are different. To assign the (m1) of M Series, please refer to the next page; For the (m1) of VB Series, each special module is consecutively assigned from k1 to k8, it begins with the closest one to the Main Unit.
- When X20= "OFF", the instruction will not be performed but the data (which was written into the BFM previously) will still remain.

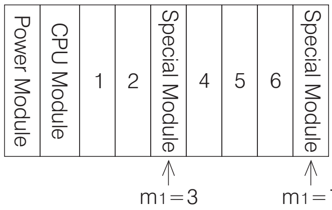
Number m1 of the Slot where Special Module is Located (For M Series only)

- M Series PLC is a module structure programmable controller. The system is composed of various I/O modules and installed on the base unit. M Series can be connected up to 4 bases and the I/O slot number is shown in the following diagram:

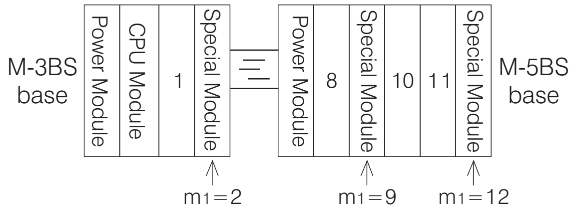


- The m1 operand "number of the slot where the special module is located" of the FROM/TO instruction is a location referring to the diagram above. In the following M-3BS is a base with 3 I/O slots, M-5BS is a base with 5 I/O slots and M-8BS is a base with 8 I/O slots.

Example 1: Only an M-8BS base



Example 2: An M-3BS base connected with an M-5BS base

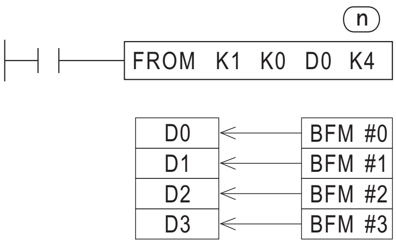


BFM Number m2

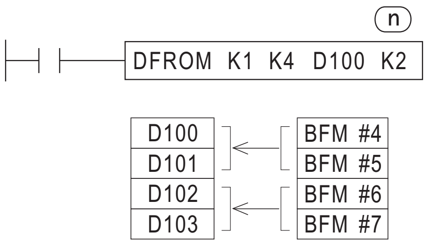
- M Series and VB Series PLC contain the Buffer Memory (BFM) which is used to store the setting value of the special module and various operation statuses. Each BFM data register has a length of 16 bits, and different special modules have different numbers of BFM registers. The number of BFM register is coded in decimal, such as #0, #1,....#9, #10,.....
- If a module is used the BFM to transfer data between itself and the Main Unit, it is called the Special Module.

Number of Data Groups n Transferred

- 16-bit instruction



- 32-bit instruction



- The number of the data groups transferred is determined by "n". n=4 in a 16-bit instruction has the same meaning with n=2 in a 32-bit instruction.

6-10 Serial Communication Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
80		RS		Serial Communications Instruction	○	○	○
81	D	PRUN	P	Parallel Run	○	○	
82		ASCI	P	Converts HEX to ASCII	○	○	○
83		HEX	P	Converts ASCII to HEX	○	○	○
84		CCD	P	Check Code	○	○	○
85		VRRD	P	VR Volume Read	○	○	○
86		VRSC	P	VR Volume Scale	○	○	○
89		LINK		Easy Link Communication	○	○	

FNC 80 RS	RS (S) (m) (D) (n)	Serial communications Instruction

M	VB	VH
○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
m											○				○	
D											○					○
n											○				○	

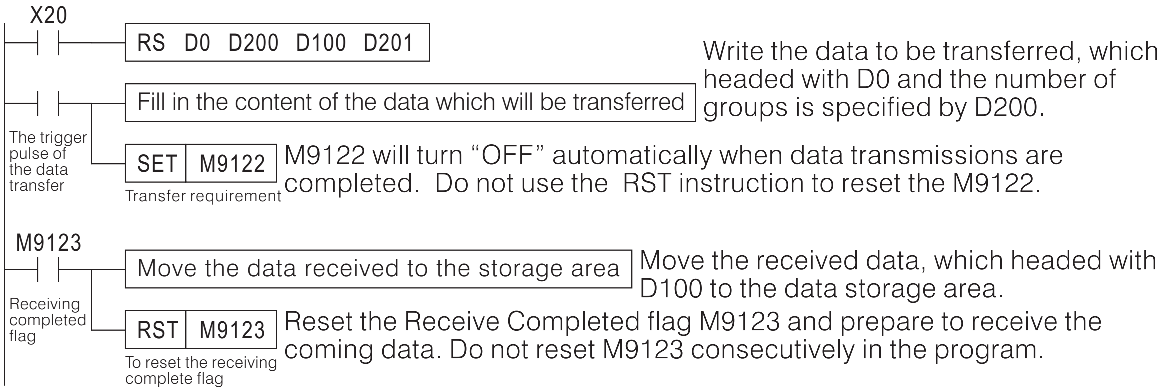
- m,n=0 ~ 256

X20 — RS D0 D200 D100 D201 (S = D0, m = D200, D = D100, n = D201)

S : Head ID number of the register transferring data
m : Number of groups transferring data
D : Head ID number of the register receiving data
n : Number of groups receiving data

- When M Series PLC's M1-CPU1 module is equipped with the communication expansion card M-232R or M-485R, therefore this CPU module is provided with the CP2 (the second Communication Port). Then the instruction can be used to transfer or receive the data via the serial communications interface of external peripheral facilities.
- When VB or VH Series PLC's Main Unit is equipped with the communication expansion card (VB-232 or VB-485) or expansion module (VB-485A , VB-CADP etc.), therefore this CPU module is provided with the CP2 (the second Communication Port). Then the instruction can be used to transfer or receive the data via the serial communications interface of external peripheral facilities.
- The CP2 is a multi-functional expansion communication port, it can operation various communication types. When the CP2 is assigned to this instruction, the manage type should select to "Non protocol". About the CP2, to select the manage type and related parameter setting, please specify it from the programming software (Ladder Master - System - 2nd COM Port Setting).
- Designate "m" as K0 where data transmission (send) is not needed, and designate "n" as K0 where data received is not needed.
- As many commercialized peripheral facilities (e.g. Inverters, barcode readers, card readers, electronic displays, etc.) equipped with serial communications interface have their individual protocols, PLC users have to use the RS instruction writing communication programs (in accordance with the communication protocol format of peripheral facilities), when M series PLC is to be connected with peripheral facilities, to transfer data between PLC and those peripherals.
- If the communication of the RS instruction is performed, data transmissions can be divided into 16-bit mode (M9161= "OFF") and 8-bit mode (M9161= "ON").
- M9063 will turn "ON" when any error occurs during data transmissions and receiving and the error code will store in D9063.
- More than one RS instruction can be programmed but only one may be active at any one time.

Sequence of Data Transmissions and Receiving



Related Flags and Data Register

①Transmission Trigger Flag M9122

- When the conditional contact X20= "ON", the RS instruction is performed. At this time, if the pulse signal forces the status of M9122 to be "ON", the content value of the register initiating from D0 will be transferred via the serial interface. When the data transmission is completed, M9122 will be reset to "OFF" automatically.

②Receive Completed Flag M9123

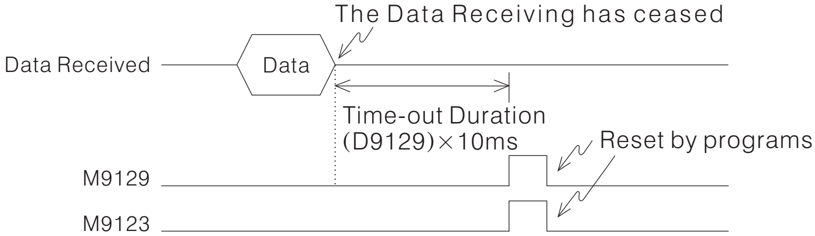
- When the conditional contact X20= "ON", the RS instruction is performed. PLC is ready for the status of receiving.
- When the data receiving is completed, M91213= "ON". At this moment, the received data in the buffer will be moved to the data storage area, and then M9123 will be reset to "OFF". Afterwards, PLC will be ready for the status of receiving immediately.

③Carrier Detection Flag M9124 (the VH series does not support this flag)

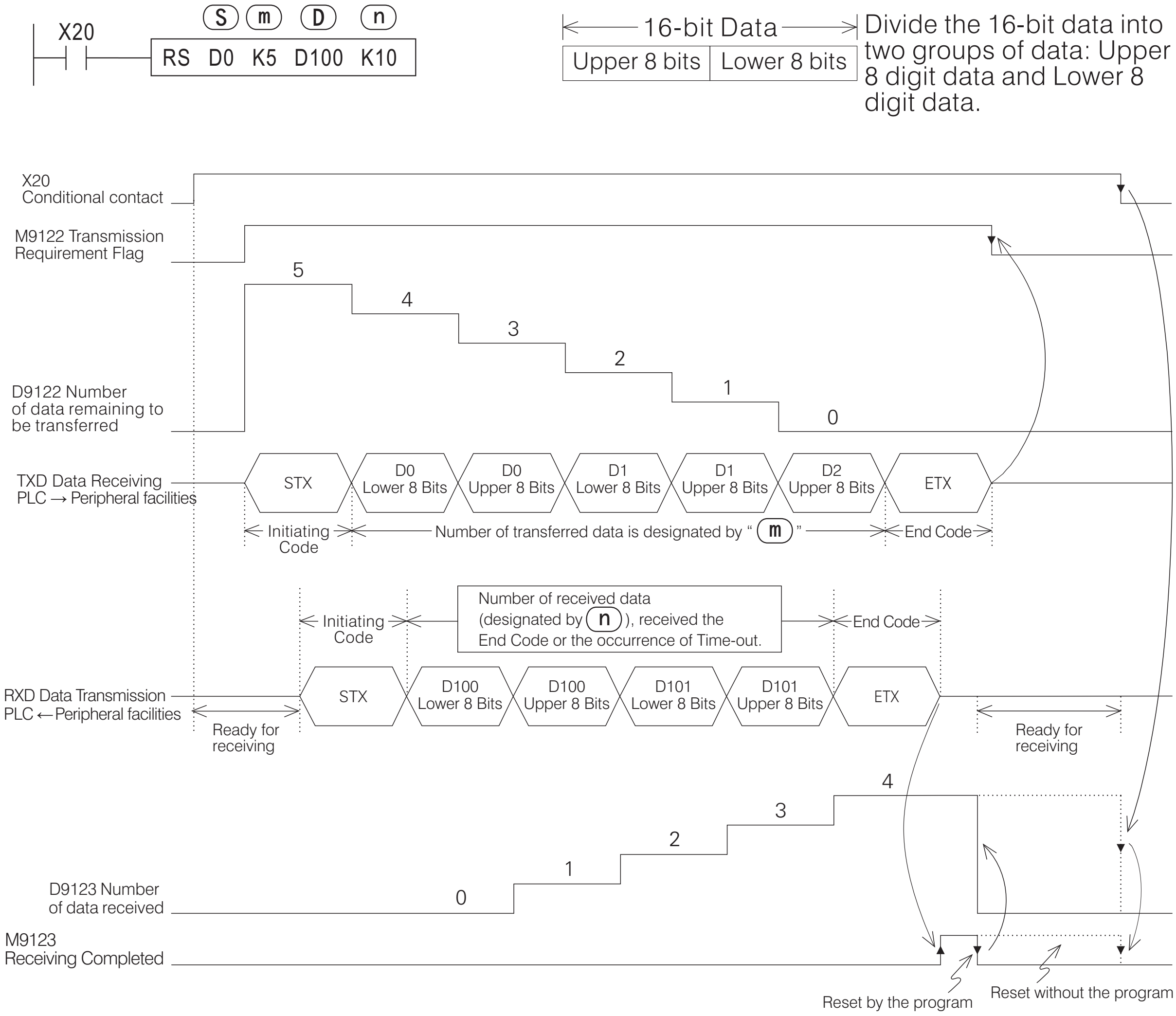
- When PLC receives the CD (Carry Detect) signal from the serial interface, M9124= "ON".
- When PLC is connected with a MODEM, the CD signal is used to represent the status of MODEM. If M9124= "OFF", the transmission of the dialing signal can be performed. If M9124= "ON", data transmissions and receiving can be performed.

④Time-out Flag M9129

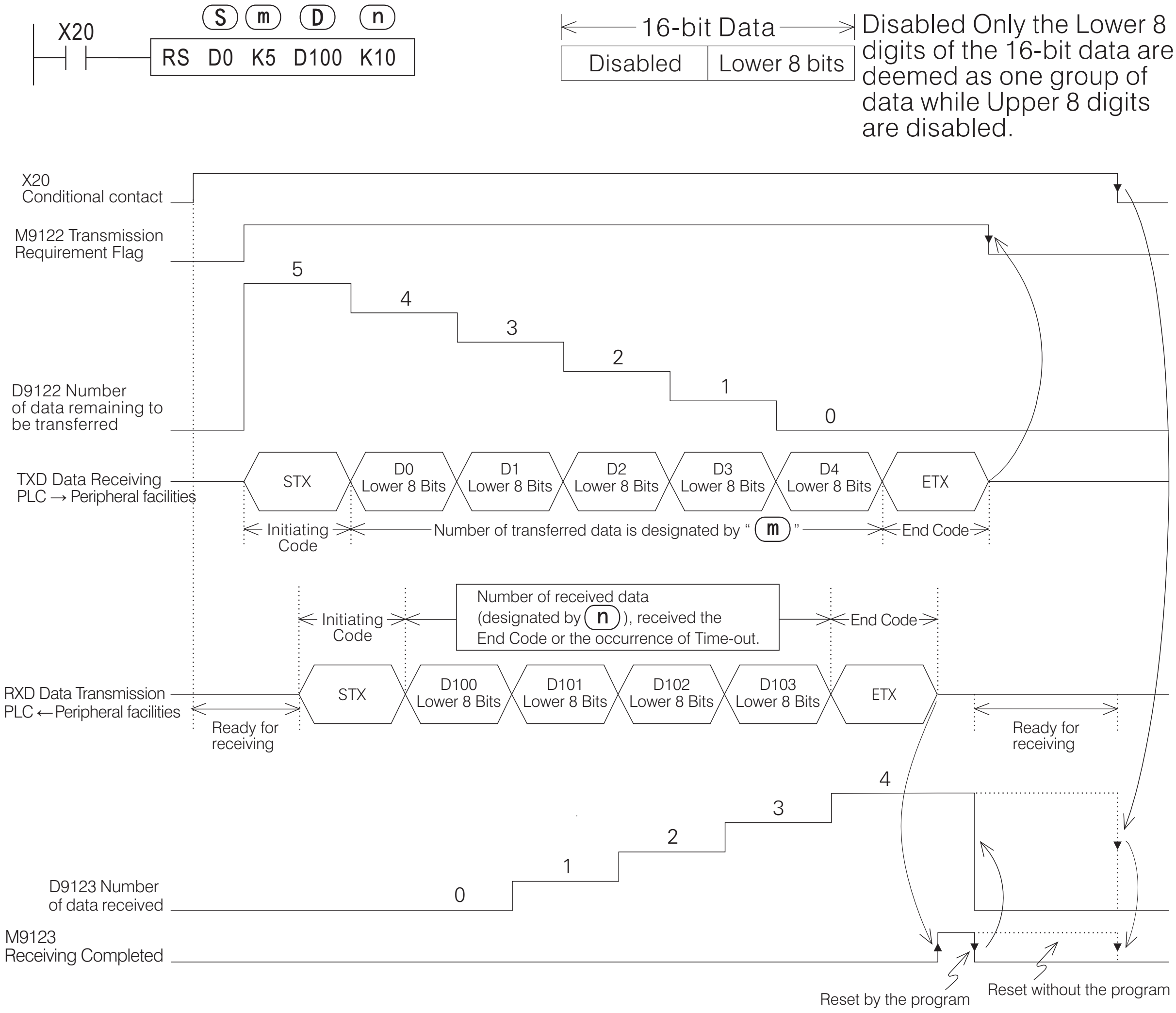
- During the data receiving, if the receiving time exceeds the time-out duration (designated by D9129), M9129 will turns "ON" to represent as the occurrence of Time-out, and also the Receive Completed flag M9123 will be forced "ON" to close the data receiving action.
- The M9129 will not be reset automatically, must using an instruction in the program to reset the status of M9129.
- By applying the Time-out function, PLC will receive the data of transferred from peripheral facilities which is no "End Code" or no length can be predicted.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) × 10ms. When D9129=0 (the default value), the Time-out duration is 100ms.



Description of Data Transmissions and Receiving Actions: 16-bit Mode (M9161="OFF")



Description of Data Transmissions and Receiving Actions: 8-bit Mode (M9161="ON")



D	FNC 81 PRUN	P	DPRUNP (S) (D)	Parallel Run	M	VB	VH
					○	○	

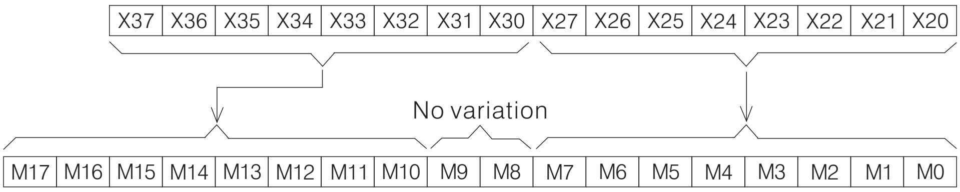
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○		○									○
D						○	○									○

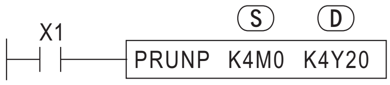
- The X, Y and M in the KnX, KnY and KnM should assigned an ID number which the least digit is a zero "0".
- When S=KnX, D should be KnM; And when S=KnM, D should be KnY.

S : Transmission source devices

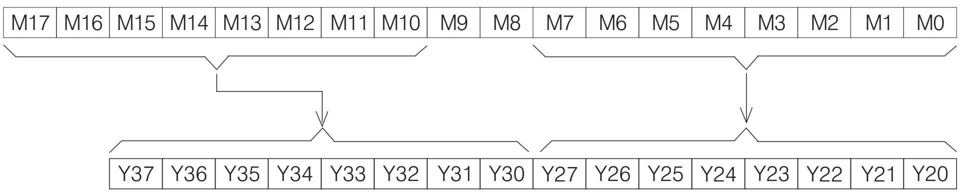
D : Transmission destination devices

- Transfer the source devices (Octonary number system, designated by S) to D.
- When X0= "ON", transfer the content of K4X20 to K4M0 in Octonary number system.





- When X1="OFF"→ "ON", transfer the content of K4M0 to K4Y20 in Octonary number system.



	FNC 82 ASCI	P	ASCIP (S) (D) (n)	Converts HEX to ASCII	M	VB	VH
					○	○	○

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to K*n*X, K*n*Y, K*n*M or K*n*S, it should be designated to K4X, K4Y, K4M or K4S.
- n= 1 ~ 256

```
        (S) (D)  (n)
X20
─┤├───ASCI D0 D100 K8
```

S : Head ID number of data source

D : Head ID number of the position where conversion results are stored

n : The number of hexadecimal data characters is selected

- When the instruction is performed, it converts each HEX character of the source devices (designated by (S)) into ASCII codes and transfers they to the designated devices (D). The number of the converted characters is determined by (n).
- ASCII codes corresponding to HEX values 0 ~ F are shown in the following table:

HEX Value	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH
ASCII Code	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H	41H	42H	43H	44H	45H	46H

- When X20= "ON", the instruction converts the 8-digit HEX value in D0 and D1 to ASCII codes, and transfers to the designated registers which are headed by D100.
- The instruction has two operation modes depending on the status of M9161:
Assume "(S)"
(D0)=4567H
(D1)=89ABH

M9161="OFF" (16-bit Conversion Mode)

- This mode will divide each designated device (D) into Upper 8 bits and Lower 8 bits, where two ASCII codes are stored respectively.

(D)	n=8	n=7	n=6	n=5	n=4	n=3	n=2	n=1
D100 Lower 8 Bits	38H	39H	41H	42H	34H	35H	36H	37H
D100 Upper 8 Bits	39H	41H	42H	34H	35H	36H	37H	
D101 Lower 8 Bits	41H	42H	34H	35H	36H	37H		
D101 Upper 8 Bits	42H	34H	35H	36H	37H			
D102 Lower 8 Bits	34H	35H	36H	37H				
D102 Upper 8 Bits	35H	36H	37H					
D103 Lower 8 Bits	36H	37H						
D103 Upper 8 Bits	37H							

M9161="ON" (8-bit Conversion Mode)

- This mode will divide each designated device (D) into Upper 8 bits and Lower 8 bits, while Upper 8 bits are filled with zero ("0") and Lower 8 bits store an ASCII codes, each register stores an ASCII code only.

(D)	n=8	n=7	n=6	n=5	n=4	n=3	n=2	n=1
D100	38H	39H	41H	42H	34H	35H	36H	37H
D101	39H	41H	42H	34H	35H	36H	37H	
D102	41H	42H	34H	35H	36H	37H		
D103	42H	34H	35H	36H	37H			
D104	34H	35H	36H	37H				
D105	35H	36H	37H					
D106	36H	37H						
D107	37H							

FNC 83 HEX	P	HEXP (S) (D) (n)	Converts ASCII to HEX	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to K*n*X, K*n*Y, K*n*M or K*n*S, it should be designated to K4X, K4Y, K4M or K4S.
- n= 1 ~ 256

```
X21          (S)   (D)  (n)
|-| |----[HEX  D100  D0  K8]
```

S : Head ID number of data source

D : Head ID number of the position where conversion results are stored

n : Number of ASCII codes converted

- When the instruction is performed, convert each ASCII code of the source device (which is designated by(S)) into a HEX value and transfer it to the designated devices(D). The number of ASCII codes converted is determined by(n).
- The following is a contrast table of ASCII codes and HEX values 0 ~ F:

ASCII Code	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H	41H	42H	43H	44H	45H	46H
HEX Value	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH

- When X21 = "ON", convert the ASCII code of the register initiating from D100 into a HEX value and transfer it to (D0) and (D1).
- If the content designated by Data source(S)is not an ASCII code of 0H ~ FH, PLC will regard it as an operation error and disable the instruction.
- The instruction has two operation modes depending on the status of M9161:

M9161="OFF" (16-bit Conversion Mode)

- This mode will convert the ASCII codes (stored in Upper 8 bits and Lower 8 bits) of each designated device(S)into HEX values.

(S)	
D100 Lower 8 Bits	38H
D100 Upper 8 Bits	39H
D101 Lower 8 Bits	41H
D101 Upper 8 Bits	42H
D102 Lower 8 Bits	34H
D102 Upper 8 Bits	35H
D103 Lower 8 Bits	36H
D103 Upper 8 Bits	37H

D1 b15	D1 Digit #3	D1 Digit #2	D1 b0	D0 b15	D0 Digit#3	D0 Digit#2	D0 b0	(n)
				0H	0H	0H	8H	n=1
				0H	0H	8H	9H	n=2
				0H	8H	9H	AH	n=3
				8H	9H	AH	BH	n=4
0H	0H	0H	8H	9H	AH	BH	4H	n=5
0H	0H	8H	9H	AH	BH	4H	5H	n=6
0H	8H	9H	AH	BH	4H	5H	6H	n=7
8H	9H	AH	BH	4H	5H	6H	7H	n=8

M9161="ON" (8-bit Conversion Mode)

- This mode will convert the ASCII codes (stored in Lower 8 bits) of each designated device(S)into HEX values.

(S)	
D100 Lower 8 Bits	38H
D101 Lower 8 Bits	39H
D102 Lower 8 Bits	41H
D103 Lower 8 Bits	42H
D104 Lower 8 Bits	34H
D105 Lower 8 Bits	35H
D106 Lower 8 Bits	36H
D107 Lower 8 Bits	37H

D1 b15	D1 Digit #3	D1 Digit #2	D1 b0	D0 b15	D0 Digit #3	D0 Digit #2	D0 b0	(n)
				0H	0H	0H	8H	n=1
				0H	0H	8H	9H	n=2
				0H	8H	9H	AH	n=3
				8H	9H	AH	BH	n=4
0H	0H	0H	8H	9H	AH	BH	4H	n=5
0H	0H	8H	9H	AH	BH	4H	5H	n=6
0H	8H	9H	AH	BH	4H	5H	6H	n=7
8H	9H	AH	BH	4H	5H	6H	7H	n=8

FNC 84 CCD	P	CCDP (S) (D) (n)	Check Code	M	VB	VH
				○	○	○

Operand	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to KnX, KnY, KnM or KnS, it should be designated to K4X, K4Y, K4M or K4S.
- n= 1 ~ 256
- D occupies 2 consecutive points

```
        (S)  (D)   (n)
  X20
--| |----[CCD  D0  D100  K8]
```

S : Head ID number of Data source

D : Position where the result of SumCheck is stored

n : Number of data

- Sum up the content of (n) byte (8-bit) data headed with (S), total of the sum is stored in the designated device (D) while the Parity bits are stored in the next register.
- When the instruction is used for communication, the "SumCheck" (or "error detect code") applied to ensure the accuracy of the data transmission.
- When X20= "ON", sum up 8 consecutive 8-bit data headed with D0, total of the sum is stored in D100 while the Parity bits are stored D101.
- The instruction has two operation modes depending on the status of M9161:

M9161= "OFF" (16-bit Mode)

- This mode will take Upper 8 bits and Lower 8 bits of each device (designated by (S)) as an 8-bit data, and do the aggregate operation and generate the Parity data.

		Data Content value	MSB	Content value in Binary						LSB
(S)	D0 Lower 8 Bits	255	1	1	1	1	1	1	1	1
	D0 Upper 8 Bits	80	0	1	0	1	0	0	0	0
	D1 Lower 8 Bits	135	1	0	0	0	0	1	1	1
	D1 Upper 8 Bits	28	0	0	0	1	1	1	0	0
	D2 Lower 8 Bits	100	0	1	1	0	0	1	0	0
	D2 Upper 8 Bits	73	0	1	0	0	1	0	0	1
	D3 Lower 8 Bits	210	1	1	0	1	0	0	1	0
	D3 Upper 8 Bits	5	0	0	0	0	0	1	0	1
(D)	D100	886								
	D101		1	1	0	0	1	1	1	0

When there is an odd number of "1", the bit corresponding to D101 = 1.

When there is an even number of "1", the bit corresponding to D101 = 0.

M9161="ON" (8-bit Mode)

- This mode will take Lower 8 bits of each device (designated by (S)) as an 8-bit data (while ignore its Upper 8 bits), and do the aggregate operation and generate the Parity data.

		Data Content value	MSB	Content value in Binary						LSB
(S)	D0 Lower 8 Bits	255	1	1	1	1	1	1	1	1
	D1 Lower 8 Bits	80	0	1	0	1	0	0	0	0
	D2 Lower 8 Bits	135	1	0	0	0	0	1	1	1
	D3 Lower 8 Bits	28	0	0	0	1	1	1	0	0
	D4 Lower 8 Bits	100	0	1	1	0	0	1	0	0
	D5 Lower 8 Bits	73	0	1	0	0	1	0	0	1
	D6 Lower 8 Bits	210	1	1	0	1	0	0	1	0
	D7 Lower 8 Bits	5	0	0	0	0	0	1	0	1
(D)	D100	886								
	D101		1	1	0	0	1	1	1	0

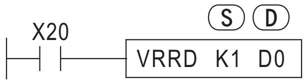
When there is an odd number of "1", the bit corresponding to D101 = 1.

When there is an even number of "1", the bit corresponding to D101 = 0.

FNC 85 VRRD	P	VRRD P (S) (D)	VR Volume Read	M	VB	VH
				○	○	○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S															○	○
D						○	○	○	○	○	○	○		○		○

- S=1~2

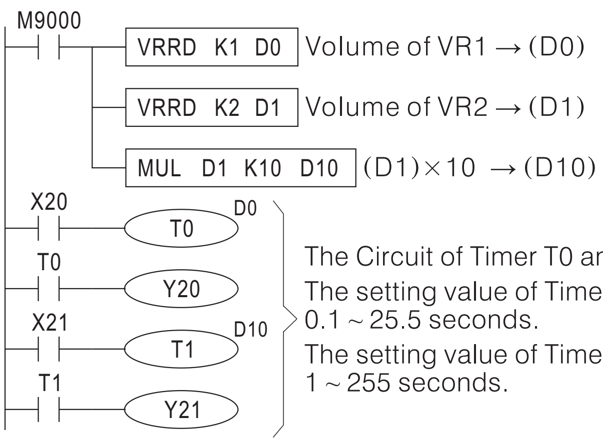


S : ID number of the Knob

D : Destination device where the volume is stored

- The VRRD instruction is used to read the volume of VR1 or VR2 in M series M1-CPU1 Module or VB series Main Unit. Convert the volume into a value ranging from 0 to 255 and store it in the designated device (D).
- When X20= "ON", convert the volume of VR1 into a BIN format which ranging from 0 to 255 and store it in D0.
- To acquire a value larger than 255, can multiply D0 by a constant.

M9161= "OFF" (16-bit Mode)



The Circuit of Timer T0 and T1

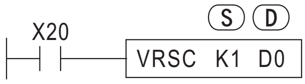
The setting value of Timer T0 is adjusted by VR1, the range will be 0.1 ~ 25.5 seconds.

The setting value of Timer T1 is adjusted by VR2, the range will be 1 ~ 255 seconds.

				M	VB	VH
FNC 86 VRSC	P	VRSCP S D	VR Volume Scale	○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S															○	○
D						○	○	○	○	○	○	○		○		○

- S=1 ~ 2

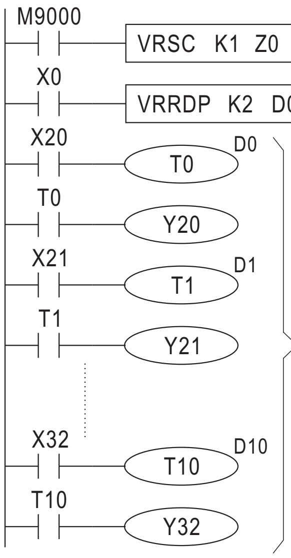


S : ID number of the Knob

D : Destination device where the volume is stored

- The VRSC instruction is used to read the scale of VR1 or VR2 in M series M1-CPU1 Module or VB series Main Unit. The scale (as a rotary switch with 11 set positions 0 ~ 10) is stored in the designated device D. When the volume is located between two scales, it will rounds up or down to an integer 0 ~ 10.
- When X20= "ON", read the scale (0 ~ 10) of VR1 and store it in D0.

Using the value of VR1 and VR2 to change the timer setting value of T0~T10



Volume Scale of VR1 → (Z0)

When X0= "OFF" → "ON", based on the scale of VR1, transfer the volume of VR2 to the corresponding register D0 ~ D10, then use the value of the register as a timer setting value of T0 ~ T10.

The Circuit of Timer T0 ~ T10

The setting value of Timer T0 ~ T10 is adjusted by VR1, VR2 and X0, the range will be 0.1 ~ 25.5 seconds.

FNC 89 LINK	LINK (S1) (S2)	Easy Link Communication	M ○	VB ○	VH

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					

X20 ──┤ ├── LINK D1000 D100 (S1 = D1000, S2 = D100)

S1 : Head ID number of the register, which describe the data transfer/receive actions

S2 : Instruction working area, occupies 4 consecutive registers

- If the M Series CPU module mounts a M-232R or M-485 communication card, the CPU module will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between PLCs.
- If the VB Series Main Unit mounts a communication card (VB-232 or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd. Communication Port). Then, via this instruction to proceed data transfer between PLCs.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 is assigned to this instruction, the communication type should use "EASY LINK" or "COMPUTER LINK". To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master "System---2nd COM Port Setting..." to get the right setting.
- At most 256 nodes of M/VB Series PLC (slave VH series). can be linked together via this instruction and the RS-485 interface. The instruction can use for transfer the data of device X, Y, M, S, T, C and D.
- As the diagram below, select one of these linked PLCs as the Master station and the rest as Slave stations. Use the program develop devices (e.g. Ladder Master) to set the "EASY LINK" or "COMPUTER LINK" as the communication mode between the Master and Slave stations, and set each Slave station properly (the range of station ID number is 1 ~ 255). And then, write the data transmission/receiving command (designated by this instruction) to the Master station, to achieve the data transmission between PLCs.

- When X20= "ON", the LINK instruction will start to be performed. Based on the designated register string (which initiating from D1000), to do the data write or read action to the appointed Slave PLC station. And also, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by (S1) is duly completed, the M9199 will be "ON" for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20="ON" → "OFF", the instruction will be stopped and the data transmission/receiving will be disabled immediately.
- The LINK instruction can be used once only in the program.

- The register headed with S1 is used to describe the data transmission/receiving information:

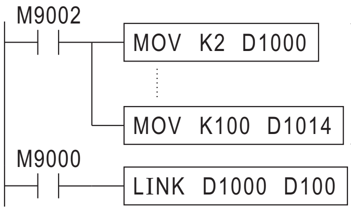
S1	Content Value	Description	
D1000	1 ~ 255	To designate the number of transferred and received data sets. Each data transmission/receiving set should be described with 7 registers.	
D1001	1 ~ 255	Designates the Slave station ID number, to proceed data transmission/receiving for the particular Slave station	Description of the 1st data transmission/receiving operation
D1002	1 ~ 2	Instruction code. 1: read data from Slave stations; 2: write data in Slave stations	
D1003	1 ~ 64	Length of data transferred or received. (If the data designated is a 32-bit counter, the content value = 1 ~ 32)	
D1004	1 ~ 6 10 ~ 13	Designates the device type of the Master station 1:Input Contact X 2:Output Contact Y 3:Auxiliary Coil M 4:State Coil S 5:Timer Contact T 6:Counter Contact C 10:The Present-value Register of the Timer 11:16-bit Counter, Present-value Register 12:32-bit Counter, Present-value Register 13:Data Register D	
D1005		Designates the initial ID number of the Master station device	
D1006	1 ~ 6 10 ~ 13	Designates the device type of the Slave station	
D1007		Designates the initial ID number of the Slave station device	
D1008	1 ~ 255	Designates the Slave station ID number	Description of the 2nd data transmission/receiving operation
D1009	1 ~ 2	Instruction code	
D1010	1 ~ 64	Length of data transferred/received	
D1011	1 ~ 6 10 ~ 13	Designates the device type of the Master station	
D1012		Designates the initial ID number of the Master station device	
D1013	1 ~ 6 10 ~ 13	Designates the device type of the Slave station	
D1014		Designates the initial ID number of the Slave station device	
⋮	⋮		

- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.

- The instruction working area headed with S2:

S2	Description	
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
D100	Upper 8 bits	Instruction working status 0:Normal data transmission/receiving 2:Error of the length of the transferred/received data (unequal to 1 ~ 64) 4:Error of the designated device type 5:Error of the designated device ID number 6:The attributes of the designated devices by the Master and Slave stations are different A:Normal communications but no response from Slave stations B:Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

- Programming Example



Build data receive/send communication table.
User can also use "Ladder Master - Edit Communication Table" function to replace this block of program.

EASY LINK communication instruction

There are totally 2 transmission/receiving data sets in this example.

① Read D10 ~ D19 of Slave station #5 to D0 ~ D9 of the Master station

② Write M0 ~ M29 of the Master station to M100 ~ M129 of Slave station #2.

S1	Content Value	
D1000	2	Two transmission/receiving data sets
D1001	5	Designates Slave station #5
D1002	1	Reads data from the Slave station
D1003	10	Length of the data to be read
D1004	13	Designates the device headed with the Master station as D0
D1005	0	
D1006	13	Designates the device headed with the Slave station as D10
D1007	10	
D1008	2	Designates Slave station #2
D1009	2	Write data to the Slave station
D1010	30	Length of the data to be written
D1011	3	Designates the device headed with the Master station as M0
D1012	0	
D1013	3	Designates the device headed with the Slave station as M100
D1014	100	

D1001 ~ D1007: The 1st transmission/receiving data set:
D10 ~ D19 of Slave station #5
↓
D0 ~ D9 of the Master

D1008 ~ D1014: The 2nd transmission/receiving data set:
M0 ~ M29 of the Master
↓
M100 ~ M129 of Slave station #2

- Edit Communication Table

Besides using program to build data receiving/sending communication table, Ladder Master provides a more user-friendly data input interface to let the users build communication table.
Select the Ladder Master "Tools ---- Edit Communication Table ...." menu to enter the communication table edition screen. Through a step-by-step guiding window, the user can easily create and edit communication table.
After the edition is done, the communication data will be stored into file register assigned by the user, and the table is created. This function also allows the user to retrieve the table data back from file register for editing.
For VB series PLCs, the file register is read-only, and its value will be treated as part of the user program. When user copy or save program file, the file register together with the program itself will be copied or saved. This feature makes the file register very suitable for communication table storing. It can be easily copied from and helps to save PLC program space. For detailed introduction on file register, please refer to "2-9 File Register (D)".

- Communication Table Example

Instruction: LINK ▼ Table Starting Position: D1000 Table Length: 15

Number	Command	Master Data		Slave ID	Slave Data	Length	Word / Bit
1	Read	D0	<--	5	D10	10	W
2	Write	M0	-->	2	M100	30	B

MEMO

MEMO

6-11 Serial Communication Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
110	D	ECMP	P	Compares two BIN floating point values		○	
111	D	EZCP	P	Compares a BIN float range with a BIN float value		○	
118	D	EBCD	P	Converts BIN floating point format to DEC format		○	
119	D	EBIN	P	Converts DEC format to BIN floating point format		○	
120	D	EADD	P	Adds up two BIN floating point numbers		○	
121	D	ESUB	P	Subtracts one BIN floating point number from another		○	
122	D	EMUL	P	Multiplies two BIN floating point numbers		○	
123	D	EDIV	P	Divides one BIN floating point number from another		○	
127	D	ESQR	P	Square root of a BIN floating point value		○	
129	D	INT	P	BIN floating point → BIN integer format		○	
130	D	SIN	P	Calculates the sine of a BIN floating point value		○	
131	D	COS	P	Calculates the cosine of a BIN floating point value		○	
132	D	TAN	P	Calculates the tangent of a BIN floating point value		○	

D	FNC 110 ECMP	P	DECMPP (S1) (S2) (D)	Compares two BIN floating point values	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D		○	○	○												○

- D occupies 3 consecutive devices

```
              (S1) (S2) (D)
   X0
 |-| |-----[DECMP  D0  D10  M0]
```

S1 : Comparative value data 1

S2 : Comparative value data 2

D : Comparison result

- The data of (S1) is compared to the data of (S2). The result is indicated by 3 bit devices which are specified with the head address entered as (D).
- When X0="ON", this instruction is activated.

  If (D1,D0) the double BIN floating number (S1) > (D11,D10) the double BIN floating number (S2), then M0="ON".

  If (D1,D0) the double BIN floating number (S1) = (D11,D10) the double BIN floating number (S2), then M1="ON".

  If (D1,D0) the double BIN floating number (S1) < (D11,D10) the double BIN floating number (S2), then M2="ON".
- When X0 turns "OFF", this instruction is deactivated. Then, the "ON"/"OFF" status of M0 ~ M2 will be kept the event before X0="OFF".
- This instruction is a 32 bit instruction, therefore must use DECMP or DECMPP in a program.
- Please combine two of M0 ~ M2when the result ≤ , ≥ or ≠ is needed.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the comparison function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 111 EZCP	P	DEZCPP (S1) (S2) (S) (D)	Compares a BIN float range with a BIN float value	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
S											○				○	○
D		○	○	○												○

- D occupies 3 consecutive devices
- $S_1 \leq S_2$

```
              (S1) (S2) (S)  (D)
  X0
--| |-------[DEZCP  D0   D2   D10  M0]
```

S_1 : Upper limit of the data range

S_2 : Lower limit of the data range

S : Comparative value

D : Compared result, occupies 3 consecutive devices

- The value of (S) is compared to the data range between (S1) and (S2). The result is indicated by 3 bit devices which are specified with the head address entered as (D).
- When X0="ON", this instruction is activated.

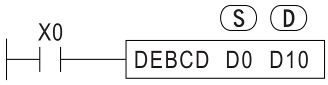
  If (D11,D10) the double BIN floating number (S) < (D1,D0) the double BIN floating number (S1), then M0="ON".

  If (D1,D0) the double BIN floating number (S1) <= (D11,D10) the double BIN floating number (S) <= (D2,D3) the double BIN floating number (S2), then M1="ON".

  If (D11,D10) the double BIN floating number (S) > (D3,D2) the double BIN floating number (S2), then M2="ON".
- When X0 turns "OFF", this instruction is deactivated. Then, the "ON"/"OFF" status of M0 ~ M2 will be kept the event before X0="OFF".
- This instruction is a 32 bit instruction, therefore must use DEZCP or DEZCPP in a program.
- When (S1) > (S2), the value of (S1) will become both Upper/Lower Limit to compares with (S).
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the comparison function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 118 EBCD	P	DEBCDP S D	Converts BIN floating point format to DEC format	M	VB	VH
						○	
D	FNC 119 EBIN	P	DEBINP S D	Converts DEC format to BIN floating point format	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○



S : Source Device of Transfer

D : Destination Device

- When X0= "ON", this instruction is activated. It will uses the BIN format value in (D1,D0) to convert the number to a DEC format number then moves the value into (D11,D10).
- This instruction is a 32 bit instruction, therefore must use DEBCD or DEBCDP in a program.
- Ex. If the content value of (D1,D0) is 1.234×10^2, then after the convert, (D10)=1234 and (D11)= -1.

S : Source Device of Transfer

D : Destination Device

- When X1= "ON", this instruction is activated. It will uses the DEC format value in (D3,D2) to convert the number to a BIN format number then moves the value into (D13,D12).
- This instruction is a 32 bit instruction, therefore must use DEBIN or DEBINP in a program.
- Ex. If the content values of (D2) = 2345 and (D3)=5, then after the convert, the content value of (D13,D12) is 2.345×10^8.

- All of floating point number will occupy two Registers.
- Please refer to CH 2-12 "Numerical System", for the formats of BIN and DEC floating point numbers are stored in Registers.

D	FNC 120 EADD	P	DEADDP (S1) (S2) (D)	Adds up two BIN floating point numbers	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○

X0 —| |— [DEADDP D0 D2 D10] (S1) (S2) (D)

S1 : Summand

S2 : Addend

D : Total

- When X0= "OFF" → "ON", the BIN floating point summand (D1,D0) will be added to the Bin floating point addend (D3,D2), and the total will be stored at the specified destination devices (D11,D10).

1.235×10^2 (D1,D0) BIN floating point number

$+\ 3.2 \times 10^0$ (D3,D2) BIN floating point number

1.267×10^2 (D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DEADD or DEADDP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the addition function.
- To execute this instruction, the result will reacted on the status of flags.

  If the result of the calculation is equal to zero, the zero flag M9020="ON".

  If the value of calculated result (D) exceeds the available range of a BIN floating point number (including positive and negative), then the carry flag M9022= "ON" and the result (D) is set to the largest value.

  If the value of calculated result (D) is smaller than the available range of a BIN floating point number (including positive and negative), then the borrow flag M9021= "ON" and the result (D) is set to the smallest value.

  The available value range of a BIN floating point number, please refer to CH 2-12 "Numerical System"
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 121 ESUB	P	DESUBP S1 S2 D	Subtracts one BIN floating point number from another	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1											○				○	○
S_2											○				○	○
D											○					○

```
        S1 S2  D
  X0
 -| |---[DESUBP D0 D2 D10]
```

S_1 : Minuend

S_2 : Subtrahend

D : Remainder

- When X0= “OFF” → “ON”, the BIN floating point subtrahend (D3,D2) will be subtracted from the BIN floating point minuend (D1,D0), and the remainder will be stored at the specified destination devices (D11,D10).

	1.235×10^2	(D1,D0) BIN floating point number
−	3.2×10^0	(D3,D2) BIN floating point number
	1.203×10^2	(D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DESUB or DESUBP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the subtraction function.
- To execute this instruction, the result will reacted on the status of flags.

  If the result of the calculation is equal to zero, the zero flag M9020=”ON”.

  If the value of calculated result D exceeds the available range of a BIN floating point number (including positive and negative), then the carry flag M9022= “ON” and the result D is set to the largest value.

  If the value of calculated result D is smaller than the available range of a BIN floating point number (including positive and negative), then the borrow flag M9021= “ON” and the result D is set to the smallest value.

  The available value range of a BIN floating point number, please refer to CH 2-12 “Numerical System”
- All of floating point number will occupy two Registers, please refer to CH 2-12 “Numerical System” for the format of a floating point number is stored in Registers.

D	FNC 122 EMUL	P	DEMULP S1 S2 D	Multiplies two BIN floating point numbers	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○

X0 ─┤ ├─ DEMUL D0 D2 D10 (S1 S2 D)

S1 : Minuend

S2 : Subtrahend

D : Remainder

- When X0= "ON", the BIN floating point multiplicand (D1,D0) will be multiplied by the BIN floating point multiplier (D3,D2), and the product will be stored at the specified destination devices (D11,D10).

	3.14×10^{0}	(D1,D0) BIN floating point number
$\times$	2.3×10^{1}	(D3,D2) BIN floating point number
	7.222×10^{1}	(D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DEMUL or DEMULP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the multiply function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 123 EDIV	P	DEDIVP (S1) (S2) (D)	Divides one BIN floating point number from another	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○				○	○
S2											○				○	○
D											○					○

X0 ─┤├─ DEDIVP D0 D2 D10 (S1)(S2)(D)

S1 : Dividend
S2 : Divisor
D : Quotient

- When X0= "OFF" → "ON", the BIN floating point dividend (D1,D0) will be divided by the BIN floating point divisor (D3,D2), and the quotient will be stored at the specified destination devices (D11,D10).

	1.23×10^{4}	(D1,D0) BIN floating point number
÷	3.0×10^{-1}	(D3,D2) BIN floating point number
	4.1×10^{4}	(D11,D10) BIN floating point number

- This instruction is a 32 bit instruction, therefore must use DEDIV or DEDIVP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the division function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- PLC will identify an operation error, if the divisor (S2) is equal to "0".

D	FNC 127 ESQR	P	DESQRP Ⓢ Ⓓ	Square root of a BIN floating point value	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○				○	○
D											○					○

```
          Ⓢ  Ⓓ
 X0
─┤ ├────DESQR D0 D10
```

S : Source device

D : Storage device for the result of square root

- This square root function is performed on the specified BIN floating point value of Ⓢ and the result is stored on Ⓓ.
- When X0= "ON", the function is activated, uses the BIN floating point source (D1,D0) to get its square root, and the result will be stored at the specified destination devices (D11,D10) by BIN floating point format.
- This instruction is a 32 bit instruction, therefore must use DESQR or DESQRP in a program.
- If the operand is assigned to an integer value K or H, this instruction will automatically converted the number to BIN floating point number then it can execute the square root function.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- If the result of the calculation is equal to zero, the zero flag M9020= "ON".
- Ⓢ can be assigned to a positive number only, if Ⓢ is a negative then an error occurs and the error flag M9067="ON".

D	FNC 129 INT	P	DINTP (S) (D)	BIN floating point BIN integer format	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○

X0 — INT D0 D10 ((S) D0, (D) D10)

S : Source device

D : Converted result

- When X0= "ON", the function is activated, uses the BIN floating point source (D1,D0) to convert the value to a equal or nearest smaller BIN integer format number, the result will be stored at the specified destination device (D10) and the number behind decimal point will be rejected.
- If the result of the conversion is equal to zero, the zero flag M9020= "ON".
  If the number behind decimal point has been rejected, the borrow flag M9021= "ON".
  If the result is exceed the range below, the carry flag M9022= "ON" to indicate overflow.
  16 bit instruction: – 32,768 ~ 32,767
  32 bit instruction: – 2,147,483,648 ~ 2,147,483,647
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.

D	FNC 130 SIN	P	DSINP (S) (D)	Calculates the sine of a BIN floating point value	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○

```
       (S) (D)
 X0
─┤ ├──[DSIN D0 D10]
```

S : Source device for the radians angle

D : Calculated result

- This instruction preforms the mathematical SIN operation on the floating point value in (S) (radian), the result is stored in (D) .
- When X0= "ON", the function is activated, uses the BIN floating point radian (D1,D0) to calculate the sine value and the result will be stored at the specified destination devices (D11,D10).
- Radian = Degree × π ÷180
- This instruction is a 32 bit instruction, therefore must use DSIN or DSINP in a program.
- In this instruction, both (S) and (D) are BIN floating point numbers.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- Below is an program example of how to calculate angles (45°) in radian using floating point, then use the radian to get the value of sine.

```
 M9000
─┤ ├──┬─[DEDIV K314159 K18000000 D0]   ( π ÷ 180 ) → (D1,D0)
      ├─[DEMUL K45 D0 D2]   45° [angel] × ( π ÷ 180 ) → (D3,D2) [radian]
      └─[DSIN D2 D10]   The result value of SIN is stored in (D11,D10).
```

D	FNC 131 COS	P	DCOSP S D	Calculates the cosine of a BIN floating point value	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○

```
X0          S   D
-| |-----[DCOS D0 D10]
```

S : Source device for the radians angle

D : Calculated result

- This instruction preforms the mathematical COS operation on the floating point value in S (radian), the result is stored in D.
- When X0= "ON", the function is activated, uses the BIN floating point radian (D1,D0) to calculate the cosine value and the result will be stored at the specified destination devices (D11,D10).
- Radian = Degree × π ÷180
- This instruction is a 32 bit instruction, therefore must use DCOS or DCOSP in a program.
- In this instruction, both S and D are BIN floating point number.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- Below is an program example of how to calculate angles (45°) in radian using floating point, then use the radian to get the value of cosine.

```
M9000
-| |--+--[DEDIV K314159 K18000000 D0]   ( π ÷ 180 ) → (D1,D0)
      |
      +--[DEMUL K45 D0 D2]   45° [angel] × ( π ÷ 180 ) → (D3,D2) [radian]
      |
      +--[DCOS D2 D10]   The result value of COS is stored in (D11,D10).
```

D	FNC 132 TAN	P	DTANP S D	Calculates the tangent of a BIN floating point value	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D											○					○

```
      X0           S   D
 ──┤ ├────[DTAN  D0  D10]
```

S : Source device for the radians angle

D : Calculated result

- This instruction preforms the mathematical TAN operation on the floating point value in S (radian) , the result is stored in D .
- When X0= "ON", the function is activated, uses the BIN floating point radian (D1,D0) to calculate the tangent value and the result will be stored at the specified destination devices (D11,D10).
- Radian = Degree × π ÷180
- This instruction is a 32 bit instruction, therefore must use DTAN or DTANP in a program.
- In this instruction, both S and D are BIN floating point number.
- All of floating point number will occupy two Registers, please refer to CH 2-12 "Numerical System" for the format of a floating point number is stored in Registers.
- Below is an program example of how to calculate angles (45°) in radian using floating point, then use the radian to get the value of tangent.

```
  M9000
──┤ ├──┬─[DEDIV  K314159  K18000000  D0]   ( π ÷ 180 ) → (D1,D0)
       ├─[DEMUL  K45  D0  D2]   45° [angel] × ( π ÷ 180 ) → (D3,D2) [radian]
       └─[DTAN  D2  D10]   The result value of TAN is stored in (D11,D10).
```

MEMO

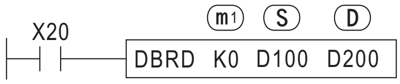
6-12 Others

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
90		DBRD	P	Reads data from the data bank	○	○	
91		DBWR	P	Writes data into the data bank	○	○	
147	D	SWAP	P	Swaps high/low byte	○	○	
176		TFT		Reads data from the data bank	○	○	○
177		TFH		Reads data from the data bank	○	○	○
178		TFK		Reads data from the data bank	○	○	○

					M	VB	VH
	FNC 90 DBRD	P	DBRDP (m1) (S) (D)	Reads data from the data bank	○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
S											○	○			○	○
D											○	○				○

- m1=0
- M series: S = 0 ~ 1021
- VB series: S = 0 ~ 2045



m1: The location of data bank

S : The data in specific page of data bank will be read

D : The initial ID of specified registers, which are assigned as the data storage

- The M and VB series PLC are able to install a data bank, it can store and apply huge data.

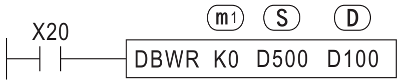
Data Bank	M series	VB series
Model number	M-DB1	VB-DB1R
Component parts	Flash ROM	SRAM + Lithium battery
Memory capacity	1022 pages (64 Words / page)	2046 pages (64 Words / page)

- The M series PLC can use this instruction to read the data in the M-DB1 data bank.
- The VB series PLC can use this instruction to read the data in the VB-DB1R data bank.
- If D100=3 and X20= "ON", it will execute to read the data in page 3 of the data bank and put the data in D200 ~ D263.
- One page of a data bank can store 64 registers' data.
- When X20= "OFF", the instruction will not be performed but the data (which was read previously) will still remain.

FNC 91 DBWR	P	DBWRP (m1) (S) (D)	Writes data into the data bank	M	VB	VH
				○	○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
m1															○	
S											○	○				○
D											○	○			○	○

- m1=0 • M series: D = 0 ~ 1021 • VB series: D = 0 ~ 2045



m1: The location of data bank

S : The source data in specific registers (which are starting from S), would be written into the data bank

D : The specific page in data bank will be covered

- The M and VB series PLC are able to install a data bank, it can store and apply huge data.

Data Bank	M series	VB series
Model number	M-DB1	VB-DB1R
Component parts	Flash ROM	SRAM + Lithium battery
Memory capacity	1022 pages (64 Words / page)	2046 pages (64 Words / page)

- The M series PLC can use this instruction to write the data into the M-DB1 data bank.
- The VB series PLC can use this instruction to write the data into the VB-DB1R data bank.
- If D100=4 and X20= "ON", it will read the data from registers D500 ~ D563 and write the data into page 4 of the data bank.
- One page of a data bank can store 64 registers' data.
- Since the M-DB1 is using the Flash ROM technique to storage data. Even though, in every page of the memory, the rewrite operate is available to be used more than 10,000 times. But, it still has the limit. So, when the program using the instruction DBWR to rewrite data into M-DB1, better change it to the instruction DBWRP. The DBWRP can avoid useless operate of rewrite, and then extend the lifespan of the Flash ROM. The VB series rewrite operate times is unlimited.
- When M series CPU module rewrites data to a M-DB1, every single page needs 10ms to execute the function. And at the same time, other executing function will be interrupted. The current value of Watch Dog timer will be reset. The VB series won't has this reaction.

D	FNC 147 SWAP	P	DSWAPP (D)	Byte Swap	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D						○	○	○	○	○	○	○		○		○

X20 — SWAPP D0 (D)

D : Device which Higher/Lower 8 bits are to be exchanged

- When X20="OFF" → "ON", Higher 8 bits and Lower 8 bits of (D0) will be exchanged.

Higher 8 Bits / Lower 8 Bits

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	D0

X20=OFF → ON

1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	D0

X21 — DSWAP D10 (D)

- When X21= "ON", Higher 8 bits and Lower 8 bits of (D10) will be exchanged. And also, Higher 8 bits and Lower 8 bits of (D11) will be exchanged.

Higher 8 Bits / Lower 8 Bits

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	D10
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	D11

X21=ON

1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	D10
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	D11

FNC 176 TFT	TFT (D1) (S) (D2)	Timer (10 ms)	M	VB	VH
			○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D1											○	○				○
S											○	○			○	○
D2		○	○	○												○

- S=0 ~ 32767, otherwise S=0

X0 —| |— TFT D0 K100 M0 ((D1) (S) (D2))

D1 : The current value of the timer (unit=10ms.)
S : The setting value of the timer (unit=10ms.)
D2: The output contact of the timer

- The unit of this instruction is used the 10ms. base timer.
- The timer count the time by up counting clock pulses. When the Current value (D1) = Setting value (S) (the value designated to a Timer), then the Timer's contact (D2) will be activated (ON).
- This timer's real setting value = 10 ms. × setting value (S) .
- The example above:
  When X0= "ON", the current value of the timer starts to count clock pulses (by unit: 10 ms). When the current value reaches the setting value K100 (1 second), the contact M0= "ON".
  When input contact X0 = "OFF" or the power failure, the Current value of Timer will return to "0" and the contact will become "OFF".

	FNC 177 TFH		TFH D1 S D2	Timer (100 ms)	M	VB	VH
					○	○	○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D1											○	○				○
S											○	○			○	○
D2		○	○	○												○

- S=0 ~ 32767, otherwise S=0

```
 X0          (D1) (S)  (D2)
|-| |-------[TFH D0 K100 M0]
```

D1 : The current value of the timer (unit=100ms.)
S : The setting value of the timer (unit=100ms.)
D2: The output contact of the timer

- The unit of this instruction is used the 100ms. base timer.
- The timer count the time by up counting clock pulses. When the Current value D1 = Setting value S (the value designated to a Timer), then the Timer's contact D2 will be activated (ON).
- This timer's real setting value = 100 ms. × setting value S .
- The example above:
  When X0= "ON", the current value of the timer starts to count clock pulses (by unit: 100 ms). When the current value reaches the setting value K100 (10 second), the contact M0= "ON".
  When input contact X0 = "OFF" or the power failure, the Current value of Timer will return to "0" and the contact will become "OFF".

FNC 178 TFK	TFK (D1) (S) (D2)	Timer (1 second)	M	VB	VH
			○	○	○

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
D1											○	○				○
S											○	○			○	○
D2		○	○	○												○

- S=0～32767, otherwise S=0

X0 — TFK D0 K100 M0 [(D1) (S) (D2)]

D1 : The current value of the timer (unit=1 sec.)
S : The setting value of the timer (unit=1 sec.)
D2: The output contact of the timer

- The unit of this instruction is used the 1 sec. base timer.
- The timer count the time by up counting clock pulses. When the Current value (D1) = Setting value (S) (the value designated to a Timer), then the Timer's contact (D2) will be activated (ON).
- his timer's real setting value = 1 sec. × setting value (S) .
- The example above:
  When X0= "ON", the current value of the timer starts to count clock pulses (by unit: 1 second). When the current value reaches the setting value K100 (100 second), the contact M0= "ON".
  When input contact X0 = "OFF" or the power failure, the Current value of Timer will return to "0" and the contact will become "OFF".

MEMO

6-13 Serial Communication Instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
160		TCMP	P	Time Compare	○	○	
161		TZCP	P	Time Zone Compare	○	○	
162		TADD	P	Time Add	○	○	
163		TSUB	P	Time Subtract	○	○	
166		TRD	P	Read RTC data	○	○	
167		TWR	P	Set RTC data	○	○	○
170	D	GRY	P	Converts BIN to Gray code	○	○	
171	D	GBIN	P	Converts Gray code to BIN	○	○	

FNC 160 TCMP	P	TCMP P S1 S2 S3 S D	Time Compare	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
S_3					○	○	○	○	○	○	○	○		○	○	○
S									○	○	○					○
D		○	○	○												○

- S_1=0 ~ 23; S_2=0 ~ 59; S_3=0 ~ 59
- Both S and D occupy 3 consecutive devices respectively

```
        S1   S2   S3   S   D
  X20
|-| |---[TCMP  K8  K30  K20  D0  M0]
              \________/
               08:30:20
```

S_1 : The "Hour" value of the time set

S_2 : The "Minute" value of the time set

S_3 : The "Second" value of the time set

S : Time compare value

D : The storages of compare result

- Compare the setting values (Hours, Minutes and Seconds which are designated by S_1 ~ S_3) to the time value (specified by the head ID S of 3 consecutive data devices), and the result of Comparison is stored to D.
- When X20= "ON", the instruction will be performed.

	D0 (Hour)	
If 8: 30:20 >	D1 (Minute)	then M0= "ON".
	D2 (Second)	

	D0 (Hour)	
If 8: 30:20=	D1 (Minute)	then M1= "ON".
	D2 (Second)	

	D0 (Hour)	
If 8: 30:20 <	D1 (Minute)	then M2= "ON".
	D2 (Second)	

- The current time of the real time clock is stored in Special Registers D9013 ~ D9015.
  D9015 (Hour), D9014 (Minute), D9013 (Second)
- When X20= "OFF", the instruction will not be performed. M0 ~ M2 will remain the status before X20= "OFF".
- Please combine two of M0 ~ M2when the result ≤ , ≥ or ≠ is needed.
- If the content value of the register designated by S exceeding the time value required, it will be regarded as an operation error.

FNC 161 TZCP	P	TZCPP (S1) (S2) (S) (D)	Time Zone Compare	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1									○	○	○					○
S_2									○	○	○					○
S									○	○	○					○
● D		○	○	○												○
● $S_1 \leq S_2$	● All S_1, S_2, S and D will occupy 3 consecutive devices respectively															

X20 ─┤ ├─ TZCP D0 D10 D20 M0

(S1) (S2) (S) (D)

S_1 : Lower limit of the setting time period
S_2 : Upper limit of the setting time period
S : Time compare value
D : The storages of compare result

- The time compare value is defined by (S), it will be compared to the lower limit of the setting time period defined by (S1) and the upper limit of the setting time period defined by (S2). And then, the compare result will be stored in (D).
- When X20= "ON", the instruction will be performed.

(S1)		(S)	
D0 (Hour)		D20 (Hour)	
D1 (Minute)	>	D21 (Minute)	,M0= "ON"
D2 (Second)		D22 (Second)	

				(S2)	
D0 (Hour)		D20 (Hour)		D10 (Hour)	
D1 (Minute)	≤	D21 (Minute)	≤	D11 (Minute)	,M1= "ON"
D2 (Second)		D22 (Second)		D12 (Second)	

D20 (Hour)		D10 (Hour)	
D21 (Minute)	>	D11 (Minute)	,M2= "ON"
D22 (Second)		D12 (Second)	

- The current time of the real time clock is stored in Special Registers D9013 ~ D9015. D9015 (Hour), D9014 (Minute), D9013 (Second)
- When X20= "OFF", the instruction will not be performed. M0 ~ M2 will remain the status before X20= "OFF".
- When (S1) > (S2), the value of (S1) will become both Upper/Lower Limit to compares with (S).
- If the content value of the register designated by (S1), (S2) or (S1) exceeding the time value required, it will be regarded as an operation error.

FNC 162 TADD	P	TADDP S1 S2 D	Time Addition	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1									○	○	○					○
S_2									○	○	○					○
D									○	○	○					○

- All S_1, S_2, S and D will occupy 3 consecutive devices respectively

X20 —| |— TADD D0 D10 D20 (S1 S2 D)

S_1 : Summand of the time value

S_2 : Addend of the time value

D : The addition result

- The time value defined by S_1 is added to the time value defined by S_2 and the result is stored in the registers defined by D.
- When X20= "ON", the time addition is performed.

S1		S2		D
D0 8 (Hour)		D10 6 (Hour)		D20 15 (Hour)
D1 30 (Minute)	+	D11 35 (Minute)	=	D21 5 (Minute)
D2 0 (Second)		D12 30 (Second)		D22 30 (Second)
8:30:0		6:35:30		15:5:30

- If the result of the time addition is longer than 24 hours, then the Carry Flag M9022 will be set "ON" and D will display the value where 24 hours is subtracted from the total.

S1		S2		D	
D0 8 (Hour)		D10 20 (Hour)		D20 4 (Hour)	
D1 25 (Minute)	+	D11 10 (Minute)	=	D21 35 (Minute)	,And M9022= "ON"
D2 30 (Second)		D12 20 (Second)		D22 50 (Second)	
8:25:30		20:10:20		4:35:50	
				28-24=4	

- If the result of the time addition equals "0" (0 hour 0 min 0 sec), then the Zero Flag M9020 will be set "ON".
- If the content value of the register designated by S_1 or S_2 exceeding the time value required, it will be regarded as an operation error.

	FNC 163 TSUB	P	TSUBP S1 S2 D	Time Subtraction	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1									○	○	○					○
S2									○	○	○					○
D									○	○	○					○
● All S1, S2, S and D will occupy 3 consecutive devices respectively																

X20 —| |— TSUB D0 D10 D20 (S1 S2 D)

S1 : Minuend of the time value

S2 : Subtrahend of the time value

D : The subtraction result

- The time value defined by S1 is subtracted by the time value defined by S2 and the result is stored in the register defined by D.
- When X20= "ON", the time subtraction is performed.

S1		S2		D
D0 18(Hour)		D10 8(Hour)		D20 9(Hour)
D1 28(Minute)	−	D11 40(Minute)	=	D21 48(Minute)
D2 50(Second)		D12 20(Second)		D22 30(Second)
18:28:50		8:40:20		9:48:30

- If the result of the time subtraction is a negative, then the Borrow Flag M9021 will be set "ON" and D will display the value where the negative value is added to 24 hours.

S1		S2		D	
D0 6(Hour)		D10 20(Hour)		D20 10(Hour)	
D1 30(Minute)	−	D11 20(Minute)	=	D21 10(Minute)	,And M9021= "ON"
D2 20(Second)		D12 10(Second)		D22 10(Second)	
6:30:20		20:20:10		10:10:10	

(-4)+24=10

- If the result of the time subtraction equals "0" (0 hour 0 min 0 sec), then Zero Flag M9020 will be set "ON".
- If the content value of the register designated by S1 or S2 exceeding the time value required, it will be regarded as an operation error.

FNC 166 TRD	P	TRDP (D)	Time Read	M	VB	VH
				○	○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
D									○	○	○					○

● D will occupy 7 consecutive devices respectively

X20 —| |— TRD (D) D0

D : The subtraction result

- M Series PLC's M1-CPU1 can install the M-RTC, M-232R or M-485R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- VB Series PLC's Main Unit can install the VB-RTC, VB-MP1R or VB-DB1R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- Programmers do not need to memorize the location of real time clock is stored, they can use this instruction to read the current time and date of the real time clock and store the data to contiguous 7 registers which is specified by (D).
- When X20= "ON", as the diagram below, the data of the real time clock will be read and stored into designated registers D0 ~ D6.

Item	Special Register ID	Content Value of the RTC	Read and Store ⇨	
Year	D9018	2000 ~ 2099		D0
Month	D9017	1 ~ 12		D1
Day	D9016	1 ~ 31		D2
Hour	D9015	0 ~ 23		D3
Minute	D9014	0 ~ 59		D4
Second	D9013	0 ~ 59		D5
Week	D9019	0 ~ 6		D6

- The content value of D9019=0 represents Sunday
  The content value of D9019=1 represents Monday
  The content value of D9019=2 represents Tuesday
  The content value of D9019=3 represents Wednesday
  The content value of D9019=4 represents Thursday
  The content value of D9019=5 represents Friday
  The content value of D9019=6 represents Saturday

FNC 167 TWR	P	TWRP (S)	Time Write	M	VB	VH
				○	○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S									○	○	○					○

```
      X20          (S)
 |--| |-------[TWR  D0]
```

S : The source registers which store the new current value of the real time clock

- M Series PLC's M1-CPU1 can install the M-RTC, M-232R or M-485R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- VB Series PLC's Main Unit can install the VB-RTC, VB-MP1R or VB-DB1R expansion card. After one of those expansion card has been installed, the PLC will be provide with the real time clock functions. The real time clock has seven sets of data such as year, month, day, hour, minute, second and week, the data will be stored in Special Register D9013 ~ D9019.
- When X20 = "ON", as the diagram below, the data in designated source registers D0 ~ D6 will be read and reset the current value of real time clock.

(S)	Content Value of the RTC		Special Register ID		Content Value of the Real Time Clock
D0	2000 ~ 2099	Rewrite	D9018	Rewrite	Year
D1	1 ~ 12		D9017		Month
D2	1 ~ 31		D9016		Day
D3	0 ~ 23		D9015		Hour
D4	0 ~ 59		D9014		Minute
D5	0 ~ 59		D9013		Second
D6	0 ~ 6		D9019		Week

- The content value (0 ~ 6) of D6 represents Sunday, Monday...Saturday.
- The content value of the source registers (defined by (S)) exceeding the valid range (as shown above), it will be regarded as an operation error.
- Also can use the program develop software Ladder Master to perform setting of the real time clock (rewrite RTC data).

D	FNC 170 GRY	P	DGRYP (S) (D)	Converts BIN to Gray Code	M	VB	VH
					○	○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

- For a 16-bit instruction, S= 0 ~ 32767
- For a 32-bit instruction, S= 0 ~ 2147483647

X20 — GRY D0 K4Y20 [(S) (D)]

S : Source device (Gary Code)

D : The destination device where the converted BIN value is stored

- When the instruction is performed, it converts the content BIN value of the source devices (designated by (S)) into Gary Code and transfers they to the designated devices (D).
- When X20= “ON”, the content value of (D0) will be converted to Gary Code and then 16 output points (Y0 ~ Y37) will be exported to the terminals.

BIN Value 6,513

b15															b0	
0	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	D0

X20 = ON

Y37															Y20	
0	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	K4Y20

Gary Code 6,513

- The valid range of (S) is shown below. Any value exceeding such a range will be regarded as an operation error.
  For a 16-bit instruction: 0 ~ 32,767
  For a 32-bit instruction: 0 ~ 2,147,483,647

D	FNC 171 GBIN	P	DGBINP S D	Converts Gray Code to BIN	M	VB	VH
					○	○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D						○	○	○	○	○	○	○		○		○

- For a 16-bit instruction, S= 0 ~ 32767
- For a 32-bit instruction, S= 0 ~ 2147483647

X0 — GBIN K4X20 D0 (S = K4X20, D = D0)

S : Source device where the Gary Code is stored

D : The destination device where the converted BIN value is stored

- When the instruction is performed, it converts the content Gary Code of the source devices (designated by S) into BIN Value and transfers they to the designated device D.
- This instruction is always used to convert the code from an Absolute Rotary Encoder (which is connected to the PLC's input terminal and generally uses the Gary Code) to a BIN Value and transfer it to the register in the PLC.
- When X0= "ON", the code of an Absolute Rotary Encoder connected to 16 output points (Y20 ~ Y37) will be converted to BIN Value and then transfered to D0.

Gary Code 6,513

X37															X20	
0	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	K4X20

X0=ON

b15															b0	
0	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	D0

BIN Value 6,513

- The valid range of S is shown below. Any value exceeding such a range will be regarded as an operation error.
For a 16-bit instruction: 0 ~ 32,767
For a 32-bit instruction: 0 ~ 2,147,483,647

MEMO

6-14 In-line Comparisons

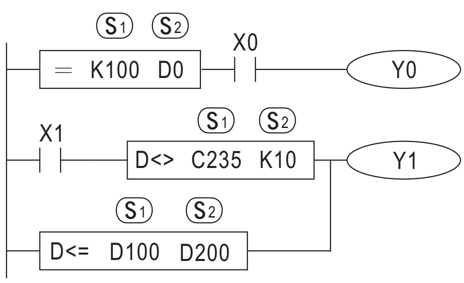
FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
224	D	LD=		Initial comparison contact. Active when (S1)=(S2)		○	○
225	D	LD>		Initial comparison contact. Active when (S1)>(S2)		○	○
226	D	LD<		Initial comparison contact. Active when (S1)<(S2)		○	○
228	D	LD<>		Initial comparison contact. Active when (S1)≠(S2)		○	○
229	D	LD<=		Initial comparison contact. Active when (S1)≤(S2)		○	○
230	D	LD>=		Initial comparison contact. Active when (S1)≥(S2)		○	○
232	D	AND=		Serial comparison contact. Active when (S1)=(S2)		○	○
233	D	AND>		Serial comparison contact. Active when (S1)>(S2)		○	○
234	D	AND<		Serial comparison contact. Active when (S1)<(S2)		○	○
236	D	AND<>		Serial comparison contact. Active when (S1)≠(S2)		○	○
237	D	AND<=		Serial comparison contact. Active when (S1)≤(S2)		○	○
238	D	AND>=		Serial comparison contact. Active when (S1)≥(S2)		○	○
240	D	OR=		Parallel comparison contact. Active when (S1)=(S2)		○	○
241	D	OR>		Parallel comparison contact. Active when (S1)>(S2)		○	○
242	D	OR<		Parallel comparison contact. Active when (S1)<(S2)		○	○
244	D	OR<>		Parallel comparison contact. Active when (S1)≠(S2)		○	○
245	D	OR<=		Parallel comparison contact. Active when (S1)≤(S2)		○	○
246	D	OR>=		Parallel comparison contact. Active when (S1)≥(S2)		○	○

					M	VB	VH
D	FNC 224 LD=		D= S1 S2	Initial comparison contact. Active when (S1)=(S2)		○	○
D	FNC 225 LD>		D> S1 S2	Initial comparison contact. Active when (S1)>(S2)		○	○
D	FNC 226 LD<		D< S1 S2	Initial comparison contact. Active when (S1)<(S2)		○	○
D	FNC 228 LD<>		D<> S1 S2	Initial comparison contact. Active when (S1)≠(S2)		○	○
D	FNC 229 LD<=		D<= S1 S2	Initial comparison contact. Active when (S1)≤(S2)		○	○
D	FNC 230 LD>=		D>= S1 S2	Initial comparison contact. Active when (S1)≥(S2)		○	○
D	FNC 232 AND=		D= S1 S2	Serial comparison contact. Active when (S1)=(S2)		○	○
D	FNC 233 AND>		D> S1 S2	Serial comparison contact. Active when (S1)>(S2)		○	○
D	FNC 234 AND<		D< S1 S2	Serial comparison contact. Active when (S1)<(S2)		○	○
D	FNC 236 AND<>		D<> S1 S2	Serial comparison contact. Active when (S1)≠(S2)		○	○
D	FNC 237 AND<=		D<= S1 S2	Serial comparison contact. Active when (S1)≤(S2)		○	○
D	FNC 238 AND>=		D>= S1 S2	Serial comparison contact. Active when (S1)≥(S2)		○	○
D	FNC 240 OR=		D= S1 S2	Parallel comparison contact. Active when (S1)=(S2)		○	○
D	FNC 241 OR>		D> S1 S2	Parallel comparison contact. Active when (S1)>(S2)		○	○
D	FNC 242 OR<		D< S1 S2	Parallel comparison contact. Active when (S1)<(S2)		○	○
D	FNC 244 OR<>		D<> S1 S2	Parallel comparison contact. Active when (S1)≠(S2)		○	○
D	FNC 245 OR<=		D<= S1 S2	Parallel comparison contact. Active when (S1)≤(S2)		○	○
D	FNC 246 OR>=		D>= S1 S2	Parallel comparison contact. Active when (S1)≥(S2)		○	○

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2					○	○	○	○	○	○	○	○		○	○	○

S1 : The first source value of the comparison

S2 : The second source value of the comparison



When the content value of D0=K100 and X0= "ON", then Y0= "ON".

When X1= "ON" and the content value of C235≠K10, then Y1= "ON".

OR, the content value of (D101, D100) is≤the content value of (D201,D200), then Y1= "ON".

MEMO

MEMO

APPENDIX

A. High-Speed I/O Function of VB1 Series

The VB1 Series PLCs are the logic controllers specially designed for high-speed input and positioning control applications in the VB-PLC family.
Talking about high-speed input functions, besides the External Input Interrupt , Speed Detection and C235 ~ C255 Software High-Speed Counter functions the VB1 series PLCs originally have as VB-PLC family member, two other new Hardware High-Speed Counters (HHSC) are added also. These two Hardware High-Speed Counters not only have counting frequency of AB phrase 200 KHz high, they also have Hardware Compare Interrupt function, which can do precise positioning control.
Talking about positioning control, VB1-PLCs provide 4 points high-speed pulse output, which support four independent axis positioning control at the same time. Within them, output points Y0 and Y1 can output 20 KHz pulse; Y2 and Y3 can output pulse up to 200 KHz. It creates the best economic benefit for multi-axis positioning control applications.

A-1 High-Speed Output Functions of VB1

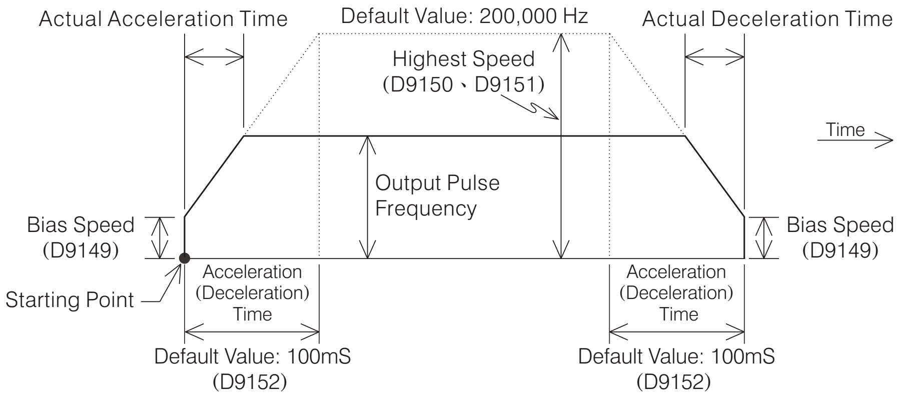
A-1-1 Positioning Control Instructions

VB1-PLCs not only have multi-points high-speed pulse output ability, but also provide many positioning control instructions, which help the user to accomplish positioning control easily.

FNC No.	Instruction Title			Function
	D		P	
155	D	ABS		Reads the absdute position from a servo motor driver
156	D	ZRN		Home (zero point) position Return
157	D	PLSV		Variable frequency pulse output
158	D	DRVI		To drive position incrementally
159	D	DRVA		To drive position by absolute measurement
151	D	DVIT		One-speed Interrupt Constant Quantity Feed
153	D	LIR		Relatively Linear Interpolation
154	D	LIA		Absolutely Linear Interpolation

Common Guidelines for using positioning control instructions on VB1 series PLCs:

- FNC155 (ABS) ~ FNC159 (DRVA) instructions are only supported by VB1 series PLCs, they are not supported by the rest of VB series PLCs like VB0 and VB2.
- FNC156 (ZRN) ~ FNC159 (DRVA) instructions all belong to positioning control instructions. They can be used for unlimited times in the program, but please note that DO NOT drive the same output point repeatedly.
- Before running the ZRN, DRVI and DRVA instructions, the D9149 ~ D9152 related parameters should be configured first.
- Users are recommended to use DRVI instruction for positioning control instead of PLSY and PLSR instructions for the VB1 series PLCs
- The Y0 ~ Y3 are high-speed output points, Load Voltage DC 5 ~ 24 V, Load Current 0 ~ 100 mA.
  The highest output pulse frequency of Y0 and Y1 is 20 KHz. The highest output pulse frequency of Y2 and Y3 is 200 KHz.
  Y 0 ~ Y3 can be used as normal output points, Load Current 0 ~ 0.5 A.
- When Y0 ~ Y3 are used for high-speed pulse output, they can be used together with any output point output direction signals. And the pulse input form of the servo or the step motor must be set to "pulse train + direction".
- The parameters of positioning control operation are shown in the diagram below.



- Related Components of Positioning Control Instructions.

  For components with symbol "■" in the list below, their flags cannot be driven by instructions and no data can be written to the register.

Coil ID. No.	Instruction of Function	
M9140	Output clear signal to servo motor drive when home positioning done.	
M9141	Interrupt signal logic reverse flag for Y0.	For DVIT instruction only. OFF: normal logic (trigger by risen edge); ON: reverse logic (trigger by fallen edge).
M9142	Interrupt signal logic reverse flag for Y1.	
M9143	Interrupt signal logic reverse flag for Y2.	
M9144	Interrupt signal logic reverse flag for Y3.	
M9145	Make Y0 stop pulse output immediately.	
M9146	Make Y1 stop pulse output immediately.	
M9147	Make Y2 stop pulse output immediately.	
M9148	Make Y3 stop pulse output immediately.	
■ M9149	Y0 pulse output monitor, ON means in pulse output.	
■ M9150	Y1 pulse output monitor, ON means in pulse output.	
■ M9151	Y2 pulse output monitor, ON means in pulse output.	
■ M9152	Y3 pulse output monitor, ON means in pulse output.	

Register ID.	Instruction of Function	
D9140	Lower 16 bits	Current value register for Y0 output positioning control instruction.
D9141	Upper 16 bits	
D9142	Lower 16 bits	Current value register for Y1 output positioning control instruction.
D9143	Upper 16 bits	
D9144	Lower 16 bits	Current value register for Y2 output positioning control instruction.
D9145	Upper 16 bits	
D9146	Lower 16 bits	Current value register for Y3 output positioning control instruction.
D9147	Upper 16 bits	
D9148	To assign the input points of the interrupt signals of DVIT instruction.(the default value is H3210)	
D9149	Bias speed when executing ZRN, DRVI , DRVA and DVIT instructions, but when this value exceeds 1/10 of the highest speed (D9151, D9150), the 1/10 of the highest speed will be used as the bias speed.	
D9150	Lower 16 bits	Highest speed when executing ZRN, DRVI , DRVA and DVIT instructions. Initial value 200,000 Hz, configurable range 10 ~ 200, 000 Hz. When this value exceeds the maximum acceptable value, the maximum acceptable value will be used as reference.
D9151	Upper 16 bits	
D9152	The acceleration/deceleration time from starting speed to highest speed when the ZRN, DRVI , DRVA and DVIT instructions execute. Initial value 100 mS, configurable range 50 ~ 5, 000 mS.	

D	FNC 155 ABS		DABS (S) (D1) (D2)	Reads the absliute Position from a Servo Motor Drive	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S	○	○	○	○												○
D1		○	○	○												○
D2						○	○	○	○	○	○	○		○		○

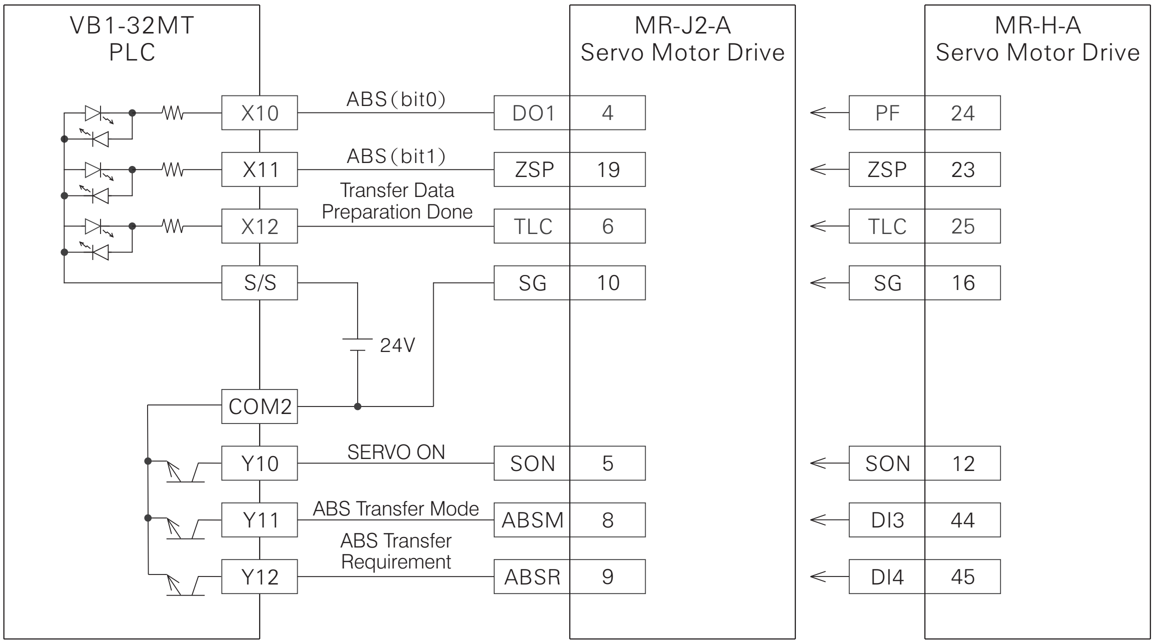
- S occupies 3 points
- D1 occupies 3 points
- D2 occupies 2 points

S : Signal from the servo motor drive

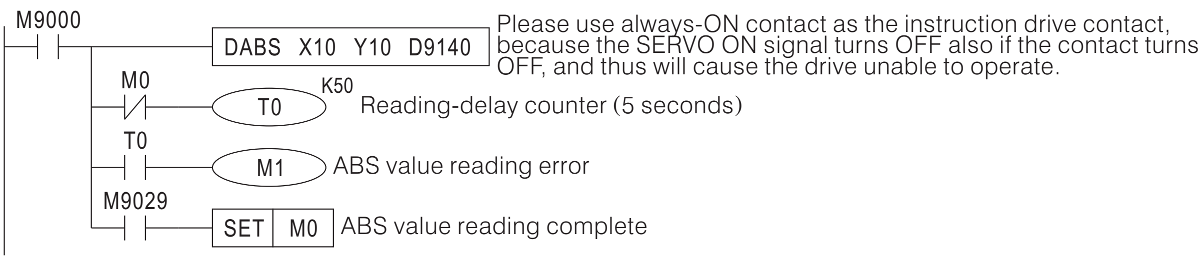
D1 : Signal to the servo motor drive

D2 : Stores the reading result

- The actual wiring of the above example is shown as below.



- Since the reading result of this instruction is the current position of the servo motor drive, which corresponds to the (D9141, D9140), (D9143, D9142), (D9145, D9144), (D9147, D9146) of the controller, the (D2) will usually assign D9140 (output axis of Y0), D9142 (output axis of Y1), D9144 (output axis of Y2) or D9146 (output axis of Y3).
- This instruction reads the current position of Mitsubishi machine MR-H or MR-J2 servo motor drive (with built-in absolute position detection function).
- Reading starts when X20 turns from OFF → ON. And when the reading completes, M9029 will be ON for a scan time. If X20 turns OFF during the reading process, the reading will be aborted.
- This instruction is 32-bits. Be sure to input as DABS.
- Example Program



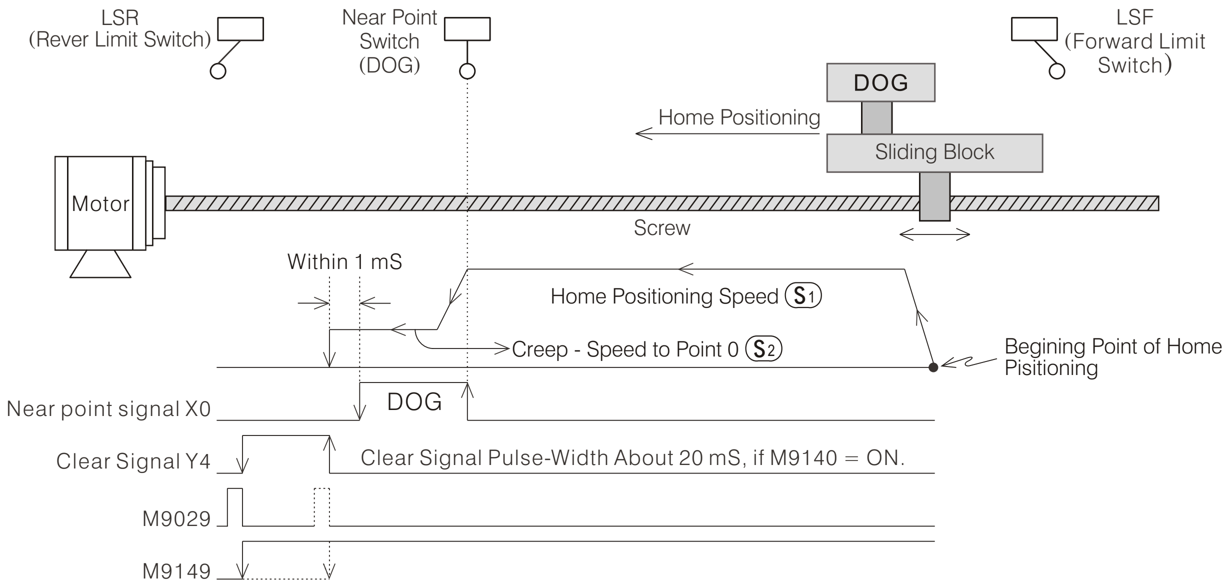
D	FNC 156 ZRN		DZRN S1 S2 S3 D	Home (zero) Position Return	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
S_3	○	○	○	○												○
D		○														○

- D=Y0 ~ Y3
- When D=Y0 or Y1, S_1 = 10 ~ 20,000, S_2 = 10 ~ 20,000
- When D=Y2 or Y3, 16-Bit instruction, S_1=10 ~ 32,767, S_2=10 ~ 32,767
  32-Bit instruction, S_1=10 ~ 200,000, S_2=10 ~ 32,767

```
X20          S1      S2   S3 D
--| |-----[ZRN K10,000 K500 X0 Y0]
```

S_1 : home positioning speed
S_2 : Creep speed to point zero
S_3 : near point signal (DOG)
D : pulse output point



Near point signal should use X0 ~ X7 input points as possible, to avoid inaccuracy caused by scan time.
When M9140=OFF, Clear Signal is not produced, and M9029 and M9149 show dash-line signals.
When M9140=ON, Clear Signal is not produced, and M9029 and M9149 show solid-line signals.

- When X20=ON, ZRN instruction starts. The sliding block moves in S1 home positioning speed (10,000Hz) towards S3 near point (X0) direction. When S3 near point signal turns from OFF → ON, the sliding block changes to speed S2 point zero creep speed (500 Hz) and continues moving. When S3 signal turns from ON → OFF, pulse output will be stopped, and the current value register (D9141, D9140) which corresponds to the output point Y0 will be cleared to 0. Y0 pulse output monitor M9149 will then turn OFF and the operation complete flag M9029 will be ON for a scan time. By now, the home positioning completes.
- When X20=OFF→ON, ZRN instruction decides the home positioning process according to the D9149 deviation speed, (D9151, D9150) highest speed, D9152 deceleration time, S1 home positioning speed and S2 point zero creep speed. During this instruction execution , all parameter configuration changes are ineffective. So the D9149 ~ D9152 parameters configurations should be done before the instruction starts.
- When the home positioning instruction executes, if X20 turns OFF, the execution will be aborted immediately.
- If M9140 is set to ON, after the home positioning completes, the current value of drive is cleared to 0 according to the servo motor drive clear signal which corresponds to the pulse output point. The clear signal pulse-width is about 20 mS. The corresponding clear signals for output Y0 ~ Y3 are output by Y4 ~ Y7.
- The Near Point (DOG) Search function is not supported. So the home positioning action should be started from the front side of the near point signal.

D	FNC 157 PLSV		D PLSV (S) (D1) (D2)	Variable Speed Pulse Output	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D_1		○														○
D_2		○	○	○												○

- D_1=Y0 ~ Y3
- When D_1=Y0 or Y1, S = 1 ~ 20,000, or = -1 ~ -20,000
- When D1=Y2 or Y3, the configuration range of S is as below:
  16-Bit instruction, S=1 ~ 32,767 or -1 ~ -32,768; 32-Bit instruction, S=1 ~ 200,000 or -1 ~ -200,000

X20 —| |— [PLSV D0 Y0 Y10] (S) (D1) (D2)

S : output pulse frequency

D_1: pulse output point

D_2: direction signal output point

- When X20=ON, Y0 outputs pulse frequency as set by the value of D0.
  Y10 outputs direction signal. When D0 value > 0, Y10=ON means positive rotation.
  When D0 value < 0, Y10=OFF means negative rotation.
- (S) value can be changed during the pulse output, to change the output frequency
- The pulse output stops immediately when the condition contact X20 turns OFF during the pulse output.
- Acceleration/deceleration function is not supported when the frequency changes, so if necessary, please use FNC67(RAMP) instruction and change the (S) value to gradually increase/decrease the frequency.
- Double check the pulse output monitor flag (M9149 ~ M9152) before running this instruction, if the corresponding flag signal is ON, the instruction will not start.
- The (D2) direction signal is decided by the positive/negative sign of the pulse output frequency (S) value.
  If (S) value > 0, (D2) direction signal is ON, means positive rotation, and the current value register data will increase.
  If (S) value < 0, (D2) direction signal is OFF, means negative rotation, and the current value register data will decrease.
- Since the output frequency of Y0 and Y1 are 20 KHz at most, when (D1) is set to be Y0 or Y1, the configuration range of (S) is 1 ~ 20,000 or -1 ~ -20,000.
- Since the output frequency of Y2 and Y3 are 200 KHz at most, when (D1) is set to be Y2 or Y3, the configuration range of (S) is:
  16-bits instruction: 1 ~ 32,767 or -1 ~ -32,768
  32-bits instruction: 1 ~ 200,000 or -1 ~ -200,000.

D	FNC 158 DRVI		DDRVI (S1) (S2) (D1) (D2)	To Drive Position Incrementally	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D_1		○														○
D_2		○	○	○												○
● D_1=Y0 ~ Y3																
● 16-Bit instruction, S_1=-32,768 ~ 32,767; 32-Bit instruction, S_1=-2,147,483,648 ~ 2,147,483,647																
● When D_1=Y0 or Y1, S_2=10 ~ 20,000																
● When D_1=Y2 or Y3, 16-Bit instruction, S_2=10 ~ 32,767; 32-Bit instruction, S_2=10 ~ 200,000																

```
                   (S1)    (S2)  (D1) (D2)
  X20
 |-| |----[DRVI  K20000  K10000  Y0  Y10]
```

S_1 : output pulse number
S_2 : output pulse frequency
D_1 : pulse output point
D_2 : direction signal output point

- When X20=ON, Y0 outputs 20,000 pulses with frequency 10,000 Hz. Y10=ON means positive rotation. (D9141, D9140) current value register data will increase to 20,000.
- When X20=OFF → ON, DRVI instruction decides the relative positioning process according to the D9149 bias speed, (D9151, D9150) highest speed, D9152 acceleration/deceleration time, (S1) output pulse number and (S2) output pulse frequency. During the instruction execution, all parameter changes will be treated as ineffective. So the D9149 ~ D9152 parameter configurations should be done before the instruction starts.
- When the output pulse number set by (S1) is reached, the execution complete flag M9029 will be ON for a scan time.
- When the condition contact X20 turns OFF during the pulse output, the operation will be decelerated to stop, but the execution complete flag M9029 will not take action then.
- Double check the pulse output monitor flag (M9149 ~ M9152) before running this instruction, if the corresponding flag signal is ON, the instruction will not start.
- The (D2) direction signal is decided by the positive/negative sign of the output pulse number (S1) value.
  If (S1) value > 0, (D2) direction signal is ON, means positive rotation, and the current value register data will increase.
  If (S1) value < 0, (D2) direction signal is OFF, means negative rotation, and the current value register data will decrease.
- Since the output frequency of Y0 and Y1 are 20 KHz at most, when (D1) is set to be Y0 or Y1, the configuration range of (S2) is 10 ~ 20,000
- Since the output frequency of Y2 and Y3 are 200 KHz at most, when (D1) is set to be Y2 or Y3, the configuration range of (S2) is:
  16-bits instruction: 10 ~ 32,767
  32-bits instruction: 10 ~ 200,000

D	FNC 159 DRVA		DDRVA S1 S2 D1 D2	To Drive Position by Absolute Measurement	M	VB1	VH
						○	

<table>
<tr><th rowspan="2">Operand</th><th colspan="16">Devices</th></tr>
<tr><th>X</th><th>Y</th><th>M</th><th>S</th><th>KnX</th><th>KnY</th><th>KnM</th><th>KnS</th><th>T</th><th>C</th><th>D</th><th>SD</th><th>P</th><th>V,Z</th><th>K,H</th><th>VZ index</th></tr>
<tr><td>S1</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><td></td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>S2</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><td></td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>D1</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><td></td><td></td><td></td><td>○</td></tr>
<tr><td>D2</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><td></td><td></td><td></td><td>○</td></tr>
<tr><td colspan="17">● D1＝Y0～Y3</td></tr>
<tr><td colspan="17">● 16-Bit instruction, S1＝-32,768～32,767; 32-Bit instruction, S1＝-2,147,483,648～2,147,483,647</td></tr>
<tr><td colspan="17">● When D1＝Y0 or Y1, S2＝10～20,000</td></tr>
<tr><td colspan="17">● When D1＝Y2 or Y3, 16-Bit instruction, S2＝10～32,767; 32-Bit instruction, S2＝10～200,000</td></tr>
</table>

```
                      S1     S2     D1  D2
   X20
|--| |----[DRVA  D100  K10000  Y0  Y10]
```

S1 : target position

S2 : output pulse frequency

D1 : pulse output point

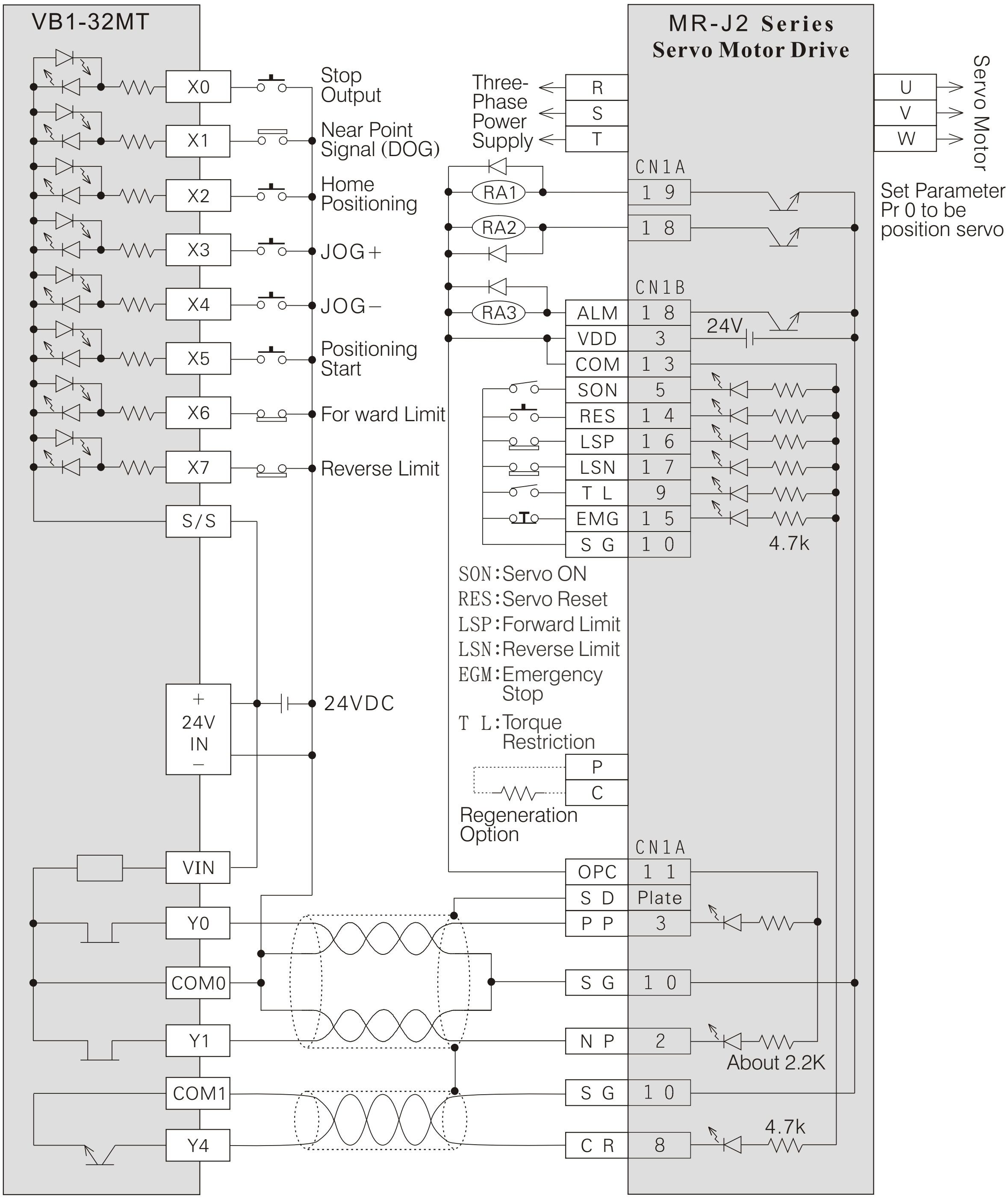
D2 : direction signal output point

- When X20＝ON, Y0 outputs pulse with frequency 10,000 Hz, until the value of (D9141, D9140) current value registers are is equal to the value of target position (D101, D100), which means the positioning point is reached, and Y0 will then stop output.
  The status of Y10 output point changes with the relation of the current position (D9141, D9140 current value register) and S1 target position (D101, D100).
  When target position > current position, Y10 is ON, means positive rotation.
  When target position < current position, Y10 is OFF, means negative rotation.
- When X20＝OFF → ON, DRVA instruction decides the absolute measurement positioning process according to the D9149 deviation speed, (D9151, D9150) highest speed, D9152 acceleration/deceleration time, S1 target position and S2 output pulse frequency.
  During the instruction execution, any parameter configuration change will be treated as ineffective. So the D9149～D9152 parameter configurations should be done before the instruction starts.
- When the target position set by S1 is reached, the execution complete flag M9029 will be ON for a scan time.
- When the condition contact X20 turns OFF during the pulse output, the operation will be decelerated to stop, but the execution complete flag M9029 will not take action then.
- Please confirm the pulse output monitor flag (M9149～M9152) before running this instruction, if the corresponding flag signal is ON, the instruction will not start.
- The D2 direction signal is decided by the positive/negative sign of the target position minus by the current position value result.
  If S1 value > current position, D2 direction signal is ON, means positive rotation, and the current value register data will increase.
  If S1 value < current position, D2 direction signal is OFF, means negative rotation, and the current value register data will decrease.
- Since the output frequency of Y0 and Y1 are 20 KHz at most, when D1 is set to Y0 or Y1, the configuration range of S2 is 10～20,000.
- Since the output frequency of Y2 and Y3 are 200 KHz at most, when D1 is set to Y2 or Y3, the configuration range of S2 is:
  16-Bits instruction: 10～32,767
  32-Bits instruction: 10～200,000

A-1-2 Positioning Control Programming Example

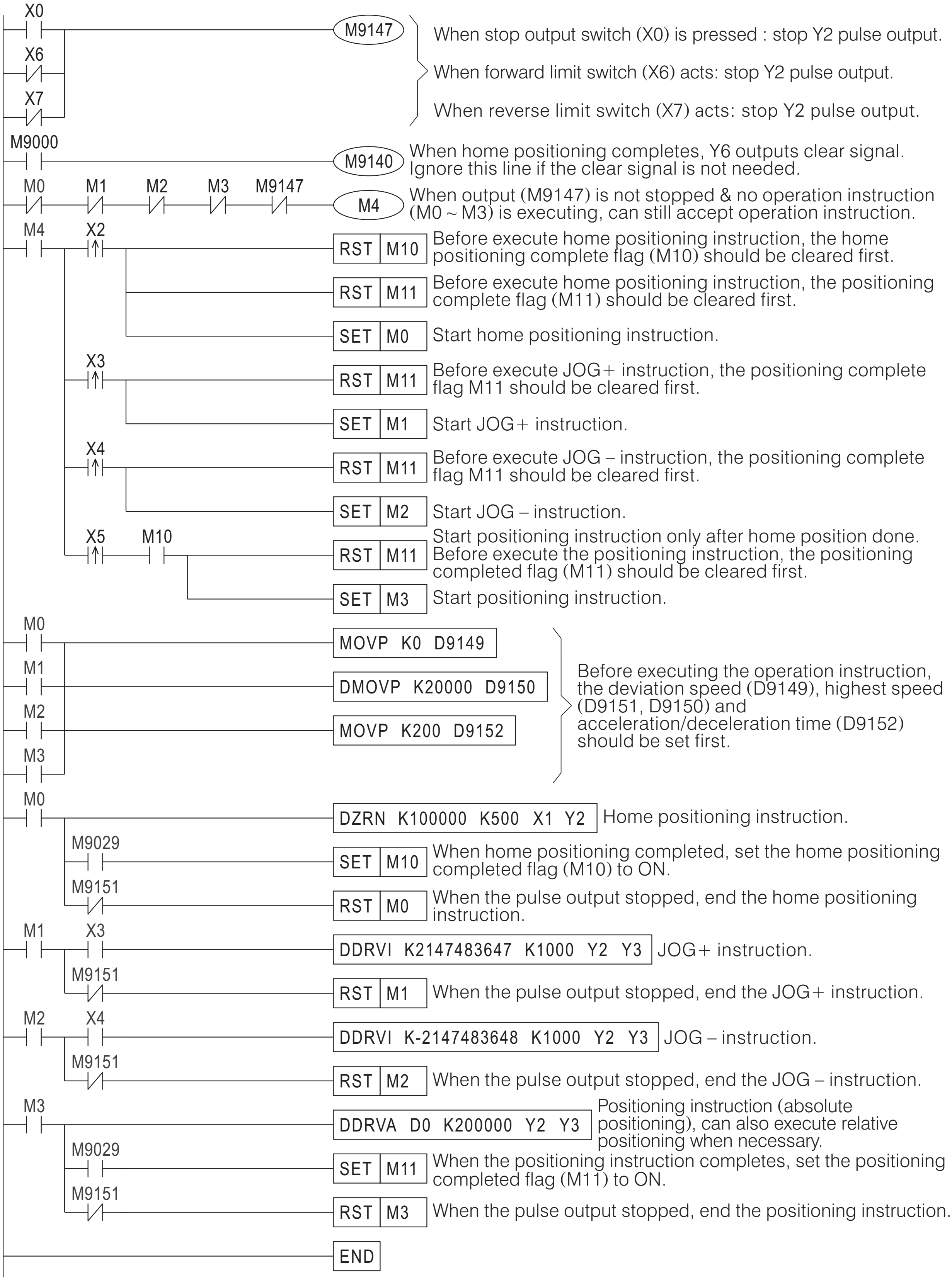
This positioning control system example is composed by a VB1 Main Unit and the Mitsubishi Servo Motor (MR-J2). To make it easier to understand, here only use the single axis control as example. For multi-axis control applications, please take note that before start the positioning instruction, the corresponding parameters (D9149 ~ D9152) of this axis should be configured first.

Wiring Example of VB1 Main Unit with Mitsubishi Servo Motor (MR-J2)



- The above diagram shows the connection position of MR-J2 drive initial parameters, this connection position of MR-J2 drive can be changed by changing the expansion parameter value.

Programming Example:



A-1-3 PLSY and PLSR Pulse Output Instruction

D	FNC 57 PLSY		─┤ ├── D PLSY S1 S2 D	Pulse Output	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D		○														○

- D = Y0 ~ Y3 • 16-Bit instruction, S_2 = 0 ~ 32,767 • 32-Bit instruction, S_2 = 0 ~ 2,147,483,647
- D = Y0 or Y1, S_1 = 2 ~ 20,000 • D = Y2 or Y3, S_1 = 2 ~ 200,000

```
              S1    S2   D
  X20
──┤ ├────[PLSY K500 D100 Y0]
```

S_1 : pulse output frequency
S_2 : pulse output number
D : pulse output point

- This introduction is only applicable for VB1 series PLCs.
- When X20=ON, Y0 outputs D100 number of pulses with frequency 500 Hz (500 pulses per second).
  S1 specifies the frequency of output pulse.
  S2 specifies the number of output pulse.
  For 16-Bit instruction, the configurable range is 0 ~ 32,767 pulses.
  For 32-Bit instruction, the configurable range is 0 ~ 2,147,483,647 pulses.
  When S2 is set to 0, it means output continuously with no pulse number limit.
  D specifies the pulse output point. The output point can only be set to Y0 ~ Y3.
- The pulse-width of output signal is shared by ON and OFF of 50% each. The CPU send pulse instantly to Y0 and Y1 using interrupt insertion mode, and send pulse instantly to Y2 and Y3 output device using dedicated hardware circuit.
- When the output pulse number set by S2 is reached, the execution complete flag M9029 will be ON for a scan time.
- Special registerD9141 (upper 16-Bits), D9140 (lower 16-Bits) will show number of pulses output from Y0 for PLSY and PLSR instructions.
  Special registerD9143 (upper 16-Bits), D9142 (lower 16-Bits) will show number of pulses output from Y1 for PLSY and PLSR instructions.
  Special registerD9145 (upper 16-Bits), D9144 (lower 16-Bits) will show number of pulses output from Y2 for PLSY and PLSR instructions.
  Special registerD9147 (upper 16-Bits), D9146 (lower 16-Bits) will show number of pulses output from Y3 for PLSY and PLSR instructions.
  The data value of registers above can be cleared using DMOV K0 D91□□ instruction.
- When the condition contact X20 turns OFF during the pulse output, the pulse output will be aborted immediately, and the pulse output point will be OFF. When X20 turns ON again, the output will start over from the first pulse.
- During the execution of this instruction, the data value of S1 can be changed by program, but the change made to S2 will be ineffective.
- There is no limit to how many times this instruction can be used in program. Y0 ~ Y3 can output pulse at the same time.

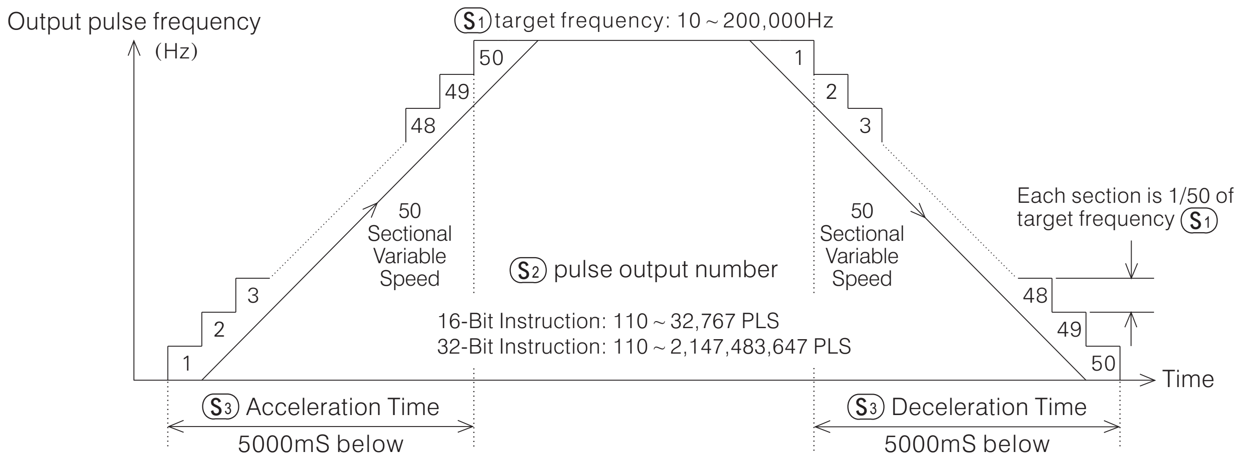
D	FNC 59 PLSR		D PLSR S_1 S_2 S_3 D	Pulse Output with Acceleration/Deceleration	M	VB1	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
S_3					○	○	○	○	○	○	○	○		○	○	○
D		○														○

- D=Y0~Y3 • 16-Bit instruction, S_2=110~32,767 • 32-Bit instruction, S_2=110~2,147,483,647 • S_3=50~5,000
- D=Y0 or Y1, S_1=10~20,000 • D=Y2 or Y3, S_1=10~200,000

```
X20            S1     S2     S3    D
-| |---[PLSR  K1000  D100  K1000  Y0]
```

S_1 : target output frequency
S_2 : pulse output number
S_3 : acceleration/deceleration time (Unit mS)
D : pulse output point



- This introduction is only applicable for VB1 series PLCs.
- When X20=ON, Y0 outputs D100 number of pulses in the way as shown by the diagram above.
  S_1 specifies the target output frequency.
  S_2 specifies the number of output pulse.
  For 16-Bits instruction, the configurable range is 110~32,767 pulses.
  For 32-Bits instruction, the configurable range is 110~2,147,483,647 pulses.
  S_3 specifies the acceleration/deceleration time, unit mS.
  The configurable range is 50~5,000 mS.
  D specifies the pulse output point. The output point can only be set to Y0~Y3.
- This instruction uses 50-sectional variable speed to reach the target frequency. So each changing section is 1/50 of the target frequency, the starting frequency is also 1/50 of the target frequency. For example, when the target frequency is set to 100,000 Hz, the starting frequency will be 2,000 Hz.
- When the output pulse number set by S_2 is reached, the execution complete flag M9029 will be ON for a scan time.

- Special registerD9141 (upper 16-Bits), D9140 (lower 16-Bits) will show number of pulses output from Y0 for PLSY and PLSR instructions.

  Special registerD9143 (upper 16-Bits), D9142 (lower 16-Bits) will show number of pulses output from Y1 for PLSY and PLSR instructions.

  Special registerD9145 (upper 16-Bits), D9144 (lower 16-Bits) will show number of pulses output from Y2 for PLSY and PLSR instructions.

  Special registerD9147 (upper 16-Bits), D9146 (lower 16-Bits) will show number of pulses output from Y3 for PLSY and PLSR instructions.

  The data value of registers above can be cleared using DMOV K0 D91□□ instruction.

- When the condition contact X20 turns OFF during pulse output, the pulse output will be aborted immediately, and the pulse output point will be OFF. When X20 turns ON again, the output will start over from the first pulse.

- During the execution of this instruction, any parameter change will be treated as ineffective.

- There is no limit to how many times this instruction can be used in program. Y0~Y3 can output pulse at the same time.

A-2 High-Speed Input Functions of VB1

The X0 ~ X7 input points of VB1 series PLC have many high-speed input functions like high-speed counting, external interrupt insertion and speed detection which are exactly the same as the VB0, VB2 series. Besides, the X0 ~ X7 input points of VB1 series also provide two hardware high-speed counters (HHSC) with counting frequency of 200 KHz high.
VB0 and VB2 series use interrupt insertion method to accomplish high-speed counting, and the processing speed is restricted by the processing efficiency of the CPU. But the VB1 series HHSC uses hardware circuit to do counting, which does not affect the CPU efficiency, and the counting speed is only restricted by the reaction time of the hardware circuit, thus, it can provide a counting frequency of 200 KHz high.
Both the interrupt insertion high-speed input method and the HHSC function occupy X0 ~ X7 input points, so once X0 ~ X7 is used by any high-speed working mode, they cannot be used by other functions.
The introduction below specifies how to use the interrupt insertion method and the HHSC functions.

A-2-1 High-Speed Input Function of Interrupt Insertion

For these functions, VB1 series work exactly in the same way as the VB0 and VB2 series do, and they have the following types:

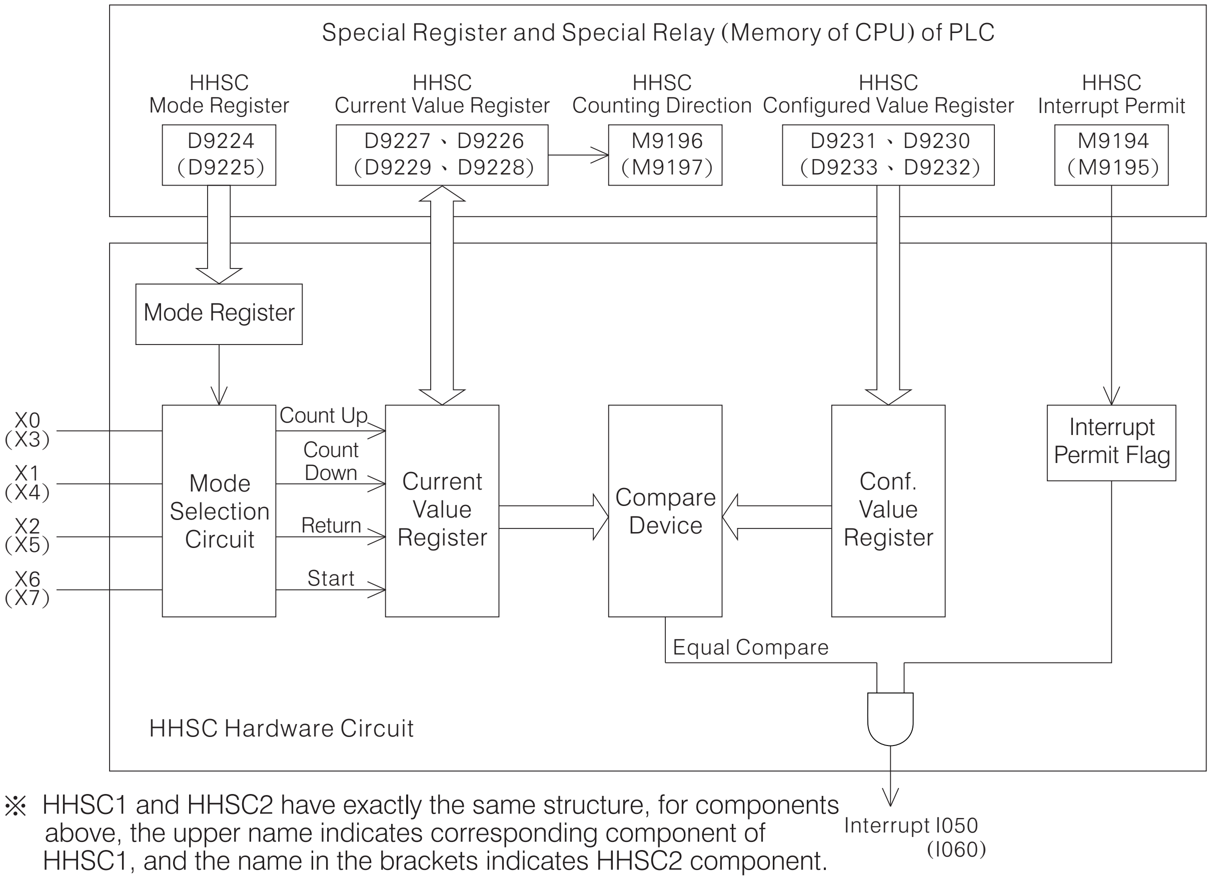
(1) For C235 ~ C255 High-Speed counters, please refer to “2-7 High-Speed Counter” for detailes.

(2) For external interrupt insertion, please refer to “2-11-2 Interrupt Pointer (I)” and FNC3 (IRET) ~ FNC5 (DI) for detailed introduction.

(3) For speed detection, please refer to FNC56 (SPD) for detailed introduction.

A-2-2 HHSC Function of Hardware High-Speed Counter

HHSC uses hardware circuit to accept high-speed pulse input & accomplish the high-speed counting task. HHSC is a 32-Bits up/down counter, it has latched function and configured value comparison function, and when the current value is equal to the configured value, it will send out high-speed counter interrupt signal. The structure of HHSC is shown in the diagram below:



※ HHSC1 and HHSC2 have exactly the same structure, for components above, the upper name indicates corresponding component of HHSC1, and the name in the brackets indicates HHSC2 component.

- The 2 hardware high-speed counters are HHSC1 and HHSC2.
- As shown in above diagram, HHSC has memory and hardware circuit registers at the same time. When use MOV and DMOV instructions to write data into HHSC related registers, CPU writes data into memory and hardware circuit registers at the same time. Please use MOV instruction for 16-Bit registers and use DMOV instruction for 32-Bit registers. Take special note that when change the HHSC related registers without MOV and DMOV instructions, only the memory registers value will be changed, and the value of hardware circuit register will not be affected.

- Related components of hardware high-speed counter (HHSC)

For components with symbol "■" or are missing from the list below, their coils can not be driven by instructions and no data can be written to registers.

Coil ID. No.	Instruction of Function
M9194	Controls whether HHSC1 has interrupt when current value=configured value (interrupt pointer I050). OFF means no interrupt, ON means has interrupt.
M9195	Controls whether HHSC2 has interrupt when current value=configured value (interrupt pointer I060). OFF means no interrupt, ON means has interrupt.
■ M9196	Displays current counting direction of HHSC1. OFF means count up, ON means count down.
■ M9197	Displays current counting direction of HHSC2. OFF means count up, ON means count down.

Register ID.	Instruction of Function	
D9224	HHSC1 working mode selection. 0 indicates do not start counting function of HHSC1, 1 ~ 18 indicates different working mode respectively. (Please refer to the table below)	
D9225	HHSC2 working mode selection. 0 indicates do not start counting function of HHSC2, 1~18 indicates different working mode respectively. (Please refer to the table below)	
D9226	Lower 16 bits	Current value registers for HHSC1.
D9227	Upper 16 bits	
D9228	Lower 16 bits	Current value registers for HHSC2.
D9229	Upper 16 bits	
D9230	Lower 16 bits	Configured value registers for HHSC1.
D9231	Upper 16 bits	
D9232	Lower 16 bits	Configured value registers for HHSC2.
D9233	Upper 16 bits	

- Table of HHSC Working Modes

Hardware High-Speed Counter No.	Input Point	HHSC Working Mode																	
		1 - Phrase Counting						2 - Phrase Counting			AB Phrase ∗ 1			AB Phrase ∗ 2			AB Phrase ∗ 4		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
HHSC1	X0	U	D	U	D	U	D	U	U	U	A	A	A	A	A	A	A	A	A
	X1							D	D	D	B	B	B	B	B	B	B	B	B
	X2			R	R	R	R		R	R		R	R		R	R		R	R
	X6					S	S			S			S			S			S
HHSC2	X3	U	D	U	D	U	D	U	U	U	A	A	A	A	A	A	A	A	A
	X4							D	D	D	B	B	B	B	B	B	B	B	B
	X5			R	R	R	R		R	R		R	R		R	R		R	R
	X7					S	S			S			S			S			S

U : Count up input　　D : Count down input　　A : A phrase input　　B : B phrase input

R : Reset Input　　S : Start - up input

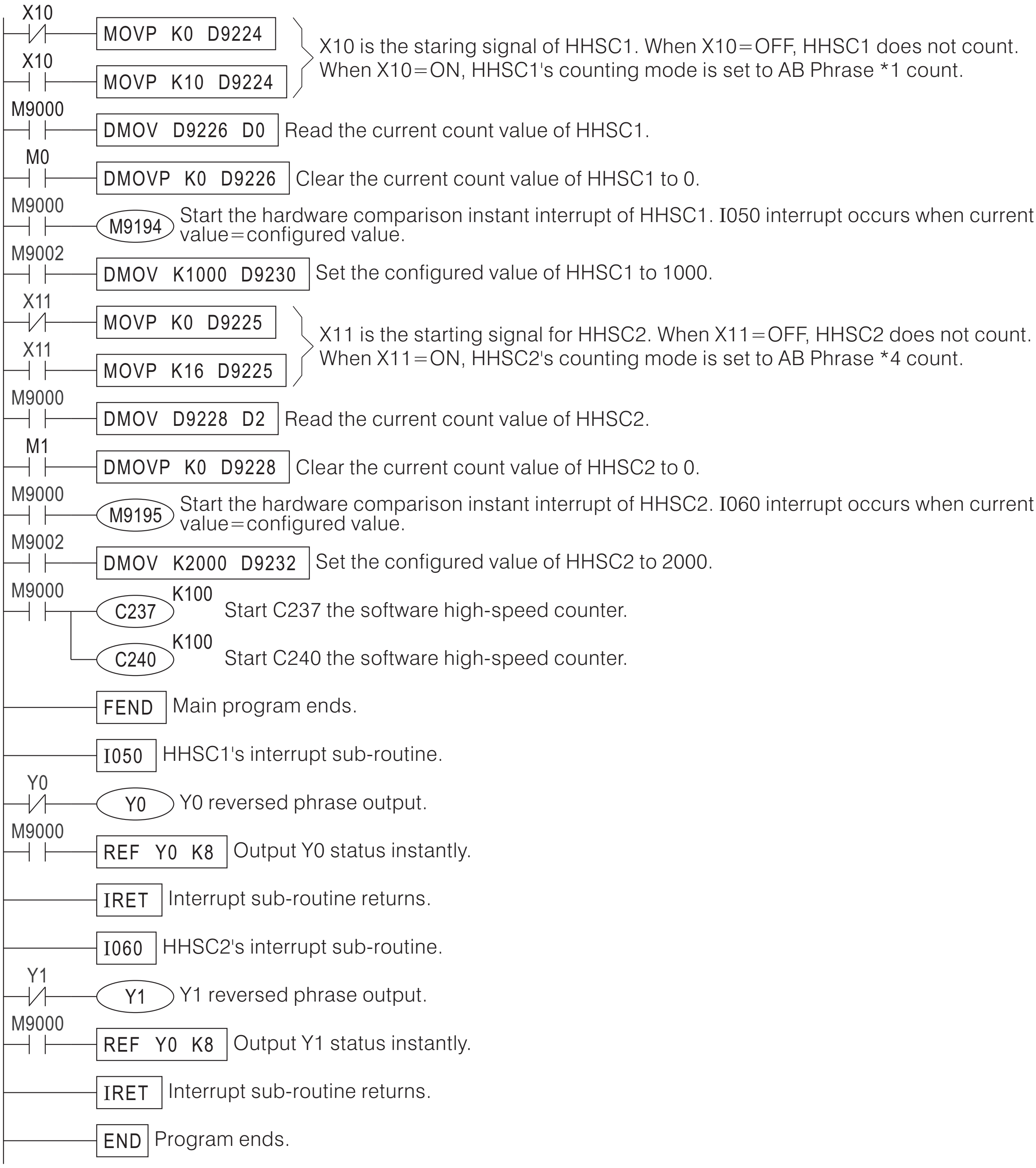
A-2-3 Hardware High-Speed Counter Programming Example

This programming example introduces a practical method of using HHSC1 and HHSC2.

When use HHSC, as long as the counting mode is configured first, the HHSC can do counting operation. The system program of PLC reads HHSC counting value through hardware circuit when the END instruction executes, and store it into the current value register. If instant counting value is needed, the DMOV instruction can be used to read current value register. When the DMOV instruction executes, the PLC system program will also read the hardware circuit counting value of HHSC. To clear the HHSC counting value, the DMOV instruction must be used instead of the RST instruction. In addition, the hardware comparison instant interrupt can be started when necessary, to avoid inaccuracy caused by scan time.

Besides the introduction on how to use HHSC, this programming example also started two software high-speed counters C237 and C240. This is to demonstrate that X0 ~ X7 are multi-usage high-speed input points, when they are not used by hardware high-speed counters, they can be used for other high-speed input functions, or as common input points.

Programming Example:



MEMO

B. Communication Functions Introduction

B-1 User Guide for Communication Functions

B-1-1 Communication Interface

The communication interfaces used by M, VB and VH series PLC are RS-232, S-422 and RS-485.

- RS-232 Interface Normally used for point to point short distance (within 15 meters) communication. The main units of M, VB and VH series PLC all have built-in RS-232 interface (CP1), which is used to connect to computer system for editing program.
- RS-422 Interface Normally used for point to point long distance communication.
- RS-485 Interface Normally used for multi-points long distance communication. Since it provides multi-points data exchange function and long distance communication function, so now is widely used in industrial control area.

B-1-2 Communication Parameters

When transfer data through communication interface, the data bit length, parity, stop bit and transfer speed need to be configured first, they are called communication parameters, and can also be treated as hardware level communication protocol. The communication parameter configurations must be consistent for all communication devices in the system.

B-1-3 Communication Protocols

All devices which can communicate have communication protocols. Communication protocol is software level protocol, and different devices exchange data through the same protocol. A communication protocol usually consists of starting character, station number, communication command, data content, end character and check code, etc. Of course, each of the devices defines its own communication protocol according to the need. Some follow the common protocols in the market, and the most commonly known one is MODBUS.

B-1-4 Communication Principles

When two or more than two devices try to exchange data, we need to connect them to form a communication circuit. And this communication circuit needs to follow the basic principles below to start working:

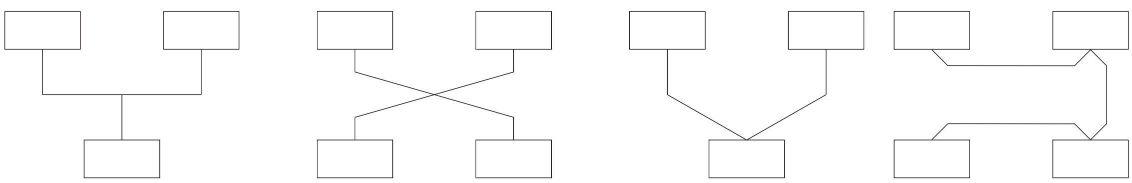
- Have consistent communication interface.
- Have consistent communication parameters.
- Have consistent communication protocols.
- The communication circuit must have a main leader role.

B-1-5 Safety Notes for Constructing Communication Systems

- Keep away from high noise source when wiring. Do not use the same groove as the power wire uses in the distribution box. Externally, keep as far away as possible from devices which have electric magnetic radiation.
- Pay attention to the communication distance and choose a suitable communication interface. Since the configurations of RS-485 interface is much better than RS-232, try to use RS-485 interface if possible for industrial control system. But there are also many guidelines need to take note when use RS-485 interface, please make sure they are strictly followed.
- Guidelines of using RS-485 interface

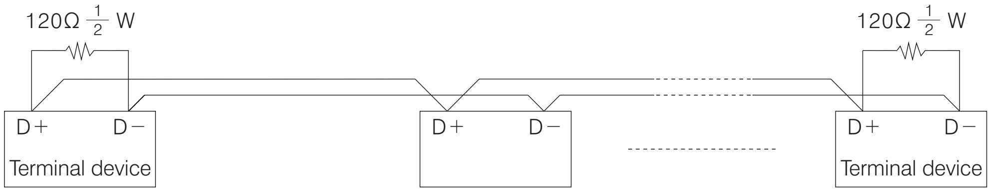
  ① The transfer wire need to use shielded twisted pair wire. Normal twisted pair wire can be used when conducting short distance communication in low noise environment to cut down cost. But in high noise environment, long distance communication or in occasions where high communication quality is required, the dedicated transfer wire for RS-485 (like Belden 9841) is recommended. It may make higher budget, but the communication quality will be improved magnificently.

  ② Make sure the principle of connect in sequence is followed when do hardware wiring, and do not use T type wiring method, star type wiring method or any other wiring method for convenience.



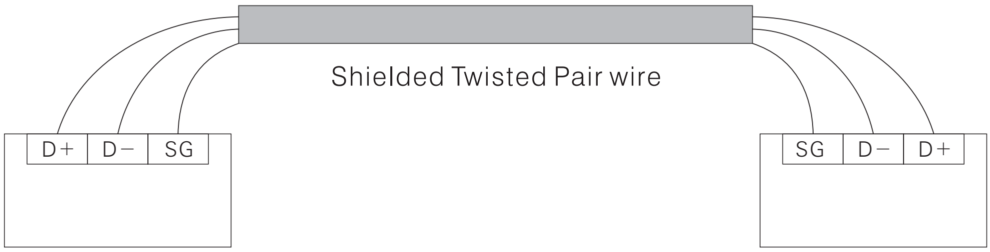
T Type Wiring　　Star Type Wiring　　Connect in Sequence Wiring

③ Terminal resistances must be parallel connected to the two terminal points of the whole communication circuit. For the twisted pair wire used by RS-485 interface, the terminal resistances should choose 120 Ω 1/2 W ones.



The communication wiring devices provided by VB and VH series PLC all have built-in terminal resistances, some of them can be enabled using sliding switch option, and some of them can be enabled using barrier terminal block style short connect option. For those communication devices which have no built-in terminal resistances, take special note during wiring to ensure that the external terminal resistances are well connected.

④ Although the RS-485 is a two-wires-style interface, when the distance between 2 communication devices is too long, communication often fails for the earth electric potential difference of the 2 devices is too big. Thus we normally recommend using the shield layer of the transfer wire to connect the SG terminals of the 2 devices, so that the earth electric potential difference can be reduced, and the communication can work well.



⑤ When the number of serial connections the RS-485 circuit has exceeds certain amount (depends on the specification of the devices connected, usually 32), an RS-485 amplifier has to be added to the circuit.

⑥ According to the standard specifications of RS-485 interface, the longest communication distance is 1200 meters. When the RS-485 communication circuit exceeds this distance, the RS-485 amplifier must be added to increase the communication distance.

- It is possible that one communication circuit connects with different devices at the same time, so when the communication fails, carefully check whether all wirings are correct and stable and whether the configuration values of each device are correct. Sometimes can even separate the devices to do individual checking to make sure it work well, before connecting it with many other devices and making it more difficult to find out the problem.
- Misconception about communication speed. The communication systems are built for various purposes and usages. People usually think that for speed, faster is the better, but this conception is actually not always true, because faster communication speed need to be supported by higher communication quality, and also means more expensive system construction budget. So the correct way is to choose a suitable communication speed according to the need, think a reasonable construction budget and target for stable communication quality.
- When the built communication system is able to function, but often has interruptions or errors, results in unsmooth and delayed transfer of the data, the following suggestions are given:

① Check whether the communication software is working properly, including whether the communication parameters (like the time-out time setting) are correct.

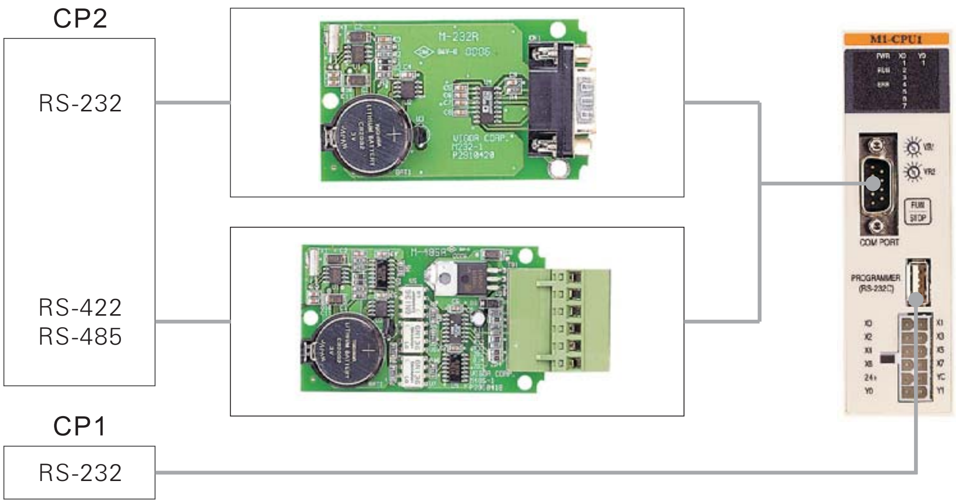
② Reduce environmental interferences. Detailed method includes lower the load frequency of frequency converters; make sure the earth connection system of the frequency converters and power suppliers are set up properly; or even add noise suppress devices to the power wire.

③ If normal transfer wire is used, the user is suggested to change it to RS-485 dedicated transfer wire.

④ Re-wire the transfer wire, and follow the keep away from noise source principle.

B-2 Communication System Structure

B-2-1 Communication System Structure of M Series PLC



◆ COM Port 1 (CP1)

The CP1 is a built-in RS-232 communication standard interface.
The applicable communication type of CP1 is the Computer Link, which is to execute the M, VB and VH Series communication protocol. Its main purposes are to:

1. Connect to the programming tools (Computer + Ladder Master or PDA + NeoTouch).
2. Connect to the HMI (Human-Machine Interface) or SCADA (Supervisor Control And Data Acquisition)
3. Connect with a MODEM, which is for remote program modification and data monitoring.

◆ COM Port 2 (CP2)

CP2 is a multi-functional expansion communication port and it can be used for various communication applications.

1. Computer Link – Uses the M, VB and VH Series communication protocol and it has the same purpose for use as CP1 in the RS-232 interface. By the RS-485 interface, a computer and several PLCs can constitute a monitoring local access network.

2. Easy Link – Uses the M, VB and VH Series communication protocol. Basically this application type is similar to the Computer Link, except this Easy Link uses a Main Unit of M or VB Series (which is called "Master PLC") to replace the computer, HMI or SCADA in the local network. For the data transfer in the network, programmer need to put the LINK instruction (FNC 89) in the Master PLC's program to access the data in Slave PLCs.

3. CPU Link – Uses the dedicated communication protocol and it is only available by the RS-485 interface. The CPU Link allows to transfer data between (2 ~ 8) PLCs, usually it is used for the distributed control system.

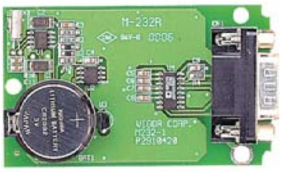
4. Parallel Link – Uses the dedicated communication protocol and it has the same purpose for use as the CPU Link, except its procedure is simpler and allows to transfer data between only 2 PLCs.

5. MODBUS – Uses the MODBUS (Slave) communication protocol (the MODBUS is a standard open source communication protocol). Usually all the SCADA (Supervisor Control And Data Acquisition) and HMI (Human-Machine Interfaces) have the MODBUS communication protocol.

6. MODEM Communication – Actively contacts with a MODEM when the PLC boots up (MODEM's "AA" sign should light on), then exercises M, VB and VH Series communication protocol. By the linked MODEMs, the PLC allows to perform remote program modification or data monitoring.

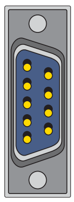
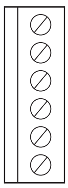
7. MODEM Dialing – Uses the function of MODEM Communication above (if the dialing function of VB Series PLC and MODEM are activated) then triggers the PLC's Dial-up Connection to link with the other PLC. The function is very useful, especially for remote abnormality report, security system and data collector.

8. Non-Protocol – It does not administer any specific communication protocol. All communication processes are customized and completed by PLC program. It uses RS instruction (FNC80) to receive and transfer communication operation. This communication type is usually used for links with other peripherals in the market, such as temperature controller, frequency converter, displayer, printer, card reader or bar code reader.

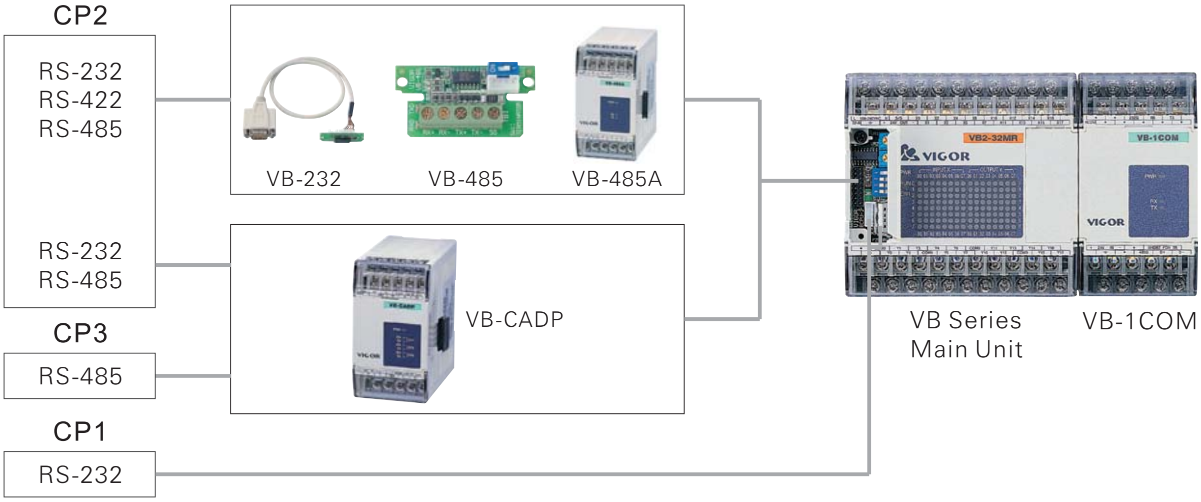


Communication Expansion Board

- M-232R and M-485R are expansion cards for M series PLC's second communication port (CP2).
- The CP2 of M series PLC is a multi-usage port which can execute many communication functions like Computer Link, CPU Link, Parallel Link, Easy Link, MODBUS, MODEM Communication, MODEM Dialing and Non Protocol Communication.

Item	M-232R	M-485R
Communication Interface	RS-232C	RS-422/RS-485
Isolation Method	Photo-coupler Isolation	
Distance	15 Meters	1000 Meters
Communication Method	Half-duplex	
Communication Speed	300/600/1200/2400/4800/9600/19200/38400 bps	
Communication Protocol	Computer Link, Easy Link, MODEM } Protocol of M, VB &VH Series PLC Parallel Link : Dedicated Protocol MODBUS : Protocol by other producer Non Protocol : User customized and complete using PLC program, then communicate with other equipment through RS instruction.	Computer Link, Easy Link } Protocol of M, VB &VH Series PLC CPU Link, Parallel Link } Dedicated Protocol MODBUS : Protocol by other producer Non Protocol : User customized and complete using PLC program, then communicate with other equipment through RS instruction.
Power Supply	DC5V 20mA (from PLC power supply)	DC5V 15mA (from PLC Power Supply) DC24V 60mA (from external Power Supply)
Wiring Method	 1 : CD 2 : RXD 3 : TXD 5 : SG 7 : RTS 8 : CTS 4,6,9 : Not Use D-Sub Connector 9Pin Male Connector	Screw - Cage Clamp Terminal Block 
Parameter Configuration	For CP2 relevant parameter configuration settings please use the “System --- 2nd. COM Port Setting… ” function of the programming software Ladder Master.	

B-2-2 Communication System Structure of VB Series PLC



◆ COM Port 1 (CP1)

The CP1 is a built-in RS-232 communication standard interface. It is available to connect with other equipment via either the USB type or the white JST 4P connector.
The applicable communication type of CP1 is the Computer Link, which is to execute the M, VB and VH Series communication protocol. Its main purposes are to:

1. Connect to the programming tools (Computer + Ladder Master or PDA + NeoTouch).
2. Connect to the HMI (Human-Machine Interface) or SCADA (Supervisor Control And Data Acquisition)
3. Connect with a MODEM, which is for remote program modification and data monitoring.

◆ COM Port 2 (CP2)

CP2 is a multi-functional expansion comm. port and can be used for many comm. Applications.

1. Computer Link – Uses M, VB and VH Series comm. protocol and has same usage as CP1 for RS-232 interface. For RS-485 interface, a pc and several PLCs can form a monitoring local access network.
2. Easy Link – Uses M, VB and VH Series comm. protocol. Basically it is similar to Computer Link, except that a M or VB Series Main Unit ("Master PLC") is used to replace the pc in the local network. For data exchange, LINK (FNC 89) need to be used in Master PLC program to access data in Slave PLCs.
3. CPU Link – Uses dedicated communication protocol and is only available for RS-485 interface.It allows to transfer data between (2 ~ 8) PLCs, usually it is used for distributed control system.
4. Parallel Link – Uses dedicated comm. protocol and has same usage as CPU Link, except its procedure is simpler and allows to transfer data between only 2 PLCs.
5. MODBUS – Uses MODBUS (Master/Slave) comm. protocol (standard open source comm. Protocol) Common SCADA and HMI have this MODBUS communication protocol. The market sold devices without VB comm. Protocol can connect to VB series PLC through this application type.
6. MODEM Communication – Actively contacts with MODEM when PLC boots up (MODEM AA sign is on), then runs M, VB and VH protocol through MODEMs to modify remote program or monitor data.
7. MODEM Dialing – Use MODEM functions above, if VB PLC connects MODEM then trigger PLC Dial-up to link with other PLCs, especially useful for remote abnormality report, security sys. And data collect.
8. Non-Protocol – Does not use specific comm. Protocol. Comm. processe is customized and done by PLC program. It uses RS instruction (FNC80) to receive/transfer data. It is usually used to link with temperature controller, frequency converter or bar code reader etc in market.

◆ COM Port 3 (CP3)

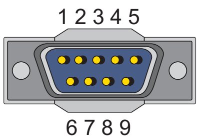
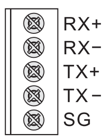
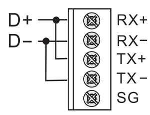
The CP3 is a RS-485 communication port which is expanded by the VB-CADP expansion module and the communication type is assigned as Computer Link (using the M,VB and VH Series communication protocol). It is usually linked with the HMI (Human-Machine Interface) or the SCADA (Supervisor Control And Data Acquisition) to make the monitoring of local networking.

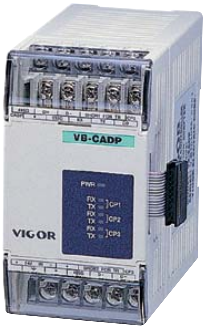
◆ VB-1COM

The VB Series PLC Serial Link Communication Module provides a RS-232/RS-485 communication port. It does not administer any specific communication protocol. All the communication processes are customized and completed by the PLC program. This module is usually used for to communicate with other peripherals, such as commercially available temperature controller, frequency converter or bar code reader. A Main Unit can expand up to 16 VB-1COM modules.

Communication Expansion Cards

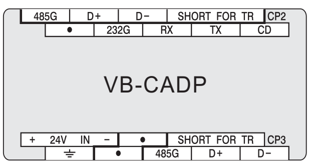
- The VB-232 and VB-485 are the Second COM Port (CP2) expansion cards of the VB Series PLC.
- The CP2 of the VB & VH Series PLC is a multi-functional communication port that can be used for multifarious communication types, e.g. Computer Link, CPU Link, Parallel Link, Easy Link, MODBUS Communication, MODEM Communication and Non-Protocol Communication.

Item	VB-232	VB-485
Communication Interface	RS-232C	RS-422/RS-485
Isolation Method	No Isolation	
LED Indicator	RXD、TXD	
Distance	15 M (48.21') Max.	50 M (164.04') Max.
Communication Method	Half-duplex	
Communication Speed	300/600/1200/2400/4800/9600/19200/38400 bps	
Communication Protocol	Computer Link, Easy Link, MODEM } M , VB and VH Series PLC communication protocol Parallel Link : Dedicated Protocol MODBUS : Protocol by other producer Non Protocol : User customized and complete using PLC program, then communicate with other equipment through RS instruction. ※The VB Series PLC supports all the communication protocols mentioned above. The VH series PLC only supports Computer Link, MODBUS and Non Protocol Communication.	Computer Link, Easy Link } M , VB and VH Series PLC communication protocol CPU Link, Parallel Link } Dedicated Protocol MODBUS : Protocol by other producer Non Protocol : User customized and complete using PLC program, then communicate with other equipment through RS instruction. ※The VB Series PLC supports all the communication protocols mentioned above. The VH series PLC only supports Computer Link, MODBUS and Non Protocol Communication.
Power Supply	DC 5V, 10mA (from PLC Main Unit)	DC 5V, 60mA (from PLC Main Unit)
Wiring Method	 D-Sub Connector 9Pin Male Connector 1 : CD 2 : RXD 3 : TXD 5 : SG 7 : RTS 8 : CTS 4,6,9 : Not Use	Screw - Cage Clamp Terminal Block  Note: 1.RS-485 Wiring  2. SW1 is the terminal resistance switch (terminal resistance 120 Ω.)
Parameter Configuration	For CP2 relevant parameter configuration settings please use the “System --- 2nd. COM Port Setting... ” function of the programming software Ladder Master.	



VB-CADP Dual-Port Communication Expansion Module

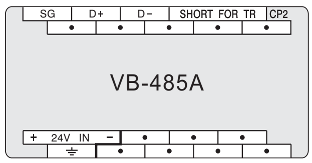
- It is a CP2 and CP3 expansion module for VB and VH series.
- The CP2 provides an isolated RS-232 or RS-485 communication interface. The communication distance of its RS-485 interface is up to 1000 M (3280').
- The CP3 provides isolated RS-485 communication interface with the communication distance of this RS-485 interface is up to 1000 M (3280').
- The CP2 of the VB Series PLC is a multi-functional communication port which can be assigned for various communication applications, e.g. Computer Link, CPU Link, Parallel Link, Easy Link, MODBUS Communication, MODEM Communication and Non-Protocol Communication.

Item	CP2		CP3
Communication Interface	RS-232	RS-485	RS-485
Isolation Method	Photocoupler Isolation		
LED Indicator	RX、TX (CP2)		RX、TX (CP3)
Distance	15 Meters	1000 Meters	1000 Meters
Communication Method	Half-duplex		
Communication Speed	300/600/1200/2400/4800/9600/19200/38400 bps		19200 bps
Communication Protocol	Computer Link Easy Link MODEM(RS-232) } M , VB and VH Series PLC communication protocol CPU Link (RS-485) Parallel Link } Dedicated Protocol MODBUS : Protocol by other producer Non Protocol : User customized and complete using PLC program, then communicate with other equipment through RS instruction. ※The VB Series PLC supports all the communication protocols mentioned above. The VH series PLC only supports Computer Link, MODBUS and Non Protocol Communication.		Computer Link : M , VB and VH Series PLC communication protocol Baud Rate : 19200 bps Data Length : 7 bits (ASCII) Parity : EVEN Stop bit : 1 bit
Power Supply	DC 24V ±10%, 70mA (External power required)		
Wiring Method	Barrier style terminal block connection		
Parameter Configuration	For selection of CP2 application types and relevant parameter configuration settings, please use the developmental software Ladder Master, then open the option: "System --- 2nd. COM Port Setting.... " .		Communication station number setting is by the rotary switch on the left side of the module. (00～99)

- When a Main Unit connects with a VB-CADP Module, the CP1 in the Main Unit will be disabled and its function will be replaced by the CP1 in the VB-CADP. The communication station number of the CP1 must assign to 0.
- The VB-CADP Module also provides the Power LED and RX, TX transmission indicators for the CP1.

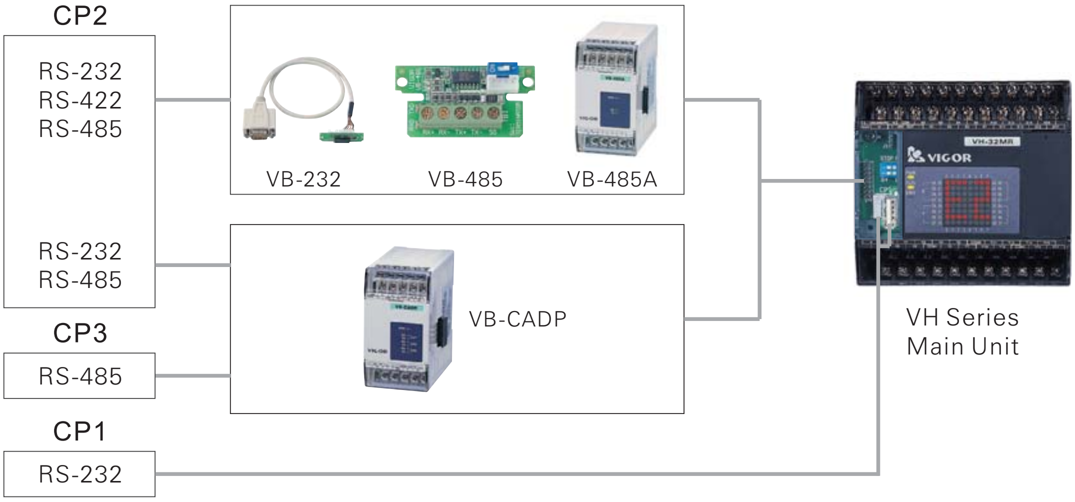
VB-485A RS-485 Communication Expansion Module

- The Second COM Port (CP2) expansion module for a Main Unit.
- It is an isolated RS-485 communication interface, the distance is up to 1000 M (3280').
- The CP2 of the VB and VH Series PLC is a multi-functional communication port that can be assigned for various communication applications, e.g. Computer Link, CPU Link, Parallel Link, Easy Link, MODBUS Communication, MODEM Communication and Non-Protocol Communication.

Item	Specification
Communication Interface	RS-485
Isolation Method	Photocoupler Isolation
LED Indicator	PWR、RX、TX
Distance	1000 Meters
Communication Method	Half-duplex
Communication Speed	300/600/1200/2400/4800/9600/19200/38400 bps
Communication Protocol	Computer Link, Easy Link } M , VB and VH Series PLC communication protocol CPU Link, Parallel Link } Dedicated communication protocol MODBUS : Protocol by other producer Non Protocol : User customized and complete using PLC program, then communicate with other equipment through RS instruction. ※The VB Series PLC supports all the communication protocols mentioned above. The VH series PLC only supports Computer Link, MODBUS and Non Protocol Communication.
Power Supply	DC 24V ±10% , 55mA (External power required)
Wiring Method	Barrier style terminal block connection 
Parameter Configuration	For CP2 relevant parameter configuration settings please use the “System --- 2nd. COM Port Setting... ” function of the programming software Ladder Master.

◆ About the specifications and introduction of VB-1 COM communication module, please refer to “B-4 VB-1 COM Serial Link Communication Module”

B-2-3 Communication System Structure of VH Series PLC



◆ COM Port 1 (CP1)

The CP1 is a built-in RS-232 communication standard interface. It is available to connect with other equipment via either the USB type or the white JST 4P connector.
The applicable communication type of CP1 is the Computer Link, which is to execute the M, VB and VH Series communication protocol. Its main purposes are to:

1. Connect to the programming tools (Computer + Ladder Master or PDA + NeoTouch).
2. Connect to the HMI (Human-Machine Interface) or SCADA (Supervisor Control And Data Acquisition)
3. Connect with a MODEM, which is for remote program modification and data monitoring.

◆ COM Port 2 (CP2)

CP2 is a multi-functional expansion comm. port and can be used for many comm. Applications.

1. Computer Link – Uses M, VB and VH Series comm. protocol and has same usage as CP1 for RS-232 interface. For RS-485 interface, a pc and several PLCs can form a monitoring local access network.
2. MODBUS – Uses MODBUS (Master/Slave) comm. protocol (standard open source comm. Protocol) Common SCADA and HMI have this MODBUS communication protocol. The market sold devices without VH comm. Protocol can connect to VB series PLC through this application type.
3. Non-Protocol – Does not use specific comm. Protocol. Comm. processe is customized and done by PLC program. It uses RS instruction (FNC80) to receive/transfer data. It is usually used to link with temperature controller, frequency converter or bar code reader etc in market.

◆ COM Port 3 (CP3)

The CP3 is a RS-485 communication port which is expanded by the VB-CADP expansion module and the communication type is assigned as Computer Link (using the M,VB and VH Series communication protocol). It is usually linked with the HMI (Human-Machine Interface) or the SCADA (Supervisor Control And Data Acquisition) to make the monitoring of local networking.

◆ For introductions on the communication expansion boards (VB-232, VB-485) and communication expansion modules (VB-485A, VB-CADP) please refer to “B-2-2 Communication System Structure of VB Series PLC”

MEMO

B-3 Communication Operation Mode

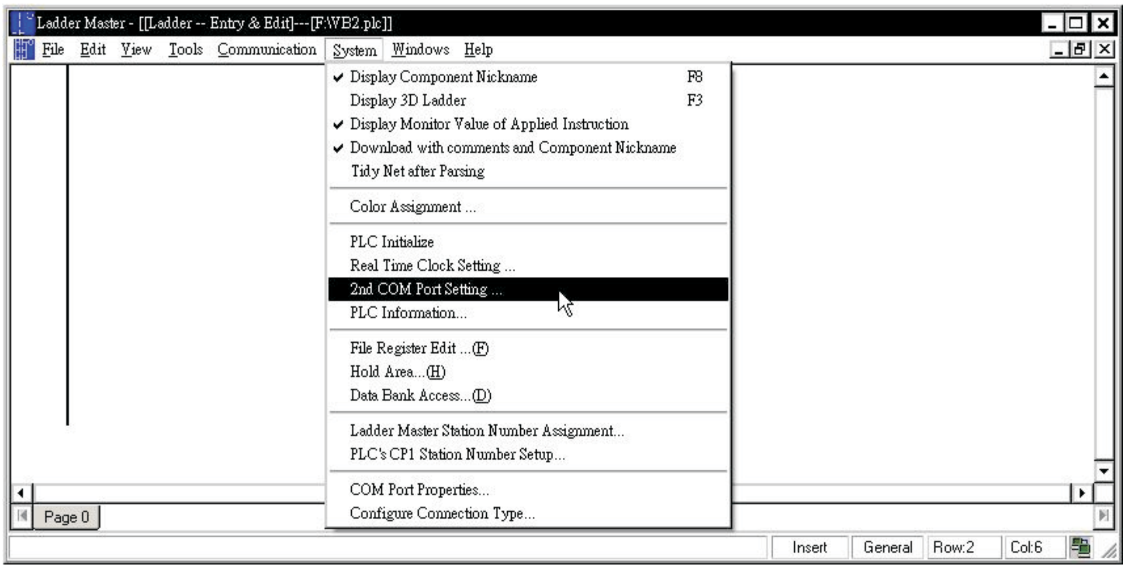
The M, VB and VH series PLCs have complete communication functions and multiple communication working modes.

CP1 and CP3 support M, VB and VH communication protocols. CP2 is a multi-functional communication port which supports many other communication applications besides the M, VB and VH communication protocol, e.g. Easy Link, CPU Link, Parallel Link, MODBUS Communication, MODEM Communication, MODEM Dialing and Non-Protocol Communication. The introductions of these working modes are listed below.

B-3-1 Choosing an Operation Mode for CP2 Communication

Since CP2 supports many operation modes, the user needs to select and set an operation mode before using it.

The operation mode of CP2 is configured by the programming tool Ladder Master, the steps are as below:



Click "System" in the menu toolbar.

↓

Select "2nd COM Port Setting" from the drop-down menu.

↓

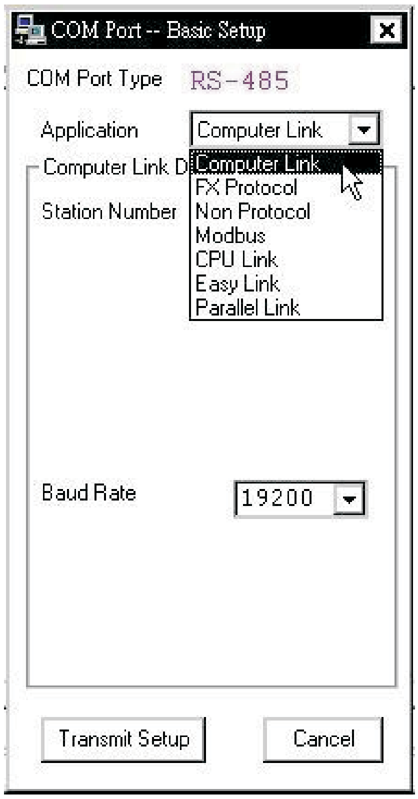
The screen will show configuration window as on the right, Select an application type as you wish.

↓

Every application type has its corresponding interface to guide the user in the parameter setting process.

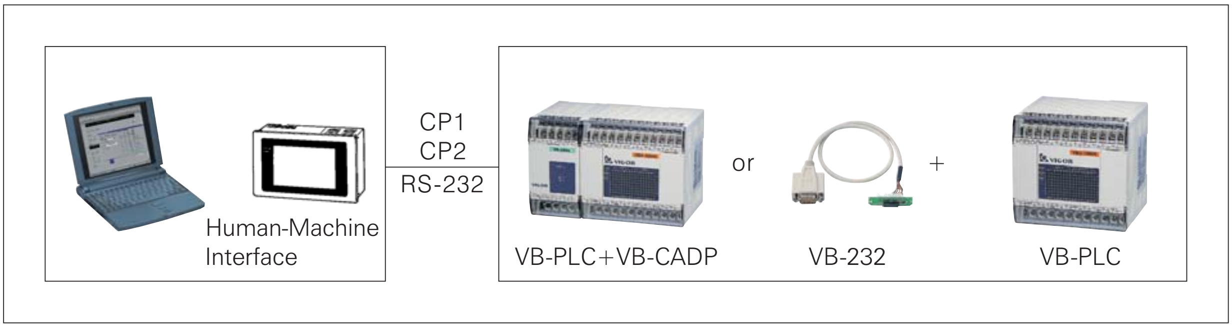
↓

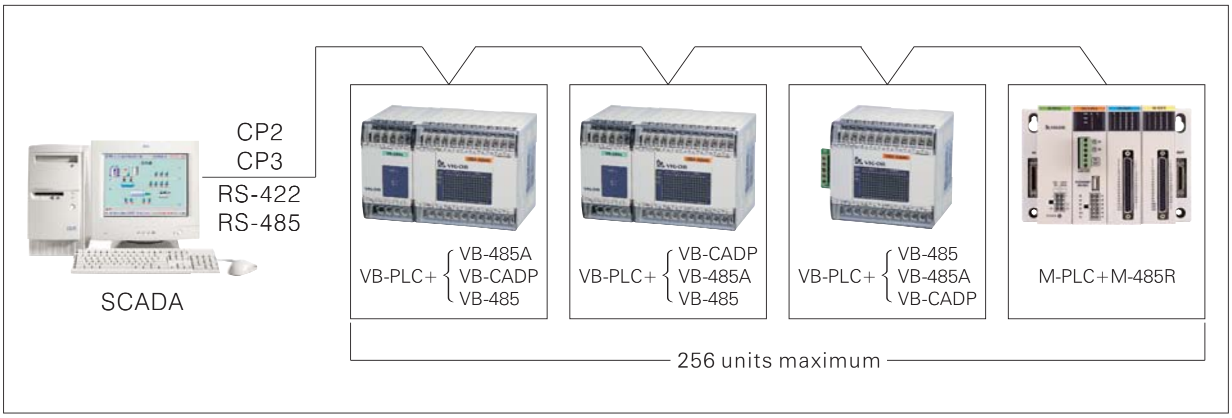
Click "Transmit Setup" after configuration done, to transfer the settings to PLC main unit.



B-3-2 Computer Link

◆ A computer, HMI (Human-Machine Interface) or SCADA (Supervisor Control and Data Acquisition) can connect to PLCs via the Computer Link. For RS-232 interface, its usage is the same as CP1. For RS-485 interface, normally a computer and many PLCs are used to form a local monitor network.



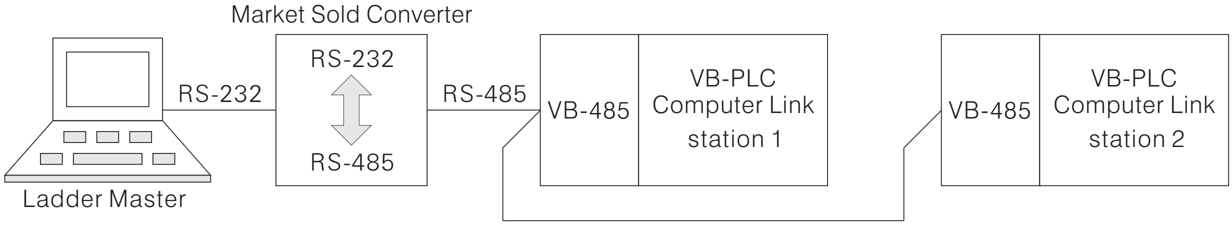


Item	Specification	
Transmission Interface	RS-232	RS-422/RS-485
Communication Protocol	M, VB and VH Series Communication Protocol	
Communication Method	Half-duplex	
Communication Parameter	Data Length: 7 bits (ASCII); Parity: EVEN; Stop Bit: 1 bit	
Baud Rate	CP1 and CP3: 19200 bps; CP2: 4800/9600/19200/38400 bps	
Distance	15 M (49')	1000 M (3280'); (50 M /164', if the network has a VB-485)
Number of Linked Stations	1 station	256 stations maximum (when more than 32 stations, a powered booster is required)
Connection Equipment	CP1: Main Unit Built-in CP2: VB-232, VB-CADP or M-232R	CP2 ： VB-485 、VB-485A 、VB-CADP or M-485R CP3 ： VB-CADP
Linkable PLC	VB Series, VH Series and M Series PLC	
Data Transfer Category	Including all of X, Y, M, S, T, C and D	

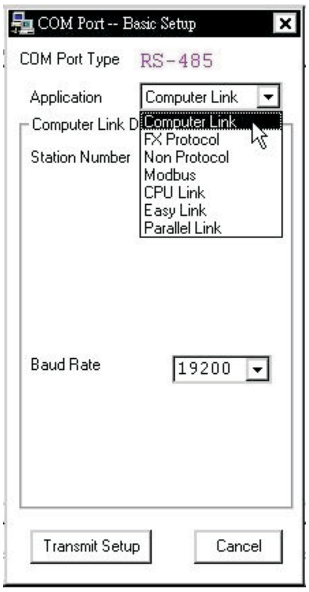
- For any device tries to communicate with M, VB and VH series PLCs, like computer, HMI, etc, as long as it follows the communication protocol of M, VB and VH series PLC to send proper command, PLC will respond to the communicating request. About the communication protocol of M, VB and VH series PLCs, please refer to "B-5 Communication Protocol of M, VB and VH Series".
- The SCADA or HMI producers usually write corresponding driver programs according to the communication protocols provided by the PLC producers. So that the SCADA and HMI users only need to choose the proper driver program at the planning stage to connect the SCADA, HMI and PLCs together to construct a monitor network.
- Since the M, VB and VH series of PLCs use the same comm. protocol, the SCADA or HMI can choose any driver program of VIGOR M, VB or VH series. Anyway, some imported SCADA or HMI do not have M, VB or VH series driver program, thus they need to connect by "Other Producer's comm. protocol (MODBUS)". For detailed introduction, please refer to "B-3-6 MODBUS Communication".
- When the CP2 is assigned for the Computer Link or MODBUS communication, its station number is shown in special register D9121.

◆ Application Example

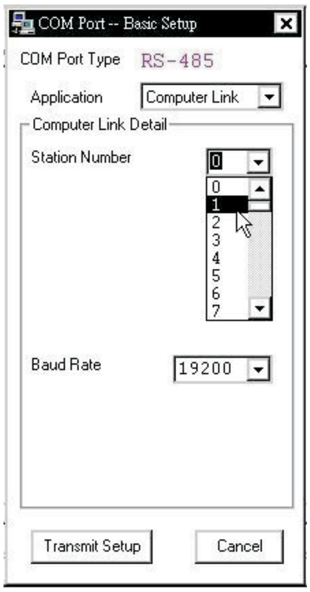
This example connects to Station 1 and Station 2 PLCs from the computer communication port (normally RS-232) through a market sold RS-232 to RS-485 converter. Then run the Ladder Master in PC to connect to station 1 and station 2 for program downloading/uploading and monitor work.



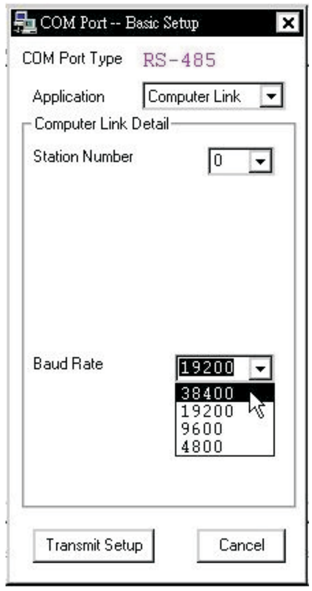
- First, set the CP2 parameter for each PLC by Ladder Master though CP1



Select the application to be Computer Link

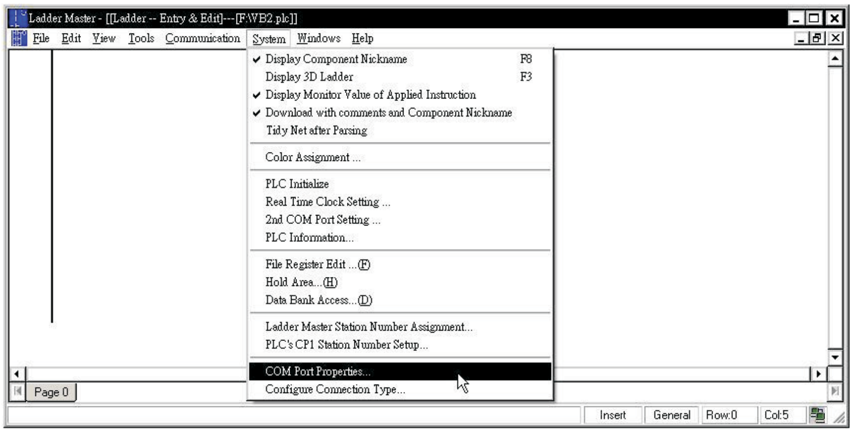


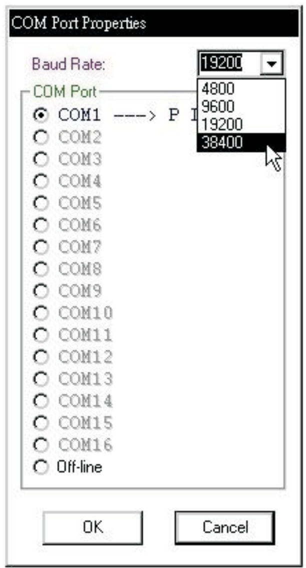
Set PLC station number to be station 1 and station 2



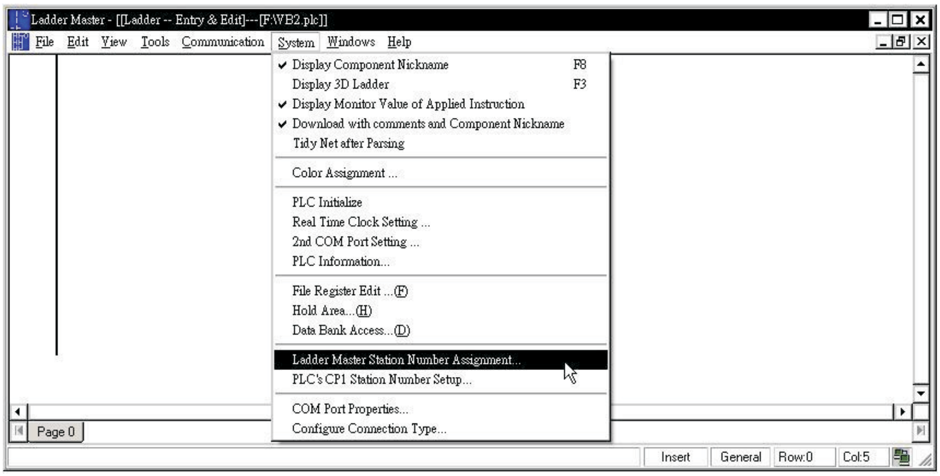
Set the baud rate, all PLCs and Ladder Master should have the same rate

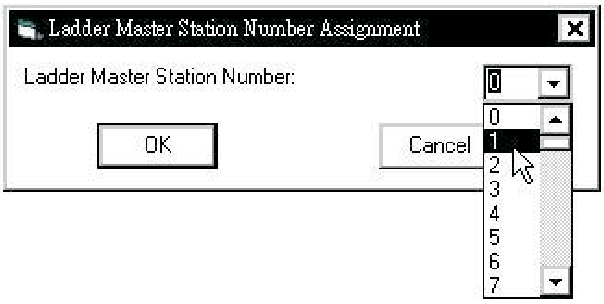
- Set the communication rate in Ladder Master.





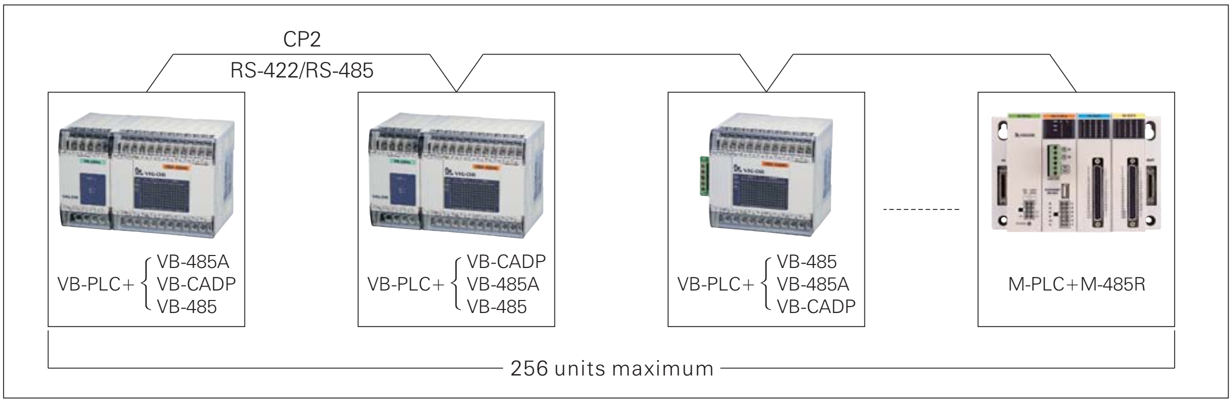
- Set the communication station number in Ladder Master to connect and communicate with this PLC station.





B-3-3 Easy Link

◆ This mode uses the M, VB and VH Series communication protocol as same as the Computer Link does, except that it uses a Main Unit of M or VB Series (which is called “Master PLC”) to replace the computer in the local network. For the data transfer in the network, the programmer needs to put the LINK instruction (FNC 89) in the Master PLC's program to access the data in Slave PLCs. This mode is mainly used for many PLCs to exchange a lot of data with each other.



Item	Specification
Transmission Interface	RS-422/RS-485
Communication Protocol	M, VB and VH Series Communication Protocol
Communication Method	Half-duplex
Communication Parameter	Data Length: 7 bits (ASCII); Parity: EVEN; Stop Bit: 1 bit
Baud Rate	4800/9600/19200/38400 bps
Distance	1000 M (3280'); (50 M /164', if the network has a VB-485)
Number of Linked Stations	256 stations maximum (when more than 32 stations, a powered booster is required)
Connection Equipment	VB or VH Series: VB-485, VB-485A or VB-CADP; M Series: M-485R
Linkable PLC	VB Series and M Series PLC (VH Series can be used as Slave)
Data Transfer Category	Including all of X, Y, M, S, T, C and D

◆ The next page introduces how to use LINK instruction.

	FNC 89 LINK		LINK (S1) (S2)	Easy Link Communication	M	VB	VH
					○	○	

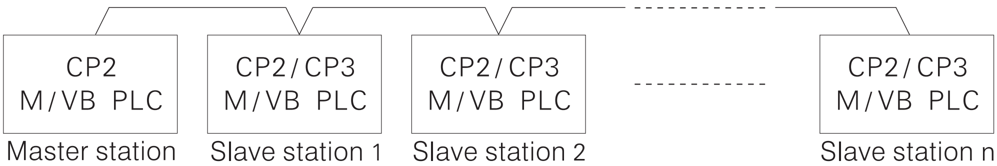
Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					

```
       X20              (S1)    (S2)
  |----| |------[ LINK  D1000   D100 ]
```

S1 : Head ID number of the register, which describe the data transfer/receive actions

S2 : Instruction working area, occupies 4 consecutive registers

- If the M Series CPU module mounts a M-232R or M-485 communication card, the CPU module will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between PLCs.
- If the VB Series Main Unit mounts a communication card (VB-232 or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd. Communication Port). Then, via this instruction to proceed data transfer between PLCs.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 is assigned to this instruction, the communication type should use "EASY LINK" or "COMPUTER LINK". To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master "System---2nd COM Port Setting..." to get the right setting.
- At most 256 nodes of M/VB Series PLC (slave VH series). can be linked together via this instruction and the RS-485 interface. The instruction can use for transfer the data of device X, Y, M, S, T, C and D.
- As the diagram below, select one of these linked PLCs as the Master station and the rest as Slave stations. Use the program develop devices (e.g. Ladder Master) to set the "EASY LINK" or "COMPUTER LINK" as the communication mode between the Master and Slave stations, and set each Slave station properly (the range of station ID number is 1 ~ 255). And then, write the data transmission/receiving command (designated by this instruction) to the Master station, to achieve the data transmission between PLCs.



- When X20= "ON", the LINK instruction will start to be performed. Based on the designated register string (which initiating from D1000), to do the data write or read action to the appointed Slave PLC station. And also, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by (S1) is duly completed, the M9199 will be "ON" for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20="ON" → "OFF", the instruction will be stopped and the data transmission/receiving will be disabled immediately.
- The LINK instruction can be used once only in the program.

- The register headed with S1 is used to describe the data transmission/receiving information:

S1	Content Value	Description
D1000	1 ~ 255	To designate the number of transferred and received data sets. Each data transmission/receiving set should be described with 7 registers.
D1001	1 ~ 255	Designates the Slave station ID number, to proceed data transmission/receiving for the particular Slave station
D1002	1 ~ 2	Instruction code. 1: read data from Slave stations; 2: write data in Slave stations
D1003	1 ~ 64	Length of data transferred or received. (If the data designated is a 32-bit counter, the content value = 1 ~ 32)
D1004	1 ~ 6 10 ~ 13	Designates the device type of the Master station 1:Input Contact X 2:Output Contact Y 3:Auxiliary Coil M 4:State Coil S 5:Timer Contact T 6:Counter Contact C 10:The Present-value Register of the Timer 11:16-bit Counter, Present-value Register 12:32-bit Counter, Present-value Register 13:Data Register D
D1005		Designates the initial ID number of the Master station device
D1006	1 ~ 6 10 ~ 13	Designates the device type of the Slave station
D1007		Designates the initial ID number of the Slave station device
D1008	1 ~ 255	Designates the Slave station ID number
D1009	1 ~ 2	Instruction code
D1010	1 ~ 64	Length of data transferred/received
D1011	1 ~ 6 10 ~ 13	Designates the device type of the Master station
D1012		Designates the initial ID number of the Master station device
D1013	1 ~ 6 10 ~ 13	Designates the device type of the Slave station
D1014		Designates the initial ID number of the Slave station device
⋮	⋮	

D1001 ~ D1007: Description of the 1st data transmission/receiving operation

D1008 ~ D1014: Description of the 2nd data transmission/receiving operation

- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.

- The instruction working area headed with S2:

S2	Description	
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
	Upper 8 bits	Instruction working status 0:Normal data transmission/receiving 2:Error of the length of the transferred/received data (unequal to 1 ~ 64) 4:Error of the designated device type 5:Error of the designated device ID number 6:The attributes of the designated devices by the Master and Slave stations are different A:Normal communications but no response from Slave stations B:Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

- Programming Example

```
M9002
─┤ ├──┬──[MOV  K2  D1000]
      ⋮
      └──[MOV  K100  D1014]
M9000
─┤ ├─────[LINK  D1000  D100]
```

Build data receive/send communication table.
User can also use "Ladder Master - Edit Communication Table" function to replace this block of program.

LINK D1000 D100: EASY LINK communication instruction

There are totally 2 transmission/receiving data sets in this example.

① Read D10 ~ D19 of Slave station #5 to D0 ~ D9 of the Master station

② Write M0 ~ M29 of the Master station to M100 ~ M129 of Slave station #2.

S1	Content Value	
D1000	2	Two transmission/receiving data sets
D1001	5	Designates Slave station #5
D1002	1	Reads data from the Slave station
D1003	10	Length of the data to be read
D1004	13	Designates the device headed with the Master station as D0
D1005	0	
D1006	13	Designates the device headed with the Slave station as D10
D1007	10	
D1008	2	Designates Slave station #2
D1009	2	Write data to the Slave station
D1010	30	Length of the data to be written
D1011	3	Designates the device headed with the Master station as M0
D1012	0	
D1013	3	Designates the device headed with the Slave station as M100
D1014	100	

D1001 ~ D1007: The 1st transmission/receiving data set:
D10 ~ D19 of Slave station #5
↓
D0 ~ D9 of the Master

D1008 ~ D1014: The 2nd transmission/receiving data set:
M0 ~ M29 of the Master
↓
M100 ~ M129 of Slave station #2

- Edit Communication Table

Besides using program to build data receiving/sending communication table, Ladder Master provides a more user-friendly data input interface to let the users build communication table.
Select the Ladder Master "Tools ---- Edit Communication Table ...." menu to enter the communication table edition screen. Through a step-by-step guiding window, the user can easily create and edit communication table.
After the edition is done, the communication data will be stored into file register assigned by the user, and the table is created. This function also allows the user to retrieve the table data back from file register for editing.
For VB series PLCs, the file register is read-only, and its value will be treated as part of the user program. When user copy or save program file, the file register together with the program itself will be copied or saved. This feature makes the file register very suitable for communication table storing. It can be easily copied from and helps to save PLC program space. For detailed introduction on file register, please refer to "2-9 File Register (D)".

- Communication Table Example

```
M9000
─┤ ├─────[LINK  D1000  D100]
```

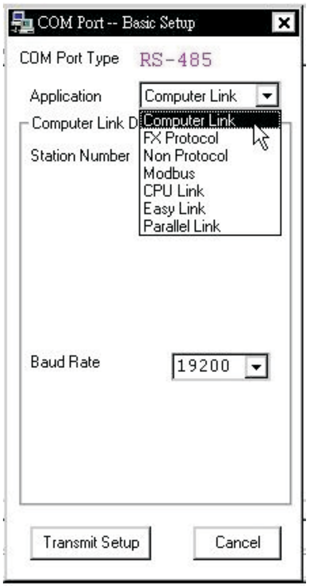
Instruction: LINK ▼ Table Starting Position: D1000 Table Length: 15

Number	Command	Master Data		Slave ID	Slave Data	Length	Word / Bit
1	Read	D0	<--	5	D10	10	W
2	Write	M0	-->	2	M100	30	B

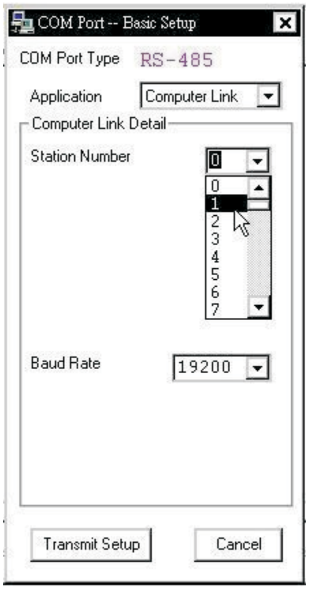
◆ Application Example

This example connects 2 VB series PLCs through RS-485 interface and executes Computer Link communication (M, VB and VH communication protocol). These 2 VB series PLCs have station number of 0 (Master) and 1 (Slave) respectively.

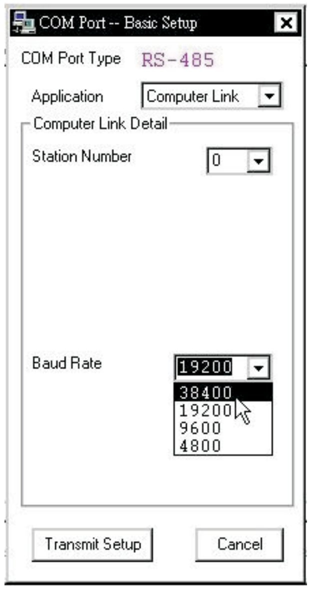
- Set the CP2 parameter for each PLC by Ladder Master though CP1



Select the application to be Computer Link



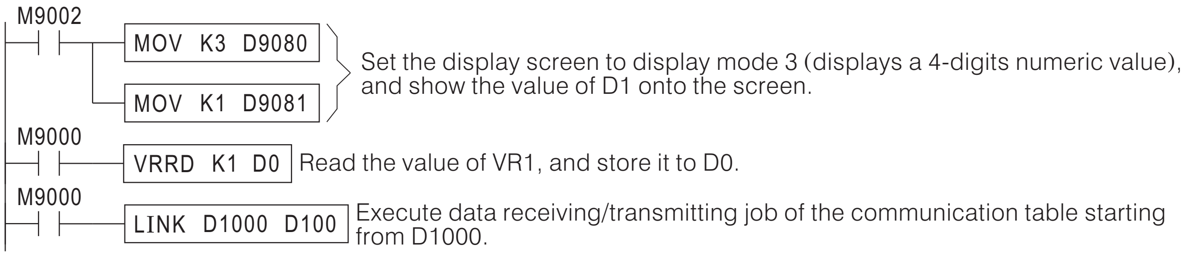
Set PLC station number to be station 0 and station 1



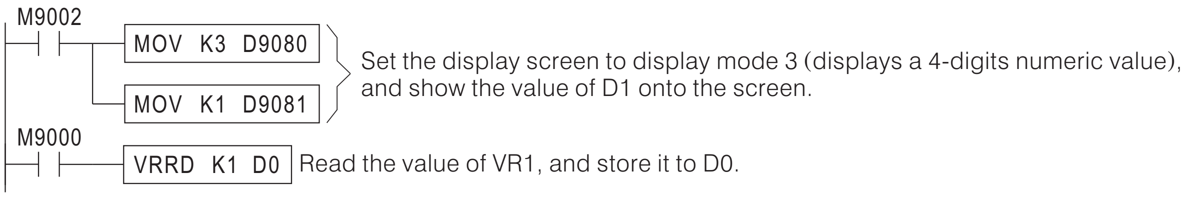
Set the baud rate, each PLC and Ladder Master should have the same rate

- When this example executes, the 2 PLCs exchange data with each other, the VR1 value of master PLC will be shown on the screen of slave PLC, and the VR1 value of slave PLC will be shown on the screen of the master PLC.
  At first, the master PLC reads the value of VR1, and then stores this value in D0 register. Then it writes the value of register D0 through communication interface into the D1 register of slave PLC. The slave PLC reads the VR1 value at the same time, and put the value into register D0. Then the master PLC reads the D0 register of the slave PLC through the communication interface, and then put this value into the D1 register of master PLC, at last, the master PLC shows the value of D1 register onto the screen.

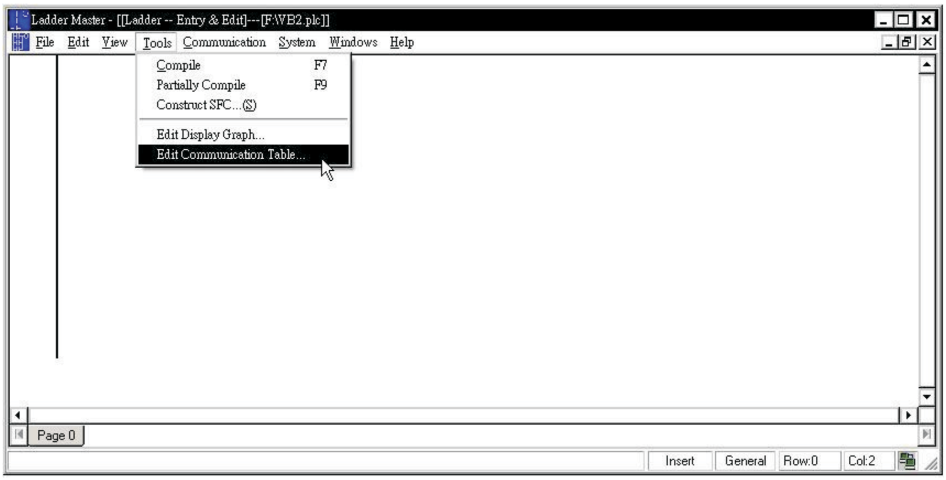
- Program of the Master PLC

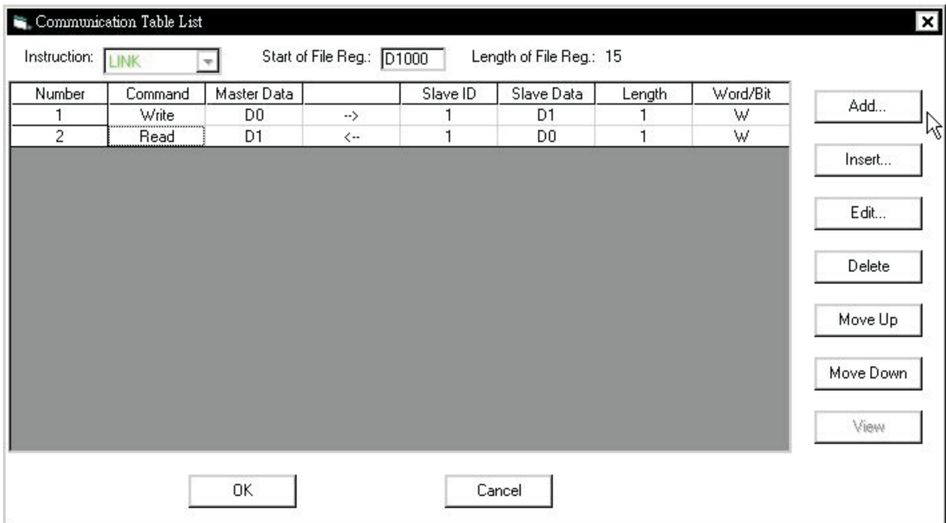


- Program of the Slave PLC



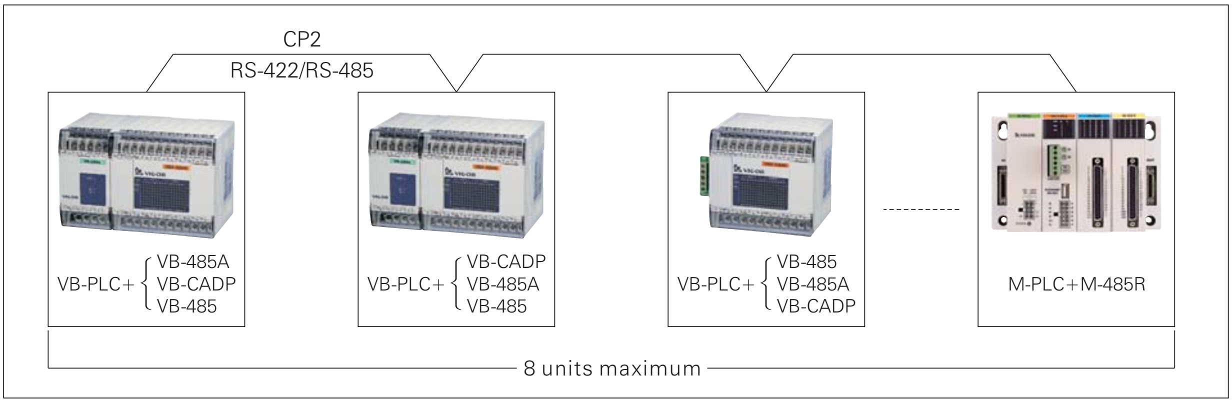
- Edit Communication Table





B-3-4 CPU Link

◆ CPU Link let 2 ~ 8 PLCs exchange data with each other, and is often used in distributed control system. In CPU Link network, PLC use dedicated communication protocol, and the PLCs in the network transfer data automatically based on configuration settings.



Item	Specification
Transmission Interface	RS-422/RS-485
Communication Protocol	Dedicated Communication Protocol
Communication Method	Half-duplex
Baud Rate	38400 bps
Distance	1000 M (3280'); (50 M /164', if the network has a VB-485)
Number of Linked Stations	2 ~ 8 stations
Connection Equipment	VB Series: VB-485, VB-485A or VB-CADP; M Series: M-485R
Linkable PLC	VB Series and M Series PLC

Transferable Data Range	Station No.	0 (Master)	1 (Slave)	2 (Slave)	3 (Slave)	4 (Slave)	5 (Slave)	6 (Slave)	7 (Slave)
	Mode 1	D0~3	D10~13	D20~23	D30~33	D40~43	D50~53	D60~63	D70~73
	Mode 2	D0~3 M1000~1031	D10~13 M1064~1095	D20~23 M1128~1159	D30~33 M1192~1223	D40~43 M1256~1287	D50~53 M1320~1351	D60~63 M1384~1415	D70~73 M1448~1479
	Mode 3	D0~7 M1000~1063	D10~17 M1064~1127	D20~27 M1128~1191	D30~37 M1192~1255	D40~47 M1256~1319	D50~57 M1320~1383	D60~67 M1384~1447	D70~77 M1448~1511

Communication Time	Linked Stations No	2	3	4	5	6	7	8
	Mode 1	7mS	11mS	15mS	19mS	23mS	27mS	31mS
	Mode 2	10mS	15mS	20mS	25mS	30mS	35mS	40mS
	Mode 3	16mS	24mS	33mS	42mS	50mS	59mS	68mS

- Nearly all the communication work modes of M, VB and VH series PLCs execute communication work after PLC completes the user program execution. Thus, the communication speed of the communication circuit is affected by not only the communication rate, but also the scan time of all the PLCs in the circuit. As a result, it is not easy to calculate the communication time of the circuit.
- CPU Link deals with communication work in instant interrupt way. So its communication speed is the fastest one, and can calculate the communication time of the circuit easily (see above table). As a result, it is suitable for distributed control system which requires instant reaction.

◆ CPU Link Related Components

For components with symbol "■" or are missing from the list below, their relay coils cannot be driven by instructions and no data can be written to them.

Coil ID. No.	Instruction of Function	M	VB	VH
■ M9183	CPU Link Comm. Failed (Master)	○	○	
■ M9184	CPU Link Comm. Failed (Slave 1)	○	○	
■ M9185	CPU Link Comm. Failed (Slave 2)	○	○	
■ M9186	CPU Link Comm. Failed (Slave 3)	○	○	
■ M9187	CPU Link Comm. Failed (Slave 4)	○	○	
■ M9188	CPU Link Comm. Failed (Slave 5)	○	○	
■ M9189	CPU Link Comm. Failed (Slave 6)	○	○	
■ M9190	CPU Link Comm. Failed (Slave 7)	○	○	

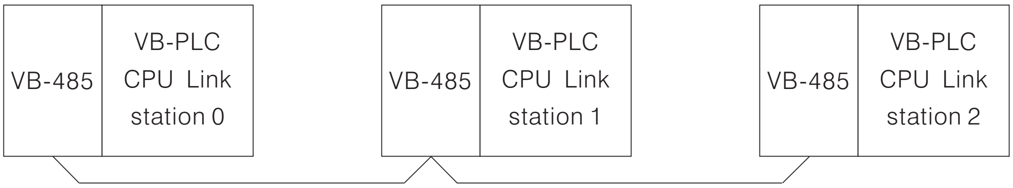
Register ID.	Instruction of Function	M	VB	VH
■ D9172	Comm. Time Out time		○	
■ D9177	Number of network slave stations	○	○	
■ D9178	Range of send component	○	○	
■ D9179	Time of comm. Retry	○	○	
■ D9201	Current network scan time	○	○	
■ D9202	Max. network scan time	○	○	
■ D9203	Time of comm. errors happen to master	○	○	
■ D9204	Time of comm. errors happen to slave 1	○	○	
■ D9205	Time of comm. errors happen to slave 2	○	○	
■ D9206	Time of comm. errors happen to slave 3	○	○	
■ D9207	Time of comm. errors happen to slave 4	○	○	
■ D9208	Time of comm. errors happen to slave 5	○	○	
■ D9209	Time of comm. errors happen to slave 6	○	○	
■ D9210	Time of comm. errors happen to slave 7	○	○	
■ D9212	Comm. error code of slave 1	○	○	
■ D9213	Comm. error code of slave 2	○	○	
■ D9214	Comm. error code of slave 3	○	○	
■ D9215	Comm. error code of slave 4	○	○	
■ D9216	Comm. error code of slave 5	○	○	
■ D9217	Comm. error code of slave 6	○	○	
■ D9218	Comm. error code of slave 7	○	○	

Communication Error Code of CPU Link (Value of D9212 ~ D9218)

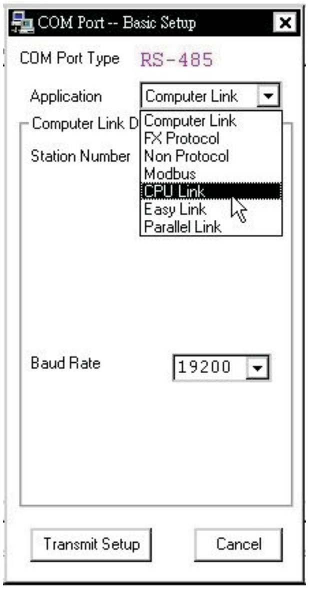
Error Code	Detail
00H	No Error
01H	Communication Time Out error
05H	Communication Check Sum error

◆ Application Example

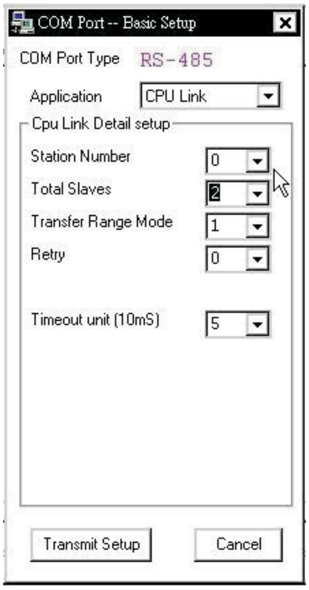
This example connects 3 VB series PLCs through RS-485 interface and executes CPU Link communication, data transfer range choose mode 1. These 3 VB series PLCs have station number of 0 (Master), 1 (Slave) and 2 (Slave) respectively.



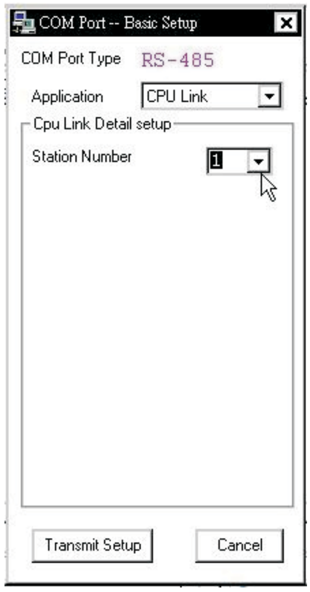
- Set the CP2 parameter for each PLC by Ladder Master though CP1.



Select the application to be CPU Link



Set station number, 0 is master, set master parameters.



Set station number, 1 ~ 7 are slaves.

- When this example executes, the PLC stations will execute the following job as programmed:

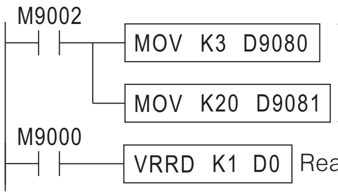
  Master PLC (Station 0): Read value of VR1 and store in register D10, show the content of register D20 on the screen.

  Slave PLC 1 (Station 1): Read value of VR1 and store in register D10, show the content of register D0 on the screen.

  Slave PLC 2 (Station 2): Read value of VR1 and store in register D20, show the content of register D10 on the screen.

- The following result will be produced after the CPU Link communication.
  The value of master VR1 will be shown on the screen of slave 1 (change the master VR1, can see the changes on slave 1 screen also.)
  The value of slave 1 VR1 will be shown on the screen of slave 2 (change the slave 1 VR1, can see the changes on slave 2 screen also.)
  The value of slave 2 VR1 will be shown on the screen of master station (change the slave 2 VR1, can see the changes on master station screen also.)

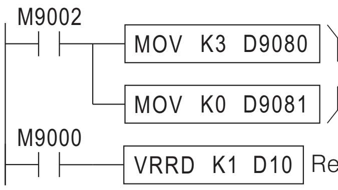
- Program of the Master PLC



Set the display screen to display mode 3 (displays a 4-digits numeric value), and show the value of D20 onto the screen.

Read the value of VR1, and store it to D0.

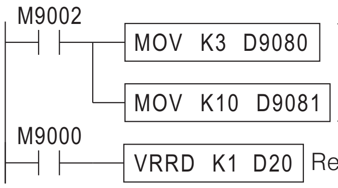
- Program of the Slave 1 PLC



Set the display screen to display mode 3 (displays a 4-digits numeric value), and show the value of D0 onto the screen.

Read the value of VR1, and store it to D10.

- Program of the Slave 2 PLC

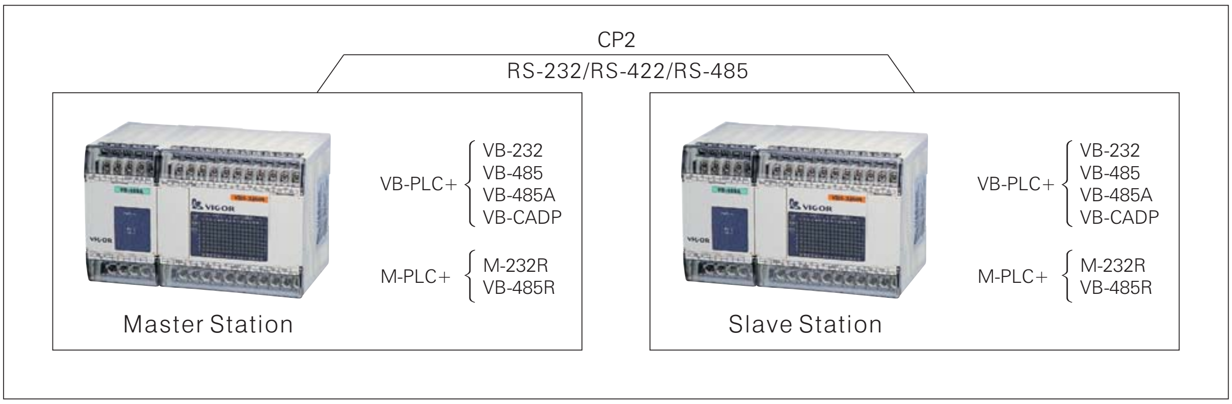


Set the display screen to display mode 3 (displays a 4-digits numeric value), and show the value of D10 onto the screen.

Read the value of VR1, and store it to D20.

B-3-5 Parallel Link

◆ PLC use dedicated communication protocol, and the 2 PLCs in the network transfer data automatically based on configuration settings.



Item		Specification	
Transmission Interface		RS-232	RS-422/RS-485
Communication Protocol		Dedicated Communication Protocol	
Communication Method		Half-duplex	
Baud Rate		4800/9600/19200/38400 bps	
Distance		15 M (49')	1000 M (3280'); (50 M /164', if the network has a VB-485)
Number of Linked Stations		2 stations	
Connection Equipment		VB Series: VB-232 or VB-CADP M Series: M-232R	VB Series: VB-485, VB-485A or VB-CADP M Series: M-485R
Linkable PLC		VB Series, M Series PLC	
Data Transfer Range	Low Speed	Master → Slave: M800～899, D490～499	Slave → Master: M900～999, D500～509
	High Speed	Master → Slave: D490, D491	Slave → Master: D500, D501
Communi-cation Time	Low Speed	73mS + Master Scan Time + Slave Scan Time (Baud Rate = value at 19200 bps)	
	High Speed	14mS + Master Scan Time + Slave Scan Time (Baud Rate = value at 19200 bps)	

- Parallel Link executes communication work after PLC completes the user program execution. Thus, the communication speed is affected by the scan time. As a result, if 2 PLCs need to exchange data fast and instantly, please use CPU Link.

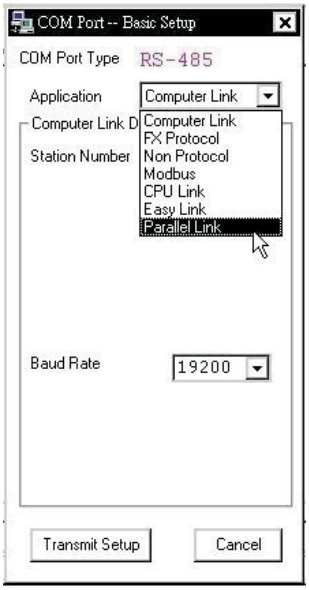
◆ Parallel Link Related Components

Coil ID. No.	Instruction of Function	M	VB	VH
■ M9063	Parallel operation or RS comm. Error. PLC keeps running.	○	○	○
■ M9070	M9070=ON indicates this Unit is master	○	○	
■ M9071	M9071=ON indicates this Unit is slave	○	○	
■ M9072	M9072=ON indicates parallel operation in normal	○	○	
■ M9162	M9162=ON indicates parallel operation high speed transfer. This msg is based on the status of master M9162.	○	○	

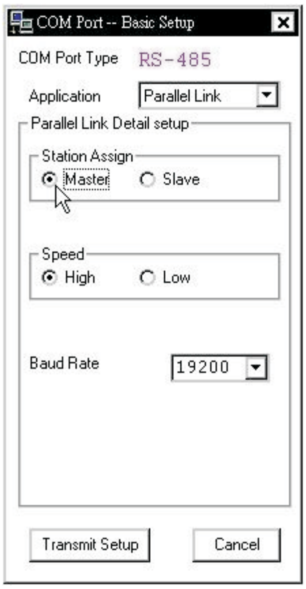
◆ Application Example

This example connects 2 VB series PLCs through RS-485 interface and executes Parallel Link communication, data transfer range choose high speed.

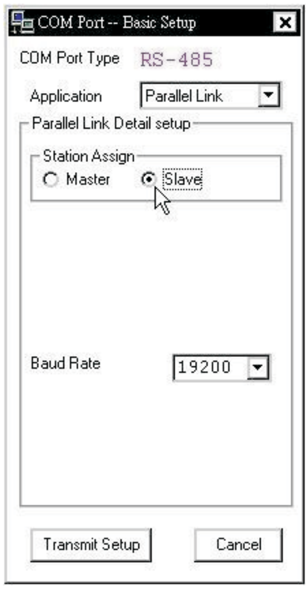
- Set the CP2 parameter for each PLC by Ladder Master though CP1.



Select the application to be Parallel Link



Set station type as master, set data transfer range and baud rate.



Set station type as slave, set baud rate. It must be the same as master baud rate.

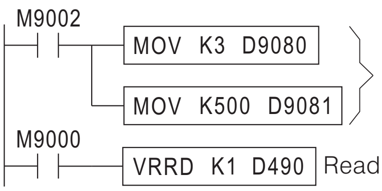
- When this example executes, the 2 PLC stations will execute the following job as programmed:

  Master PLC: Read value of VR1 and store in register D490, show the content of register D500 on the screen.

  Slave PLC: Read value of VR1 and store in register D500, show the content of register D490 on the screen.

- The following result will be produced after the Parallel Link communication.
  The value of master VR1 will be shown on the screen of slave (change the master VR1, can see the changes on slave screen also.)
  The value of slave VR1 will be shown on the screen of master station (change the slave VR1, can see the changes on master station screen also.)

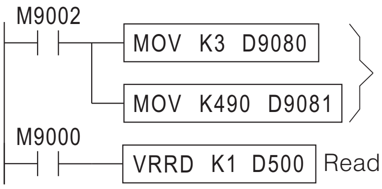
- Program of the Master PLC



Set the display screen to display mode 3 (displays a 4-digits numeric value), and show the value of D500 onto the screen.

Read the value of VR1, and store it to D490.

- Program of the Slave PLC



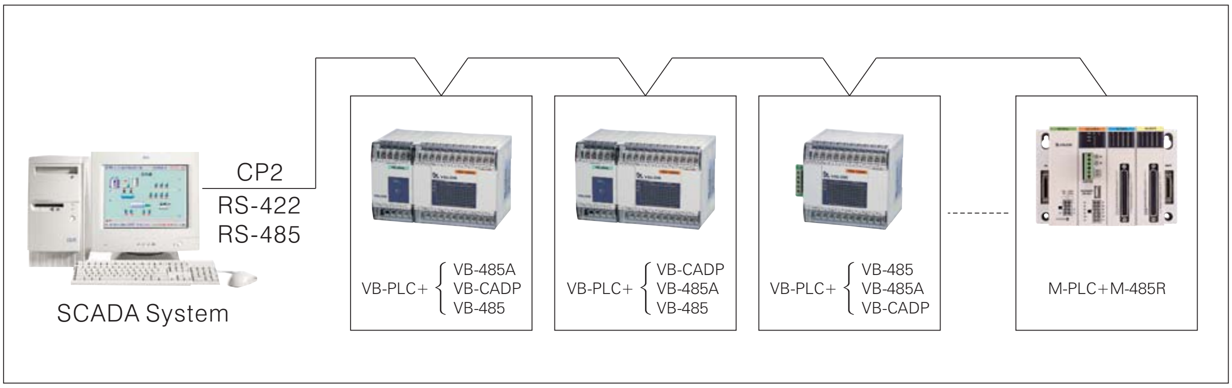
Set the display screen to display mode 3 (displays a 4-digits numeric value), and show the value of D490 onto the screen.

Read the value of VR1, and store it to D500.

B-3-6 MODBUS Communication

◆ MODBUS Passive (Slave) Communication

MODBUS is a popular communication protocol in the market, and is supported by the market sold SCADA and HMI. So when the SCADA or HMI used does not support VIGOR M, VB and VH series communication protocol, MODBUS can be used to communicate with M, VB and VH series PLCs.



Item	Specification	
Transmission Interface	RS-232	RS-422/RS-485
Communication Method	Half-duplex	
Communication Parameters	Communication Mode: ASCII or RTU Parity: None/Odd/Even	Data Length: 7 bits / 8 bits Stop Bit: 1 bit / 2 bits
Baud Rate	300/600/1200/2400/4800/9600/19200/38400 bps	
Distance	15 M (49')	1000 M (3280'); (50 M /164', VB-485)
Number of Linked Stations	1 station	247 stations at most
Connection Equipment	VB and VH Series: VB-232 or VB-CADP M Series: M-232	VB or VH Series: VB-485, VB-485A or VB-CADP M Series: M-485R
Linkable PLC	VB Series, M and VH Series PLC	

M, VB and VH Series PLC Components and MODBUS Components Compare Table

Item	PLC Component No.	MODBUS Component No.
Bit Components	X000 ~ X777	10000 ~ 10511
	Y000 ~ Y777	00000 ~ 00511
	M0 ~ M5119	00512 ~ 05631
	S0 ~ S999	05632 ~ 06631
	T0 ~ T255	06656 ~ 06911
	C0 ~ C255	06912 ~ 07167
	M9000 ~ M9255	07424 ~ 07679
Character Components	D0 ~ D8191	40000 ~ 48191
	T0 ~ T255	48192 ~ 48447
	C0 ~ C199	48448 ~ 48647
	C200 ~ C255	48648 ~ 48759
	D9000 ~ D9255	48760 ~ 49015

- Configuration Method:
  Configure the CP2 communication type of the PLC to be MODBUS by Ladder Master through CP1, set the communication parameters and station number. Every PLC (or equipment) in the communication network must have the same communication parameters.

◆ MODBUS Active (Master) Communication

Many market sold automation components and equipments (like frequency converter, temperature controller…) support MODBUS communication protocol. The VB and VH series PLCs provide MBUS instruction, through which, the VB and VH series PLCs can send command to equipments having MODBUS communication function, and thus exchange data with each other.

◆ Since there are some differences between the MBUS instructions used by VB and VH series PLC, the following chapters will introduce the ways of using MBUS instructions for VB and VH PLCs respectively.

	FNC 149 MBUS		MBUS (S1) (S2)	MODBUS Communication	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					

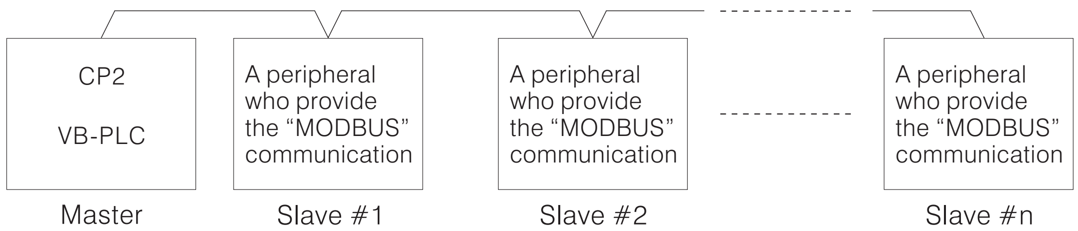
- S2 occupies 4 consecutive registers

```
X20          (S1)   (S2)
--| |------[MBUS D1000 D100]
```

S1 : To indicate the head ID number of receiving/sending data registers

S2 : Instruction working area, occupies 4 consecutive registers

- This section is for VB series PLC only; for the MBUS instruction in a VH series PLC, please revere to page 299.
- When a VB Series Main Unit has been installed a communication card (VB-232R or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between the PLC and a device who has MODBUS communication protocol.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 would like to use for this instruction, the communication type of CP2 should chose the "MODBUS". To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master "System---2nd COM Port Setting..."to get the right setting.
- As the diagram below, use the CP2 to connect the PLC and other peripherals, use the program develop devices (e.g. Ladder Master) to set the "MODBUS" communication mode and the communication parameters. Then, to properly finish all the setting of station IDs (the range of station ID number is 1 ~ 247, but when this system link is used the RS-232, there is only one slave available) and parameters for slaves (or peripherals). Write the data transmission/receiving command to the PLC (Master station), to drive the data transmission between PLCs or peripherals.



- When X20="ON", the MBUS instruction will start to be performed. Based on the designated register string (which initiating from D1000), to process writes/reads data into/from an appointed Slave PLC or peripheral. At the same time, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by (S1) is duly completed, the M9199 will be "ON" for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20="ON" → "OFF", the instruction will be stopped and the data transmission/receiving will be discontinued immediately.
- The MBUS instruction is for the Master PLC, it can be used once only and do not use the LINK or RS instruction in the program.
- For avoid the corresponding breakup, when the MBUS instruction sends a communication request to a particular Slave, if the respondent time of the Slave exceeds the Time-out duration (designated by D9129), the MBUS instruction will stops communication from the specific Slave and operates next communication command.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) ×10ms. When D9129=0 (the default value), the Time-out duration is 100 ms.
- Most of the applied situation is not necessary to change the Time-out duration. But, if an equipment in the communication link, its response is very slow, then the longer Time-out duration is necessary.

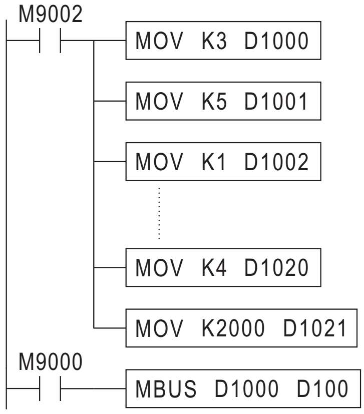
- The register headed with S1 is used to describe the data transmission/receiving information:

S1	Content Value	Description	
D1000	1 ~ 255	To designate the number of transferred and received data sets. Each data transmission/receiving set should be described with 7 registers.	
D1001	1 ~ 247	Designates the Slave station ID number, to proceed data transmission/receiving for the particular Slave station	Description of the 1st data transmission/receiving operation
D1002	1 ~ 3	Instruction commend. 1: read data from the Slave station; 2: write a series of data into the Slave station; 3: write one device's data into the Slave station.	
D1003	1 ~ 64	Length of data transferred or received. If the instruction code (S1+2)= 3, this data will be ignored.	
D1004	1 ~ 6 10,11,13	Designates the device type of the Master station 1: Input Contact X 2: Output Contact Y 3: Auxiliary Coil M 4: State Coil S 5: Timer Contact T 6: Counter Contact C 10: The Present-value Register of the Timer 11: 16-bit Counter, Present-value Register 13: Data Register D	
D1005		Designates the initial component ID number of the Master station device	
D1006	0,1,3,4	Designates the device type of the Slave station 0: A readable/writable bit device 1: A readable only bit device 3: A readable only 16 bits data Register 4: A readable/writable 16 bits data Register	
D1007	0 ~ 32767	Designates the initial component data ID number of the Slave station device	
D1008	1 ~ 247	Designates the Slave station ID number	Description of the 2nd data transmission/receiving operation
D1009	1 ~ 3	Instruction commend	
D1010	1 ~ 64	Length of data transferred/received	
D1011	1 ~ 6 10,11,13	Designates the device type of the Master station	
D1012		Designates the initial component ID number of the Master station device	
D1013	0,1,3,4	Designates the device type of the Slave station	
D1014	0 ~ 32767	Designates the initial component data ID number of the Slave station device	
⋮	⋮		

- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.
- The instruction working area headed with S2 :

S2		Description
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
	Upper 8 bits	Instruction working status 0: Normal data transmission/receiving 2: Error of the length of the transferred/received data (unequal to 1 ~ 64) 4: Error of the designated device type 5: Error of the designated device ID number 6: The characteristic of devices between the Master and Slave stations are different A: Normal communications but no response from Slave stations B: Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

Description by an Example (For the VB series only)



To create a communication table.
This block can be replaced by the function of File Registers

MODBUS communication instruction

There are totally 3 transmission/receiving data sets in this example.

① To read the data in 40000 ~ 40009 of Slave station #5 and put they to D2000 ~ D2009 of the Master station.

② To write the data in D2010~D2014 of the Master station into 41000 ~ 41004 of Slave station #2.

③ To write the data in D2015 of the Master station into 42000 of Slave station #3.

S1	Content Value	
D1000	3	Three transmission/receiving data sets
D1001	5	Designates Slave station #5
D1002	1	Reads data from the Slave station
D1003	10	Length of the data to be read
D1004	13	Designates the device in the Master station which headed with D2000
D1005	2000	
D1006	4	Designates the device in the Slave station which headed with 40000
D1007	0	
D1008	2	Designates Slave station #2
D1009	2	Write a series of data into the Slave station
D1010	5	Length of the data to be written
D1011	13	Designates the device in the Master station which headed with D2010
D1012	2010	
D1013	4	Designates the device in the Slave station which headed with 41000
D1014	1000	
D1015	3	Designates Slave station #3
D1016	3	Write the device's data to the Slave station
D1017	1	This information will be ignored
D1018	13	Designates the data in the Master station D2015
D1019	2015	
D1020	4	Designates the data in the Slave station 42000
D1021	2000	

D1001 ~ D1007: The first transmission/receiving data sets:
40000 ~ 40009 of Slave station #5
↓
D2000 ~ D2009 of the Master

D1008 ~ D1014: The second transmission/receiving data sets:
D2010 ~ D2014 of the Master
↓
41000 ~ 41004 of Slave station #2

D1015 ~ D1021: The third transmission/receiving one data set:
D2015 of the Master
↓
42000 of Slave station #3

- Use the File Registers to set up the communication table
  In the VB series PLC, the File Registers are read only registers and the their contents are assumed as a part of program.
  When a user copy or access the program file, the program itself and the File Registers will be handled together. Since the File Registers have this characteristic, use they to store the communication table were suitable. They are not only to copy the data of File Registers easily but also can minimize the program size. Please refer to CH 2-9 "File Register (D)" for more information about the File Register.
  To plan the contents of File Registers, which can use the programming tool software "Ladder Master", it provide the edit tool "System ---- File Register Edit....", easily to set the data in the registers.

- Edit Communication Table
  In addition to the File Registers' layout function; and further, the Ladder Master provides more user friendly and easily of data input interface, it provide the user to create and edit the Communication Table List.
  Please select the Ladder Master's "Tools ---- Edit Communication Table ...." function to start the Communication Table List document edit window. By the interlocutory pop-up window, user can easily create and edit the communication table step-by-step. After the Communication Table has been finished, the user can put the communication data into the designated File Registers then this communication table is completed. And also, this function provides user to retrieve, access and edit the Communication Table back from the File Registers.
  For the VB series PLCs, the File Register is read-only, and its value will be treated as a part of the user program. When user copy or save program file, the File Register together with the program itself will be copied or saved. This feature makes the File Register very suitable for communication table storing; it can be easily copied from and helps to save PLC program space. For detailed introduction on the File Register, please refer to the section "2-9 File Register (D)".

- Communication Table example :

```
 M9000
|-| |-----[MBUS  D1000  D100]
```

Instruction: MBUS ▼ Start of File Reg: D1000 Length of Reg: 22

Number	Command	Master Data		Slave ID	Slave Data Type	Slave Data #	Length	Word / Bit
1	Read	D2000	<--	5	4	0	10	W
2	Write	D2010	-->	2	4	1000	5	W
3	Single Write	D2015	-->	3	4	2000	1	W

There are totally 3 transmission/receiving data sets in this Communication Table example.

(1) To read the data in 40000 ~ 40009 of Slave station #5 and put they to D2000 ~ D2009 of the Master station.

(2) To write the data in D2010 ~ D2014 of the Master station into 41000 ~ 41004 of Slave station #2

(3) To write the data in D2015 of the Master station into 42000 of Slave station #3.

The "Slave Data Type" and "Slave Data No." in the communication table refers to the component ID number of the slave station equipment.
For example, there is a MODBUS component:

4 0 0 0 0

0000 — The component data ID No.

4 — The component data type 0:Writable & Readable Bit Component
1:Read Only Bit Component
3:Read Only Data Register (16 bits)
4:Writable & Readable Register (16 bits) , the most often type.

FNC 149 MBUS	MBUS S1 S2	MODBUS Communication	M	VB	VH
					○

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					

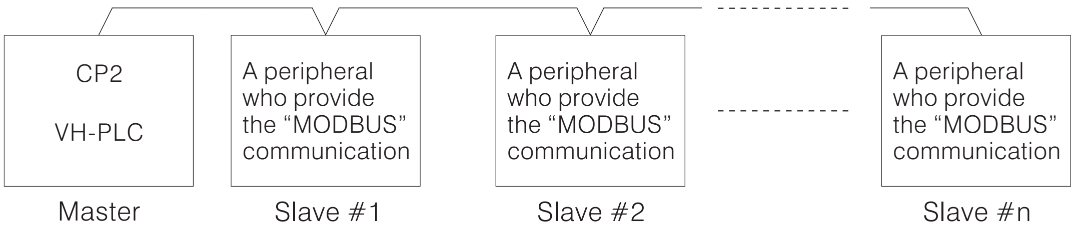
- S2 occupies 4 consecutive registers

```
        S1   S2
 X20
─┤ ├───[MBUS  D0  D100]
```

S1 : To indicate a virtual register for the communication table

S2 : Instruction working area, occupies 4 consecutive registers

- This instruction is for the VH series PLC only. The MBUS instruction for VB series, please referee to page 295
- When a VH Series Main Unit has been installed a communication card (VB-232R or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between the PLC and a device who has MODBUS communication protocol.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 would like to use for this instruction, the communication type of CP2 should chose the "MODBUS". To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master "System---2nd COM Port Setting..."to get the right setting.
- As the diagram below, use the CP2 to connect the PLC and other peripherals, use the program develop devices (e.g. Ladder Master) to set the "MODBUS" communication mode and the communication parameters. Then, to properly finish all the setting of station IDs (the range of station ID number is 1 ~ 247, but when this system link is used the RS-232, there is only one slave available) and parameters for slaves (or peripherals). Write the data transmission/receiving command to the PLC (Master station), to drive the data transmission between PLCs or peripherals.



- When X20="ON", the MBUS instruction will start to be performed. Based on the designated Comm Table string , to process writes/reads data into/from an appointed Slave PLC or peripheral. At the same time, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by S1 is duly completed, the M9199 will be "ON" for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20="ON" → "OFF", the instruction will be stopped and the data transmission/receiving will be discontinued immediately.
- The MBUS instruction is for the Master PLC, it can be used once only and do not use the LINK or RS instruction in the program.
- For avoid the corresponding breakup, when the MBUS instruction sends a communication request to a particular Slave, if the respondent time of the Slave exceeds the Time-out duration (designated by D9129), the MBUS instruction will stops communication from the specific Slave and operates next communication command.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) ×10ms. When D9129=0 (the default value), the Time-out duration is 100 ms.
- Most of the applied situation is not necessary to change the Time-out duration. But, if an equipment in the communication link, its response is very slow, then the longer Time-out duration is necessary.
- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.

- The instruction working area headed with (S2) :

(S2)		Description
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
	Upper 8 bits	Instruction working status 0: Normal data transmission/receiving 2: Error of the length of the transferred/received data (unequal to 1 ~ 64) 4: Error of the designated device type 5: Error of the designated device ID number 6: The characteristic of devices between the Master and Slave stations are different A: Normal communications but no response from Slave stations B: Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

- Edit Communication Table
  In addition to the File Registers' layout function; and further, the Ladder Master provides more user friendly and easily of data input interface, it provide the user to create and edit the Communication Table List.
  Please select the Ladder Master's “Tools ---- Edit Communication Table ....” function to start the Communication Table List document edit window. By the interlocutory pop-up window, user can easily create and edit the communication table step-by-step. After the Communication Table has been finished, the contents will become a part of the user program. The communication commands in the table will go with the user program and keep in VH PLC's system process area. And also, this function provides user to retrieve, access and edit the Communication Table.

- Communication Table Example:

Instruction: MBUS ▼ Length of Reg: 22

Number	Command	Master Data		Slave ID	Slave Data Type	Slave Data #	Length	Word / Bit
1	Read	D200	<--	5	4	0	10	W
2	Write	D210	-->	2	4	1000	5	W
3	Single Write	D215	-->	3	4	2000	1	W

This example is for communication table to execute 3 data receiving/transmitting operations.

(1) To read the data in 40000 ~ 40009 of Slave station #5 and put they to D200 ~ D209 of the Master station.

(2) To write the data in D210 ~ D214 of the Master station into 41000 ~ 41004 of Slave station #2

(3) To write the data in D215 of the Master station into 42000 of Slave station #3.

The “Slave Data Type” and “Slave Data No.” in the communication table refers to the component ID number of the slave station equipment.
For example, there is a MODBUS component:

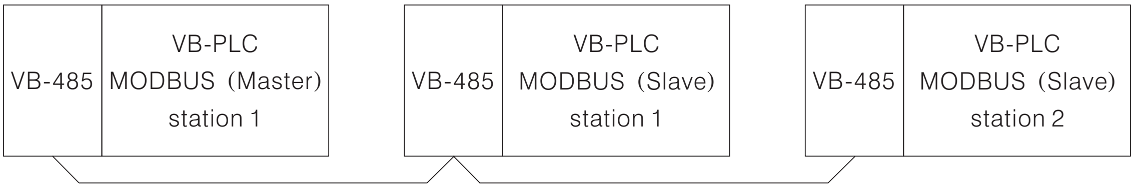
4 0 0 0 0

The component data ID No.

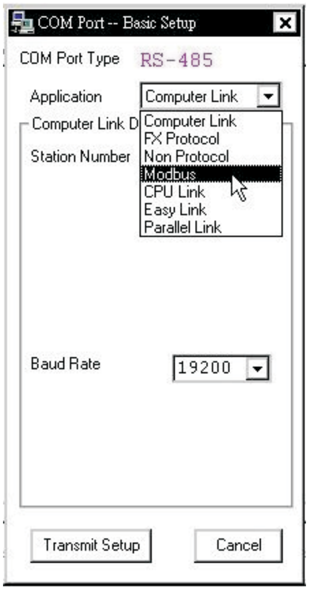
The component data type 0:Writable & Readable Bit Component
1:Read Only Bit Component
3:Read Only Data Register (16 bits)
4:Writable & Readable Register (16 bits) , the most often type.

◆ Application Example

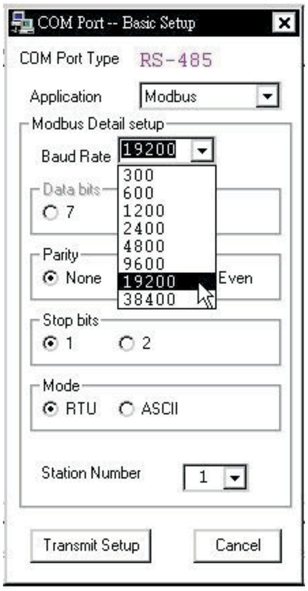
This example connects 3 VB series PLCs through RS-485 interface and executes MODBUS communication. Assign the left most PLC as master station, write MBUS instruction in its program, and use the MBUS instruction to do data receiving/transmitting job with the slave stations, then assign the other 2 PLCs as slave station 1 and slave station 2. In actual application, the slave stations usually are automation components like frequency converter or temperature controller, just for convenience this example use VB-PLC instead.



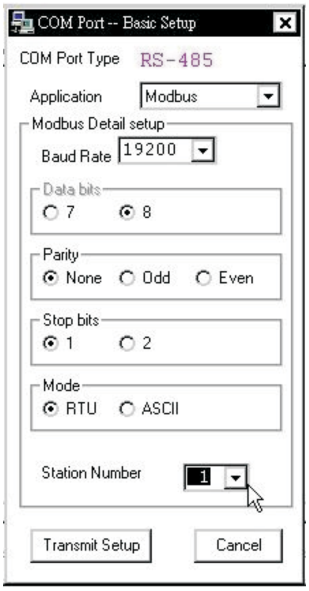
- Set the CP2 parameter for each PLC by Ladder Master though CP1.



Choose the application to be MODBUS



Set station number and the communication parameters



Each PLC (or equipment) in the network should have the same parameters.

- When this example executes, the PLC stations will execute the following job as programmed:

Master PLC: Read value of VR1 and VR2, store in registers D0 and D1, set the display screen to display mode 4 (show 2 two-digit numeric values), displays the content of register D10 on the right, and the content of register D11 on the left.

Slave Station 1: Read value of VR1, change the value range from 0 ~ 255 to 0 ~ 99 and store in register D0. Show the content of register D100 on the screen.

Slave Station 2: Read value of VR1, change the value range from 0 ~ 255 to 0 ~ 99 and store in register D0. Show the content of register D100 on the screen.

- The following result will be produced by the MODBUS communication and master station MBUS instruction.
  The value of master VR1 will be shown on the screen of slave 1, the value of master VR2 will be shown on the screen of slave 2, change the master VR1 and VR2, can see the changes on slave1 and slave 2 screens as well.
  The read value of slave station 1 VR1 (0 ~ 99) will be shown on the right side of the master station PLC screen, and the read value of slave station 2 VR1 (0 ~ 99) will be shown on the left side of the master station PLC screen

- Program of the Master PLC

```
M9002
--| |--+--[MOV K4 D9080]       Set the display screen to display mode 4 (displays two 2-digits numeric values)
       |
       +--[MOV K10 D9081]      Show the value of D10 on the right side of the screen, and show the value of D11
                               on the left side of the screen.
M9000
--| |--+--[VRRD K1 D0]         Read the value of VR1 and store it to D0.
       |
       +--[VRRD K2 D1]         Read the value of VR2 and store it to D1.
M9000
--| |-----[MBUS D1000 D100]    Execute the data receiving/transmitting job of communication table starting
                               from D1000.
```

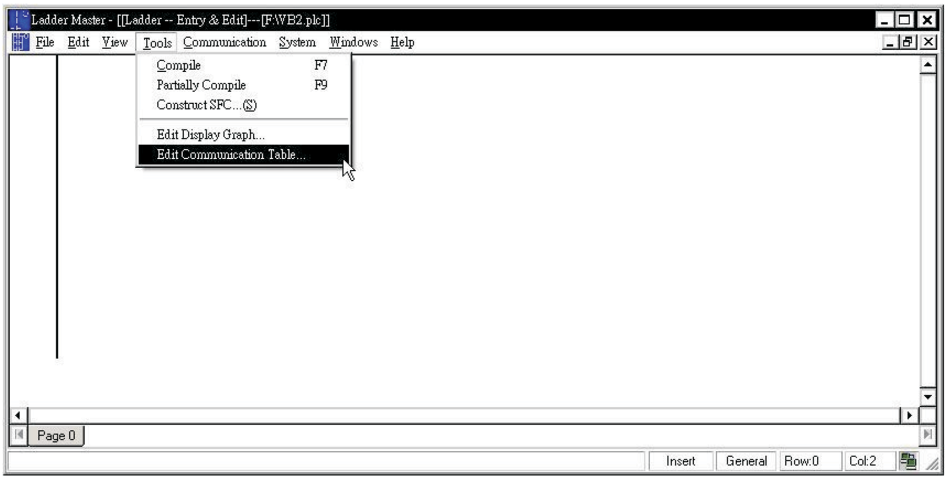
- Program of the Slave 1 and Slave 2 PLC

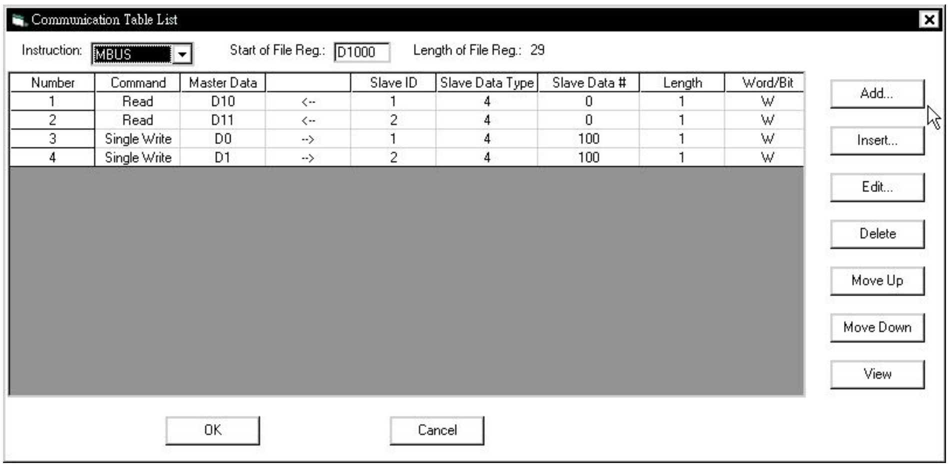
```
M9002
--| |--+--[MOV K3 D9080]       } Set the display screen to display mode 3 (displays a 4-digits numeric value),
       |                       } and show the value of D100 onto the screen.
       +--[MOV K100 D9081]     }
M9000
--| |--+--[VRRD K1 D10]        Read the value of VR1, and store it to D10.
       |
       +--[MUL D10 K99 D12]    }
       |                       }
       +--[DDIV D12 K255 D14]  } D10×99÷255→D0
       |                       } Change the value range of VR1 from 0 ~ 255 to 0 ~ 99 and store in D0.
       +--[MOV D14 D0]         }
```

- Data exchange list between the master station and the slave station.

Master PLC	Data Transmitted Direction	Slave PLC			
		MODBUS Component		VIGOR Component Identify Number	Slave ID #
		Data Type	Data #		
D10	<---	4	0000	D0	1
D11	<---	4	0000	D0	2
D0	--->	4	0100	D100	1
D1	--->	4	0100	D100	2

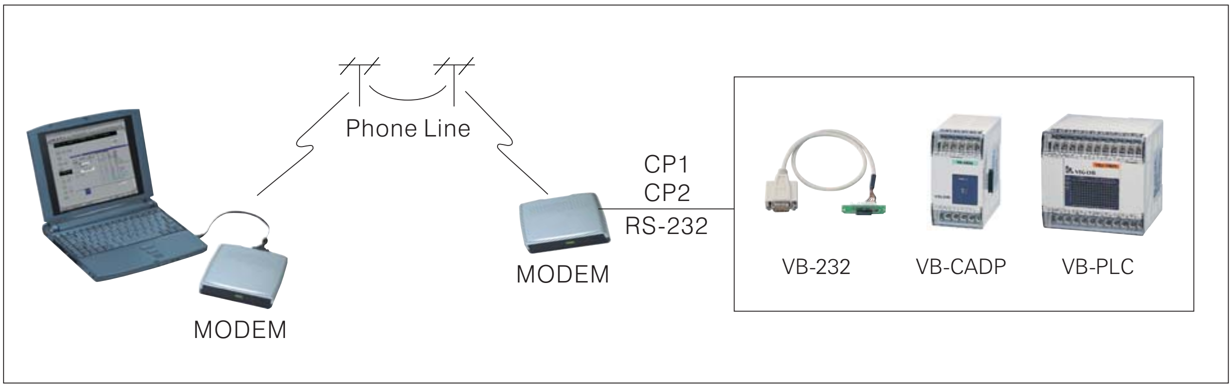
- Edit Communication Table





B-3-7 MODEM Communication

◆ Besides directly communicate with PLC through RS-232 interface, Ladder Master can also use telephone line to communicate with remote PLC though MODEM.



- M, VB and VH series PLCs all have this function, it works using the PLC main unit built in communication port (CP1) or the expansion communication port (CP2).

- When Ladder Master tries to connect to the remote PLC at the other end of the telephone line through MODEM, before Ladder Master starts dialing, the MODEM connecting to PLC should be in “auto-answer” mode (the “AA” LED on the MODEM should be ON), so that it responds the call coming in and builds a connection.

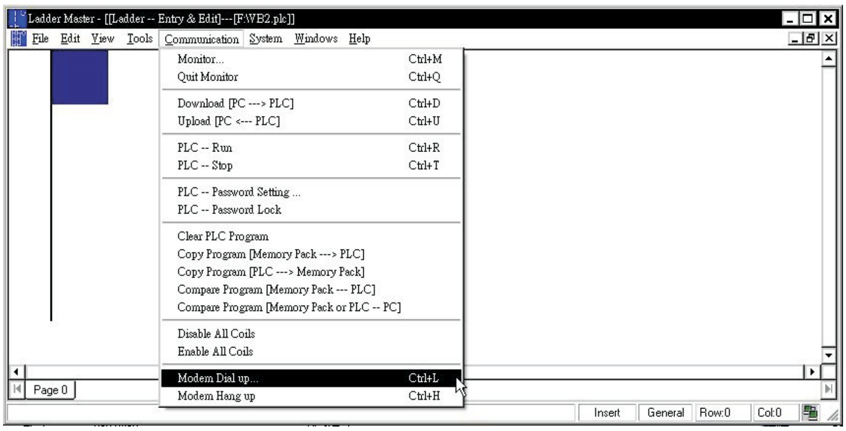
- About the “AA” LED on the MODEM
  There usually is an “AA” (Auto Answer) LED on the MODEM. When this LED is ON, means the MODEM will respond to the dialing in phone call. After the call-and-answer process, these 2 MODEMs have built a connection with each other through phone line. And then the Ladder Master and PLC at the two ends of the phone line can communicate with each other.
  When the users attach the connection lines to the communication ports of MODEM and PLC, switch on the power supply, the “AA” LED on the MODEM should be ON. If the LED is still OFF at this stage, please switch off the MODEM and PLC power. Then switch on the MODEM power first, wait for 5 seconds, switch on the PLC power, and check the MODEM “AA” LED again.
  When use MODEM connection, the user must make sure the MODEM which supposes to answer the coming call is in auto-answer mode (“AA” LED is ON), and the connection may be successful.

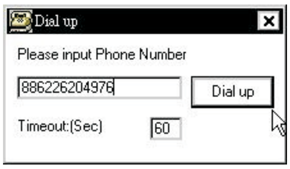
◆ Application Example

In this application example, the computer connects to a MODEM through RS-232 interface (can also use the pc built-in MODEM if any), then connects with the phone line, to be the caller. And the PLC connects to a MODEM through the RS-232 interface of VB-232 (CP2) (can also use the PLC built-in CP1 RS-232 interface), then connects with the phone line, to be the responder.

- When the PLC uses the RS-232 interface of VB-232 (CP2) to connect to the MODEM, please use Ladder Master to set the communication port type of CP2 to be "MODEM".

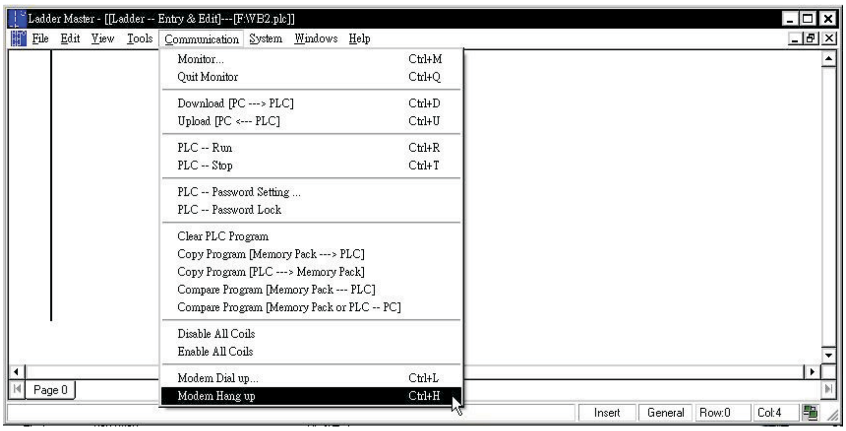
- Use the "Modem Dial Up" function of Ladder Master to connect with the PLC at the other end of the phone line.





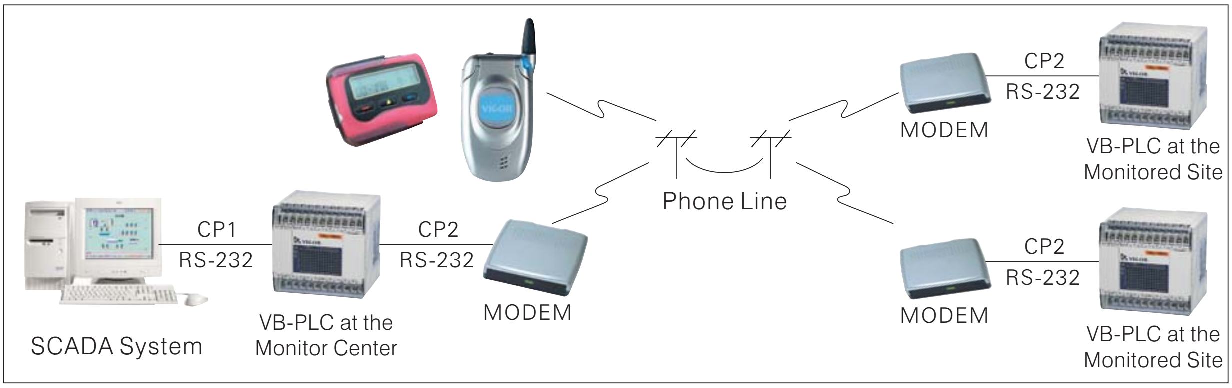
When key in telephone number, like the usual phone dialing, if the computer and PLC are in different regions, add the region code prefix; if they are in different countries, add the country code prefix.

- When the connection ends, use the "Modem Hang Up" function to cut the phone call.



B-3-8 MODEM Dialing

◆ There is a phone number register in the M/VB series PLC, for the user to execute MODEM dialing function through the CP2 communication port. The M/VB-PLC at the monitored site can send data through MODEM dialing to the M/VB-PLC in the monitor center for data collection, or dial to beeper and Mobile Phone for incoming call display. This function is usually used in security system and remote data gathering system.



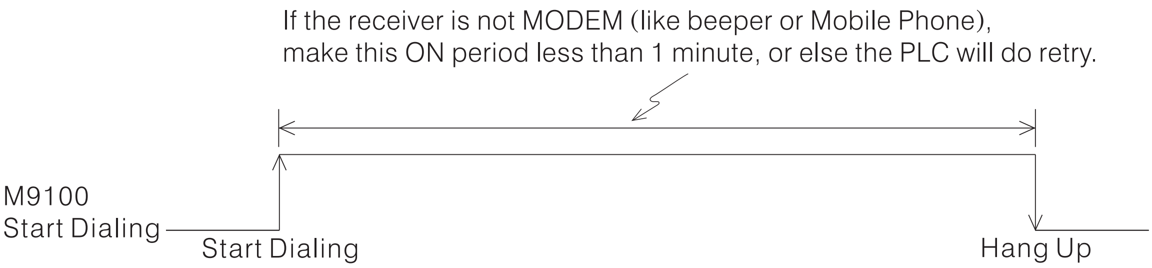
◆ There is a phone number register in the M/VB series PLC, for the user to execute MODEM dialing function through the CP2 communication port. The M/VB-PLC at the monitored site can send data through MODEM dialing to the M/VB-PLC in the monitor center for data collection, or dial to beeper and Mobile Phone for incoming call display. This function is usually used in security system and remote data gathering system.

- Introduction on the M/VB-PLC MODEM Dialing Function

Coil ID. No.	Instruction of Function	M	VB	VH
M9100	CP2 dialing start signal	○	○	
■ M9101	CP2 dialing failed	○	○	
■ M9124	When CP2 of PLC connects with MODEM, M9124 shows the CD signal	○	○	

Register ID.	Instruction of Function	M	VB	VH
D9110 ~ D9115	Telephone number register. Store the telephone number dialing out when execute MODEM dialing function.	○	○	

- M9100: Start dialing. When M9100=OFF→ON, start dialing.
  When M9100=ON→OFF, hang up call.



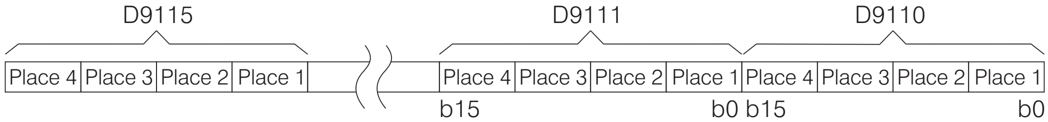
M9101: Dialing Failed. If the load signal (M9124) hasn't been received within 1 minute since the dialing starts, the retry function will be triggered to dial again. Three times of continuously dialing failure means the connection is unsuccessful, and M9101=ON indicates the dialing failed, and will stop trying.

M9124: Load signal. M9124=ON means the MODEM connection is successful, can start sending data.

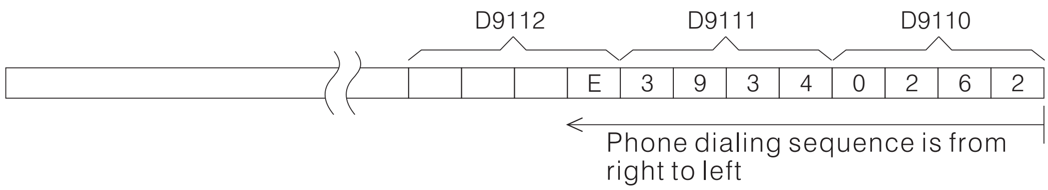
- D9110 ~ D9115 Telephone number registers. Each register can store 4 numbers, in hex code number format.
  The table below lists the meaning of hex code numbers of the telephone number registers.

No.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Code	No. 0	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	" , " Delay	"#"	"*"	—	End Code	—

- The sequence of telephone numbers stored in the registers.



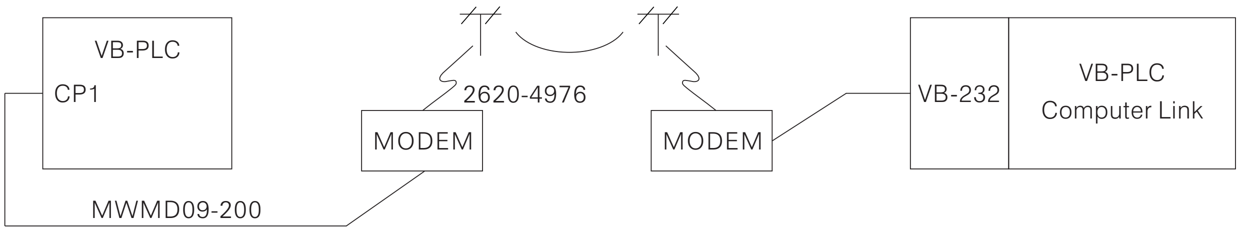
E.g. the dialing number is 2620-4393, then the content of D9110 ~ D9115 is as below:



※ Take note that the register content is in hex code.

◆ Application Example

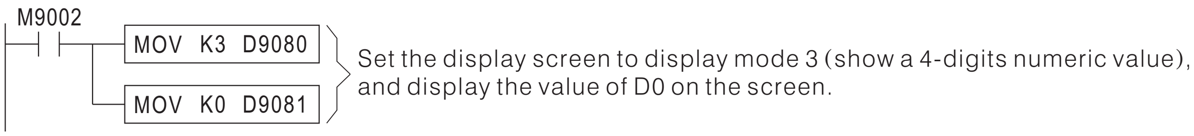
In this example, 2 VB-PLC are connected by MODEM and telephone line as shown in the diagram below. When the dialing condition of the PLC on the right is satisfied, it will dial to connect to the PLC on the left. When the connection is successful, PLC on the right will use LINK instruction (FNC89) to send the relevant data to PLC on the left. When the data sending is completed, cut off the connection. The PLC on the right transmit data to the PLC on the left in this way, obviously, it can also read data from the PLC on the left in the same way. Moreover, more than one PLC on the right side can send data to the PLC on the left side using this method.



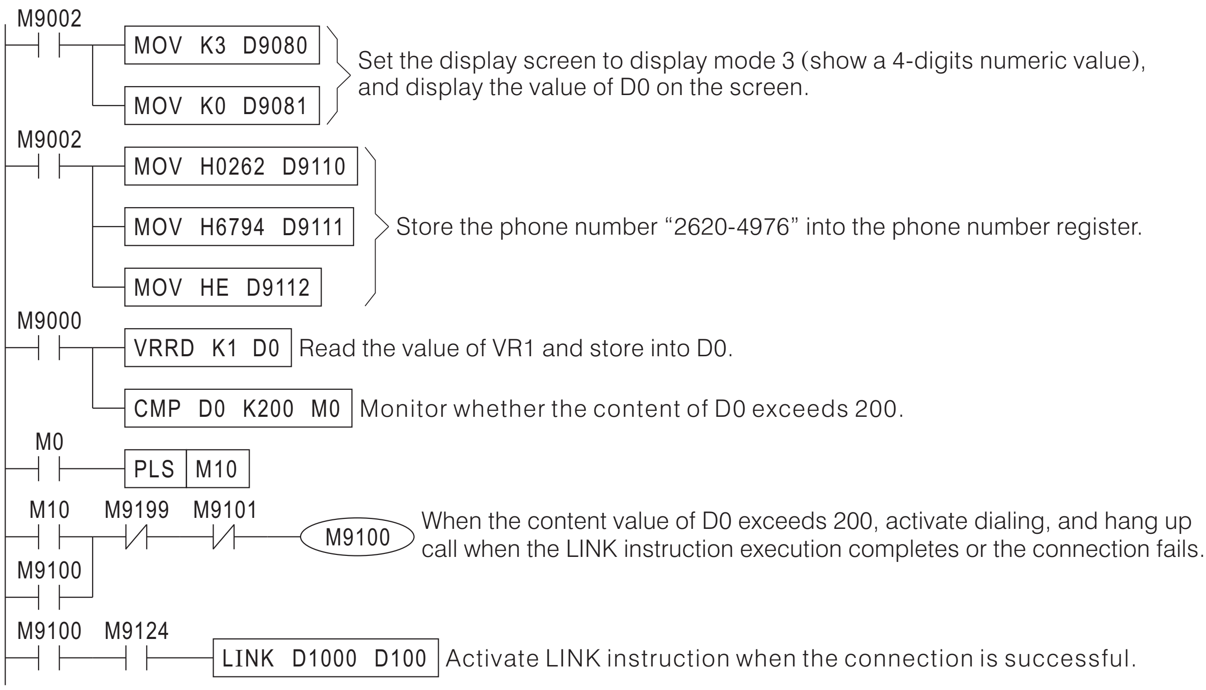
- When this example executes, the PLC on the left and the MODEM will be in auto-answer mode (the "AA" LED on the MODEM will be ON). And as programmed, the PLC on the left will display the content of D0 on the screen.

- The PLC on the right will read the value of VR1 as programmed and store in register D0, and show on the display screen. And when the value of VR1 exceeds 200, it will activate MODEM dialing, and use LINK instruction to send the value of VR1 to the register D0 of the PLC on the left, when the transmitting completes, it will automatically cut off the phone connection. At that time, the display screens of the PLC on the left and the PLC on the right will show same value.

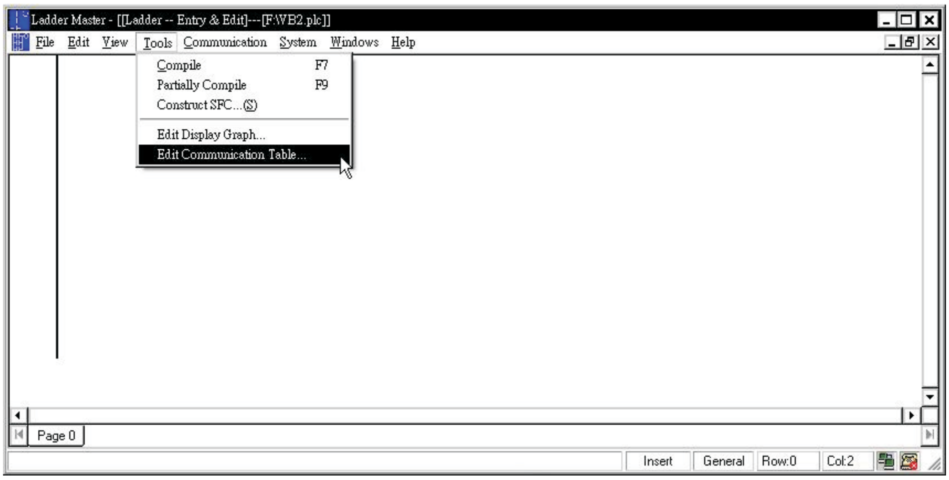
- Program of the PLC on the Left

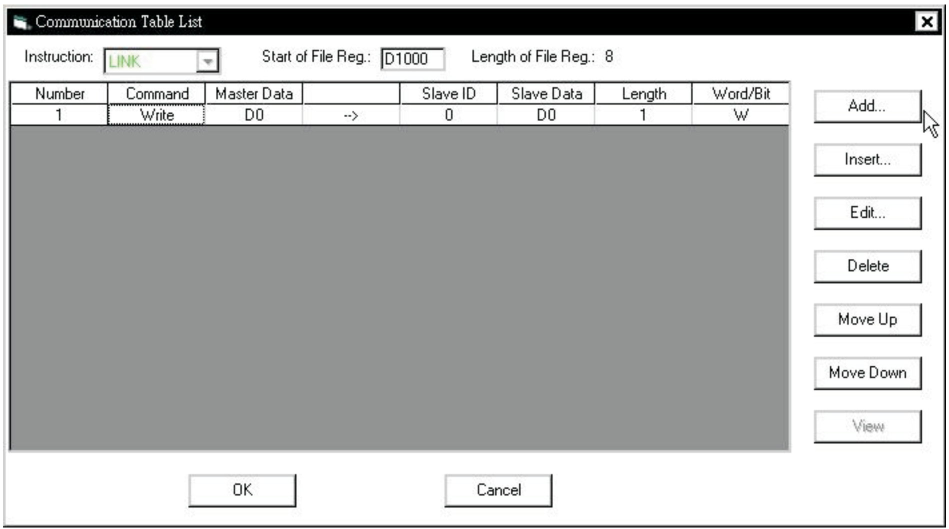


- Program of the PLC on the right



- Edit Communication Table

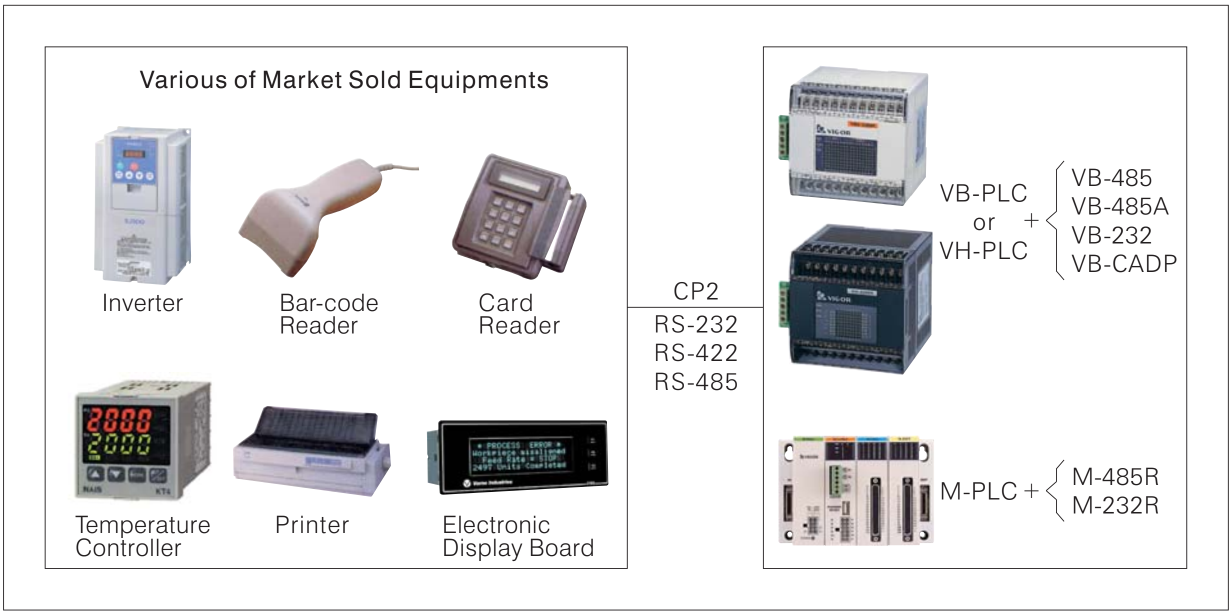




MEMO

B-3-9 Non Protocol Communication

◆ PLCs do not execute any specified communication protocol. All communication programs are defined by the user and completed by the PLC program. Then use the RS instruction (RNC 80) to receive/send communication data and accomplish the communication. This application type is usually used to communicate with market sold temperature controllers, Inverter and bar code readers,etc.



Item	Specification	
Transmission Interface	RS-232	RS-422/RS-485
Communication Protocol	No protocol	
Communication Method	Half-duplex	
Communication Parameters	Baud Rate	300/600/1200/2400/4800/9600/19200 bps
	Data Length	7 bits/8 bits
	Parity	NONE/ODD/EVEN
	Stop Bit	1 bit/2 bits
	Starting Code	None or any
	End Code	None or any
Distance (Refer to the specifications of connection equipments)	15 M (49')	1000 M (3280'); (50 M /164', if the network has a VB-485)
Connection Equipment	VB-232、VB-CADP or M-232R	VB-485、VB-485A、VB-CADP或M-485R
Linkable PLC	VB, M and VH Series PLC	

◆ RS Instruction Related Components

For components with symbol "■" or are missing from the list below, their relay coils cannot be driven by instructions and no data can be written to them.

Coil ID. No.	Instruction of Function	M	VB	VH
■ M9063	Parallel Operation or RS communication Error, PLC keeps running.	○	○	○
M9122	RS Instruction Send Flag	○	○	○
M9123	RS Instruction Receive Done	○	○	○
■ M9124	CD Signal shown by M9124 when CP2 connects with MODEM	○	○	
M9129	Time out during RS	○	○	○

Register ID.	Instruction of Function	M	VB	VH
■ D9063	Error code identifying RS communication error	○	○	○
■ D9122	Num of data left when RS Send	○	○	○
■ D9123	Num of data received by RS	○	○	○
D9124	RS starting point setting		○	○
D9125	RS ending point setting		○	○
D9129	RS time out setting	○	○	○

◆ The next page introduces the way of using RS instruction.

FNC 80 RS	RS (S) (m) (D) (n)	Serial communications Instruction	M	VB	VH
			○	○	○

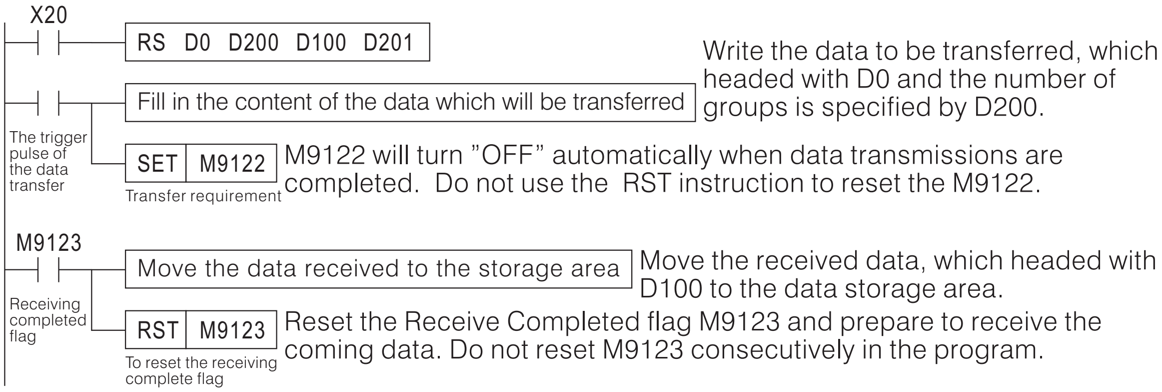
Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
m											○				○	
D											○					○
n											○				○	
• m,n=0 ~ 256																

X20 ── RS D0 (S) D200 (m) D100 (D) D201 (n)

S : Head ID number of the register transferring data
m : Number of groups transferring data
D : Head ID number of the register receiving data
n : Number of groups receiving data

- When M Series PLC's M1-CPU1 module is equipped with the communication expansion card M-232R or M-485R, therefore this CPU module is provided with the CP2 (the second Communication Port). Then the instruction can be used to transfer or receive the data via the serial communications interface of external peripheral facilities.
- When VB and VH Series PLC's Main Unit is equipped with the communication expansion card (VB-232 or VB-485) or expansion module (VB-485A , VB-CADP etc.), therefore this CPU module is provided with the CP2 (the second Communication Port). Then the instruction can be used to transfer or receive the data via the serial communications interface of external peripheral facilities.
- The CP2 is a multi-functional expansion communication port, it can operation various communication types. When the CP2 is assigned to this instruction, the manage type should select to "Non protocol". About the CP2, to select the manage type and related parameter setting, please specify it from the programming software (Ladder Master - System - 2nd COM Port Setting).
- Designate "(m)" as K0 where data transmission (send) is not needed, and designate "(n)" as K0 where data received is not needed.
- As many commercialized peripheral facilities (e.g. Inverters, barcode readers, card readers, electronic displays, etc.) equipped with serial communications interface have their individual protocols, PLC users have to use the RS instruction writing communication programs (in accordance with the communication protocol format of peripheral facilities), when M series PLC is to be connected with peripheral facilities, to transfer data between PLC and those peripherals.
- If the communication of the RS instruction is performed, data transmissions can be divided into 16-bit mode (M9161= "OFF") and 8-bit mode (M9161= "ON").
- M9063 will turn "ON" when any error occurs during data transmissions and receiving and the error code will store in D9063.
- More than one RS instruction can be programmed but only one may be active at any one time.

Sequence of Data Transmissions and Receiving



Related Flags and Data Register

① Transmission Trigger Flag M9122

- When the conditional contact X20= "ON", the RS instruction is performed. At this time, if the pulse signal forces the status of M9122 to be "ON", the content value of the register initiating from D0 will be transferred via the serial interface. When the data transmission is completed, M9122 will be reset to "OFF" automatically.

② Receive Completed Flag M9123

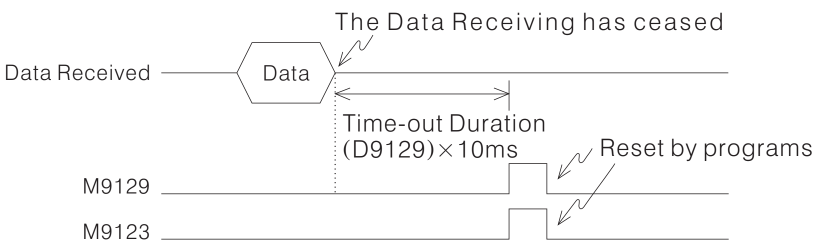
- When the conditional contact X20= "ON", the RS instruction is performed. PLC is ready for the status of receiving.
- When the data receiving is completed, M91213= "ON". At this moment, the received data in the buffer will be moved to the data storage area, and then M9123 will be reset to "OFF". Afterwards, PLC will be ready for the status of receiving immediately.

③ Carrier Detection Flag M9124 (VH Series do not support this signal)

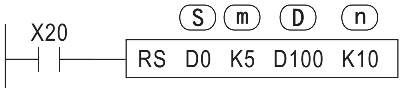
- When PLC receives the CD (Carry Detect) signal from the serial interface, M9124= "ON".
- When PLC is connected with a MODEM, the CD signal is used to represent the status of MODEM. If M9124= "OFF", the transmission of the dialing signal can be performed. If M9124= "ON", data transmissions and receiving can be performed.

④ Time-out Flag M9129

- During the data receiving, if the receiving time exceeds the time-out duration (designated by D9129), M9129 will turns "ON" to represent as the occurrence of Time-out, and also the Receive Completed flag M9123 will be forced "ON" to close the data receiving action.
- The M9129 will not be reset automatically, must using an instruction in the program to reset the status of M9129.
- By applying the Time-out function, PLC will receive the data of transferred from peripheral facilities which is no "End Code" or no length can be predicted.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) × 10ms. When D9129=0 (the default value), the Time-out duration is 100ms.

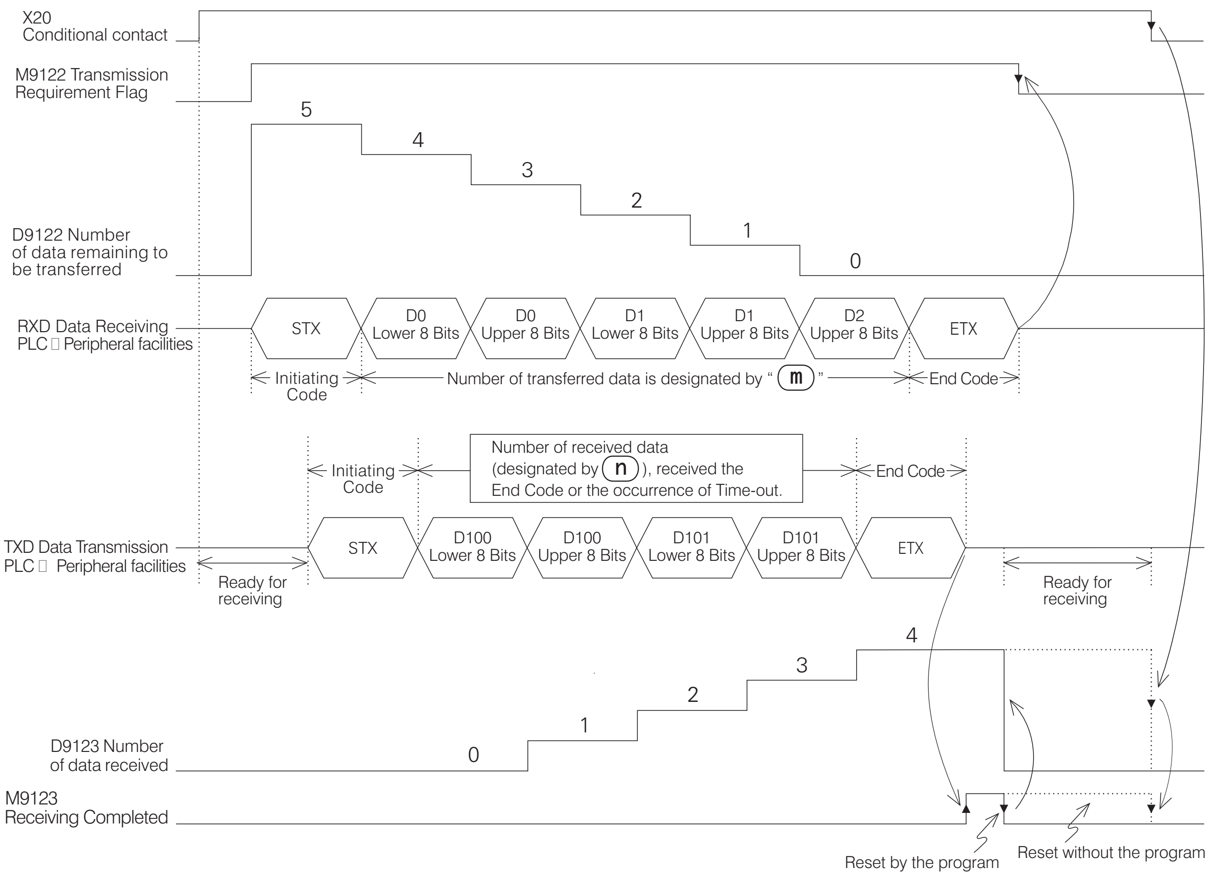


Description of Data Transmissions and Receiving Actions: 16-bit Mode (M9161="OFF")

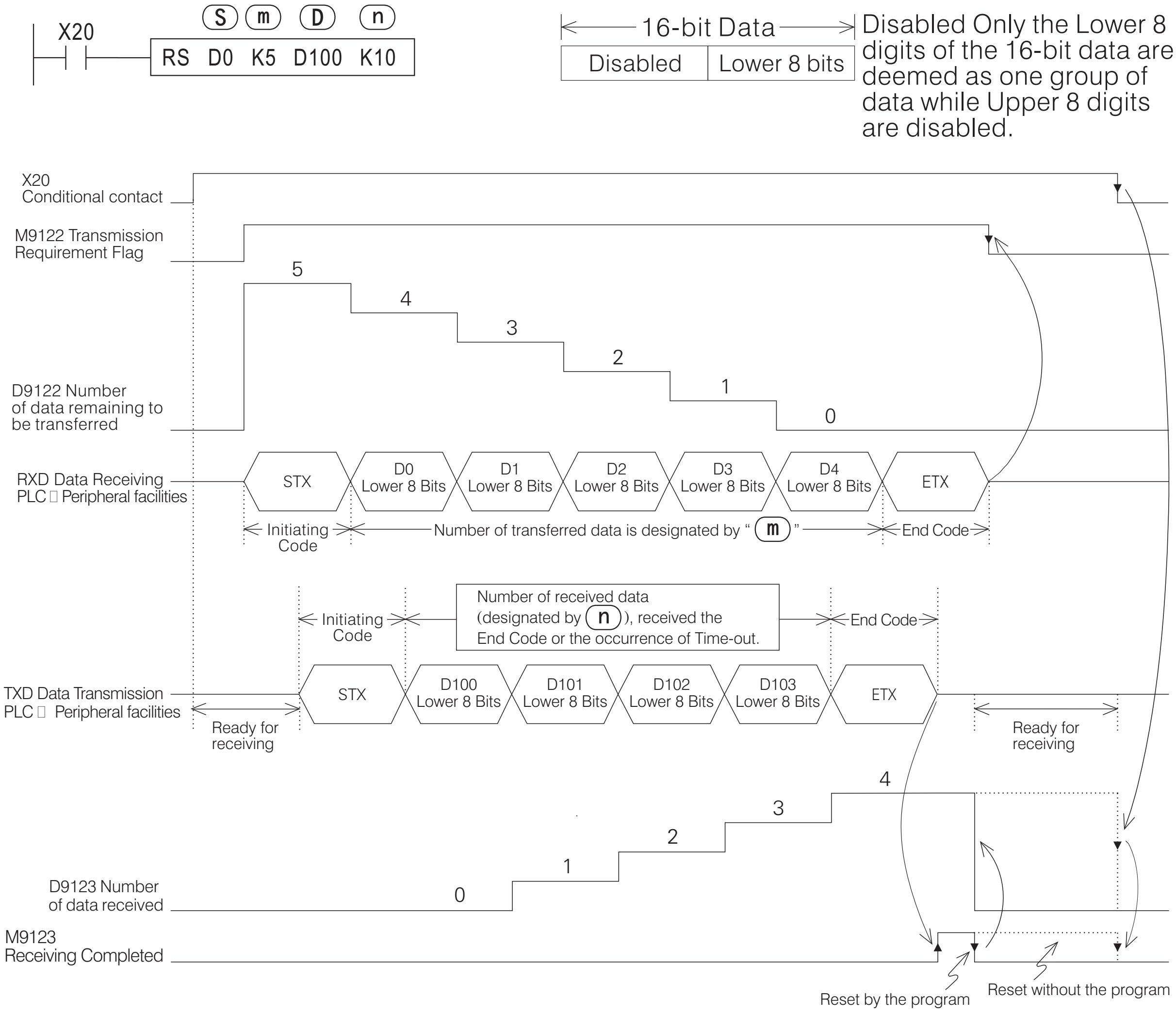


16-bit Data	
Upper 8 bits	Lower 8 bits

Divide the 16-bit data into two groups of data: Upper 8 digit data and Lower 8 digit data.



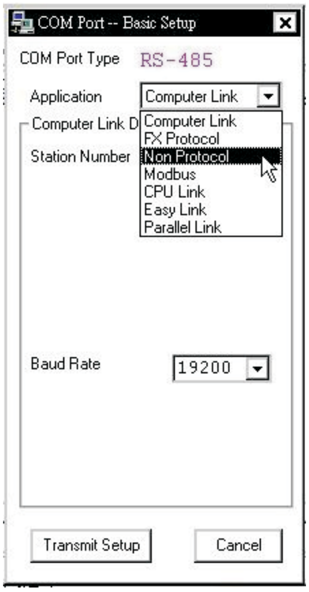
Description of Data Transmissions and Receiving Actions: 8-bit Mode (M9161="ON")



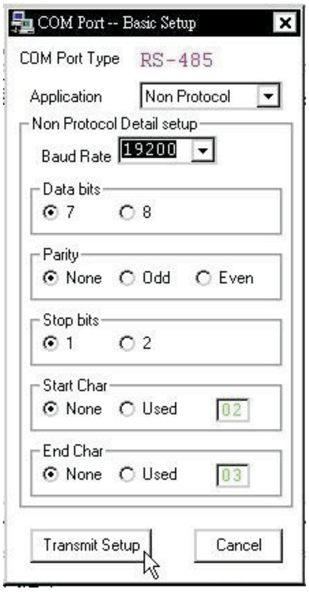
◆ Application Example

In this example, 2 VB series PLCs are connected through RS-485 interface as shown in the diagram below. Set the CP2 application type of the left side PLC to be Non protocol, then write the related instructions for M, VB and VH communication protocol in program to read/send data from/to the station 1 PLC.
Of course, in actual application, the VB series PLC can use CPU Link or Easy Link to exchange data easily without taking such trouble. The purpose of this example is to demonstrate how to use Non protocol and RS instructions. For communication protocols please refer to “Communication Protocol of M, VB and VH Series”.

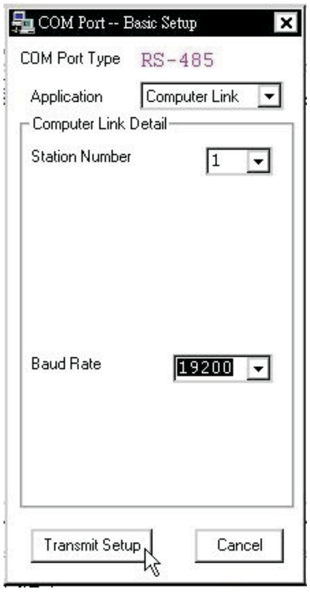
- Set the CP2 parameter for each PLC by Ladder Master though CP1.



Select application type as Non protocol for the left PLC



Set Non protocol parameters based on the M, VB and VH communication protocols.



Select application type as Computer Link for the right PLC, the baud rate must be the same as the left PLC's.

- When this example executes, the 2 PLC stations will execute the following job as programmed:

  Left (Master) PLC: Read value of VR1 and store in register D111, show the content of register D110 on the display screen.
  Execute RS instruction based on the “M, VB and VH communication protocols”. Read the register D0 on the right (slave station 1) PLC and store the value into register D110. Store the content of register D111 to the register D1 on the right (slave station 1) PLC.

  Right (Slave) PLC: Read the value of VR1 and store in register D0, show the content of register D1 on the display screen.

- Since the left (master station) PLC execute RS instruction based on the “M, VB and VH communication protocol” to transmit data, the following result will be produced:
  The VR1 value of left (master station) PLC will be shown on the display screen of the right (slave station) PLC (Change the VR1 of left PLC, can see the changes on the display screen of right PLC).
  The VR1 value of right (slave station) PLC will be shown on the display screen of the left (master station) PLC (Change the VR1 of right PLC, can see the changes on the display screen of left PLC).

- Below is a general introduction on the M, VB and VH communication protocol related instructions used in this example. For the detailed content of the communication protocol, please refer to “B-5 Communication Protocol of M, VB and VH Series”.

  - Communication parameters of the M, VB and VH communication protocols.
    Data length: 7 bits (ASCII) / Parity: EVEN / Stop bit: 1 bit

  - Calculation method of the check code

STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
	0	1	5	1	1	C	0	0	0	2		0	0
02H	30H	31H	35H	31H	31H	43H	30H	30H	30H	32H	03H	30H	30H

Accumulate then take the last two digits (HEX) and convert to ASCII code.

30H+31H+35H+31H+31H+43H+30H+30H+30H+32H+03H=200H

  - Calculation method of the check code

Data to send to slave station 1

STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
	0	1	5	1	1	C	0	0	0	2		0	0
02H	30H	31H	35H	31H	31H	43H	30H	30H	30H	32H	03H	30H	30H

Data responded by slave station 1

ACK	Station No.		Command		Error Code		Byte 1		Byte 2		ETX	Check Code	
	0	1	5	1	0	0	16^1	16^0	16^1	16^0		16^1	16^0
06H	30H	31H	35H	31H	30H	30H					03H		

D0	16^1	16^0	16^1	16^0
	1C01		1C00	

  - Communication instruction to write data into the register D1 of slave station 1 (data address 1C02H)

Data to send to slave station 1

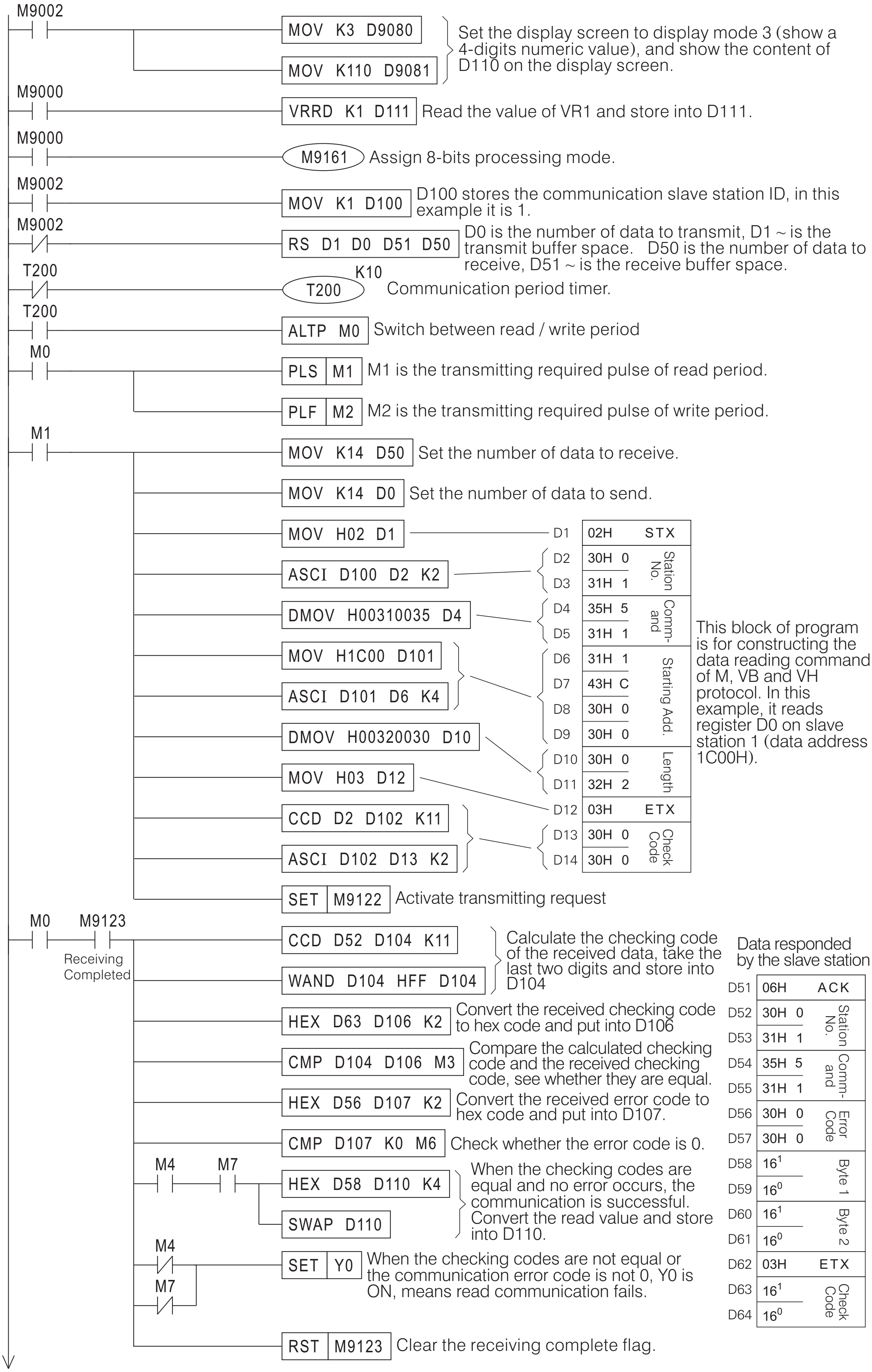
STX	Station No.		Command		Starting Add.				Length		Byte 1		Byte 2		ETX	Check Code	
	0	1	6	1	1	C	0	2	0	2	16^1	16^0	16^1	16^0		16^1	16^0
02H	30H	31H	36H	31H	31H	43H	30H	32H	30H	32H					03H		

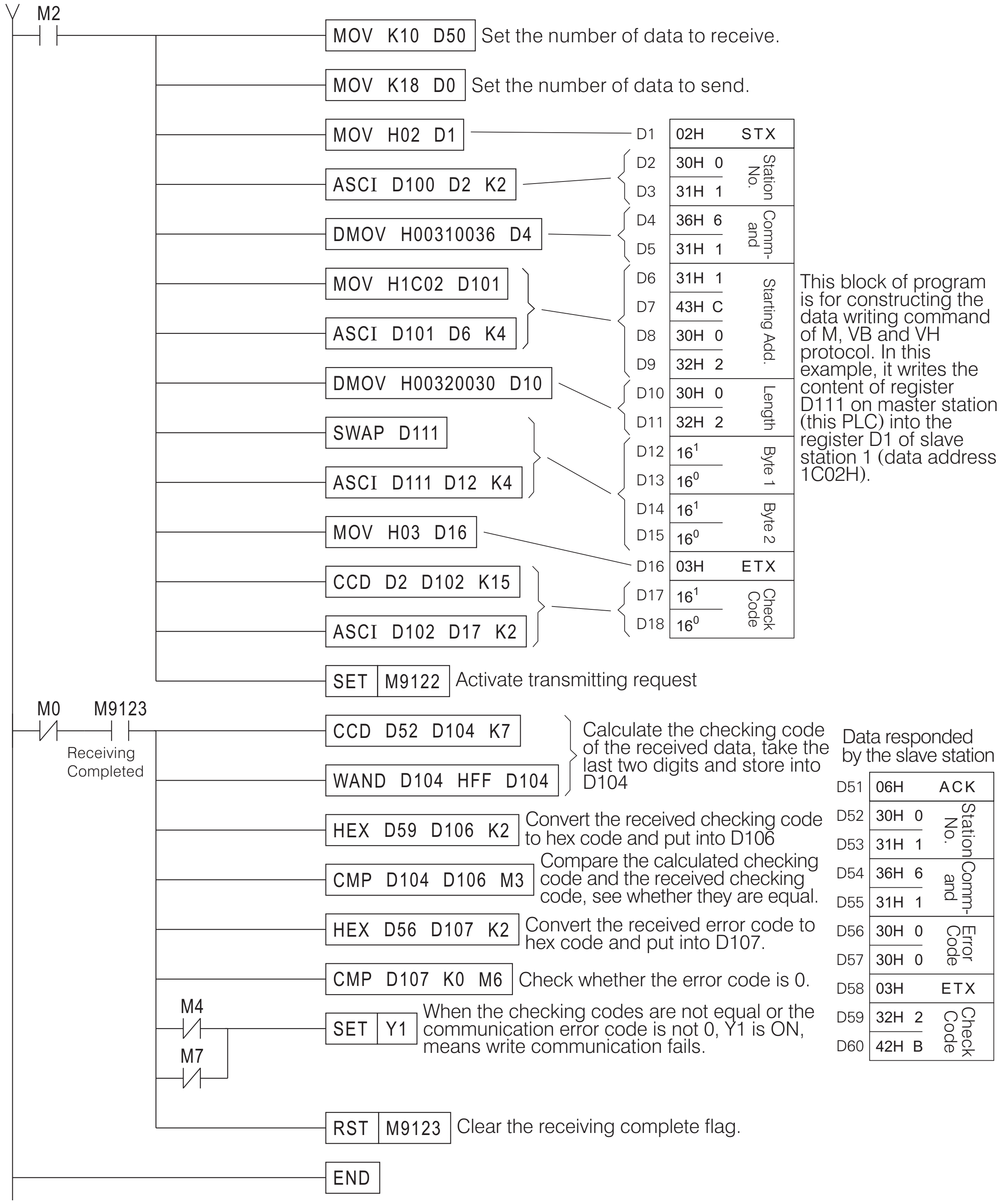
D1	16^1	16^0	16^1	16^0
	1C03		1C02	

Data responded by slave station 1

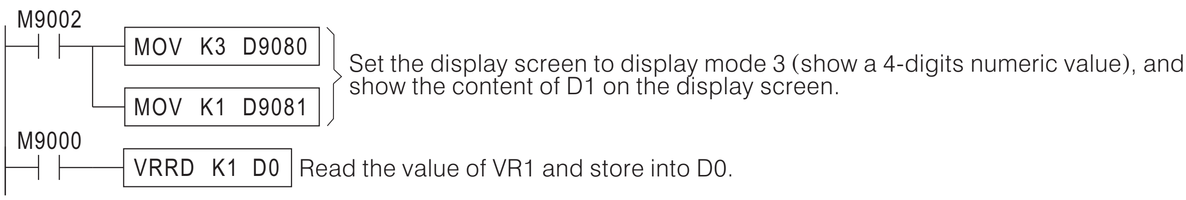
ACK	Station No.		Command		Error Code		ETX	Check Code	
	0	1	6	1	0	0		2	B
06H	30H	31H	36H	31H	30H	30H	03H	32H	42H

- Program of the left (master station) PLC





- Program of the Right side (Slave Station) PLC

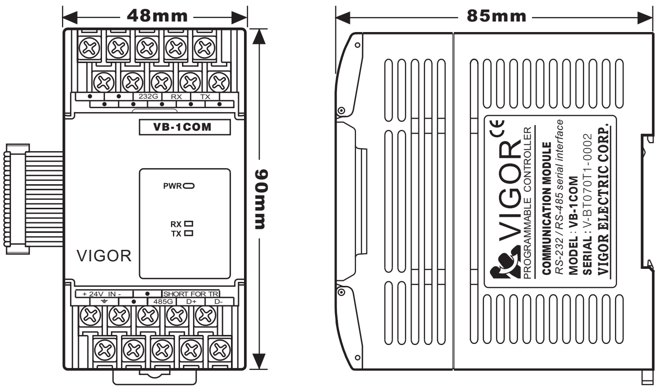


B-4 VB-1COM Serial Link Communication Module

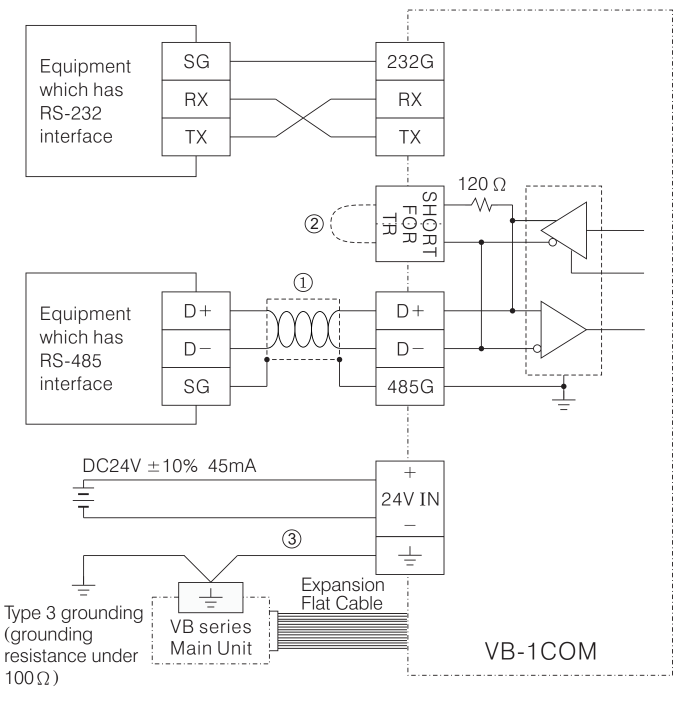
This introduction includes diagrams and texts to guide the user install and use the VB-1COM module correctly. Please read carefully before install and use the VB-1COM module.

B-4-1 Module Introduction

- VB-1COM module has the RS-232 and RS-485 interfaces at the same time, and these 2 interfaces can communication with majority of the equipment sold in market.
- RS-232 and RS-485 are both isolated style, and the distance of RS-485 can be 1000 meters.
- Automatic converting function from HEX to ASCII code of the transmitting/received data is provided.
- The VB series PLCs use the FROM/TO instructions to do data exchange and parameter setting with the VB-1COM.
- Dimensions diagram of this module.



B-4-2 External Wiring



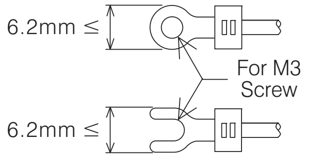
① Please use shielded twisted pair wiring as the connection wire for RS-485 communication interface. In occasions where long distance or high-speed is required, to improve the communication quality, the RS-485 dedicated communication cable (like Belden 9841) is preferred.

② Please parallel attach terminal resistance to the terminals of the communication circuit when construct RS-485 circuit.
The VB-1COM module has a built-in 120 Ω terminal resistance. When short-connect the "SHORT FOR TR" terminals with short connecting wire, the 120 Ω terminal resistance will be parallel connected to the "D+" and "D-" terminals.

③ Please parallel attach terminal resistance to Please connect the terminals of PLC main unit to VB-1COM module, and then use this terminal as type 3 grounding or connect it to the covering case of the distribution box.

④ This module provides RS-232 and RS-485 interfaces, and only one of them can be used at a time.

- Wiring Terminals



- Use O or Y type terminal when wiring as specified in the left hand side diagram.
- Tighten the screw properly to avoid mis-operation. The proper strength used to turn the terminal screw is 5 ~ 8kg-cm.

B-4-3 Module Specifications

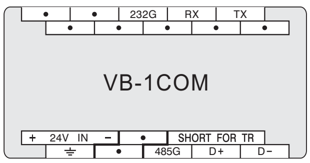
- Common Specifications

Item	Specification
Common Specifications	Same as the VB series Main Unit
Dielectric Strength	500VAC 1 min between all terminals and rack panel

- Power Specifications

Item	Specification
External Driving Circuit	24V DC +10%～−10%，45mA
Internal Circuit	5V DC, 75mA (Power supplied by the internal expansion bus)

- Functional Specifications

Item	Specification	
Transmission Interface	RS-232	RS-485
Isolation Method	Photocoupler isolation	
LED Indicator	PWR、RX、TX	
Communication Distance	15 Meters	1000 Meters
Communication Method	Half-duplex	
Baud Rate	300/600/1200/2400/4800/9600/19200/38400/76800/14400/28800/57600 bps	
Communication Protocol	Non Protocol, user defined communication process done by PLC program.	
Communication Format	Assigned by BFM (9 formats in total)	
PLC Communication	Use FROM/TO instructions through BFM	
Wiring Method	Barrier style terminal block connection	

B-4-4 Buffered Memory (BFM)

◆ BFM Table List for VB-1COM

VB-1COM modules exchange data with the VB series Main Units through the following BFM.

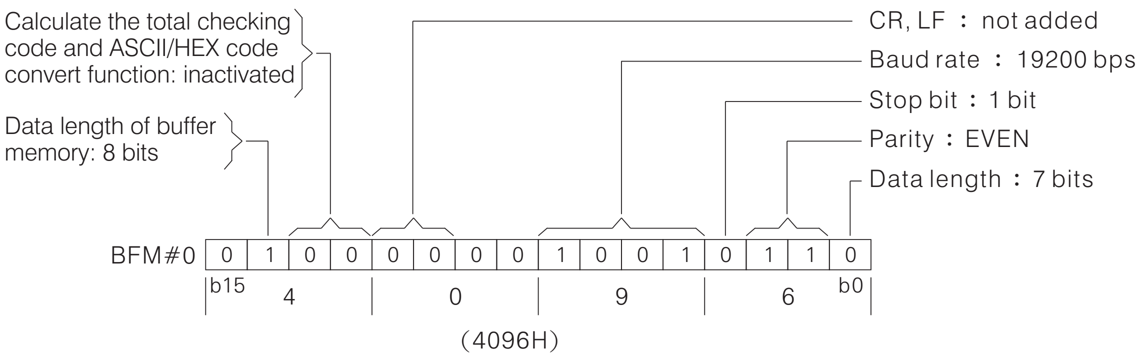
BFM Num	Name	Setting Range	Initial Value	Data Access
#0	Communication Format	—	0087H	W
#1	Command	—	0	W
#2	Upper Limit of byte num received	1 to 512 (when set buffer data length to 16 bits) 1 to 256 (when set buffer data length to 8 bits) "0" indicates "512" or "256"	0	W
#3	Receiving time-out time	0 ~ 4 byte	0	W
#4	Send start code, lower 2 bytes	0 ~ 4 byte	0 (no start code)	W
#5	Send start code, upper 2 bytes			
#6	Send end code, lower 2 bytes	0 ~ 4 byte	0 (no end code)	W
#7	Send end code, upper 2 bytes			
#8	Get start code, lower 2 bytes	0 ~ 4 byte	0 (no start code)	W
#9	Get start code, upper 2 bytes			
#10	Get end code, lower 2 bytes	0 ~ 4 byte	0 (no end code)	W
#11	Get end code, upper 2 bytes			
#13	Num of data left for sending	0 to 512 (when set buffer data length to 16 bits) 0 to 256 (when set buffer data length to 8 bits)	0	R
#14	Byte num of receive buffer memory	0 to 256	0	R
#15	Total checking code of sending data	—	0	R
#16	Total checking code of receiving data	—	0	R
#28	Status	—	0	R
#29	Error Code	—	0	R
#30	Module model ID	—	K7030	R
#1000	Byte num sent	0 to 512 (when set buffer data length to 16 bits) 0 to 256 (when set buffer data length to 8 bits)	0	W
#1001 to #1256	Send buffer	—	0	W
#2000	Byte num received	0 to 512 (when set buffer data length to 16 bits) 0 to 256 (when set buffer data length to 8 bits)	0	R
#2001 to #2256	Receive buffer	—	0	R

◆ Detailed Introduction of BFM

- BFN # 0: communication format

Bit	Introduction	0	1	Initial Value
b0	Data length	7 bits	8 bits	1 ： 8 bits
b1 b2	Parity	(00) ： None (01) ： Odd (11) ： Even		(11) ： Even
b3	Stop bit	1 bit	2 bits	0 ： 1 bit
b4 b5 b6 b7	Baud rate	(0011) : 300 (0100) : 600 (0101) : 1200 (0110) : 2400 (0111) : 4800 (1000) : 9600	(1001) : 19200 (1010) : 38400 (1011) : 76800 (1100) : 14400 (1101) : 28800 (1110) : 57600	(1000) ： 9600 bps
b8 b9	Undefined	—		0 ： undefined
b10 b11	Add CR and LF code	(0 0) ： not used (0 1) ： Add CR code only (1 1) ： Add CR and LF code		(0 0) ： Not add
b12 b13	Calculate total checking code and ASCII/HEX code convert	(0 0) ： not used (0 1) ： activate ASCII/HEX code convert function (1 0) ： activate calculate total checking code function (1 1) ： Activate calculate total checking code and ASCII/HEX code convert function		(0 0) ： not used
b14	Data length of the send/ receive buffer memory	16 bits	8 bits	0 ： 16 bits
b15	Undefined	—		0 ： undefined

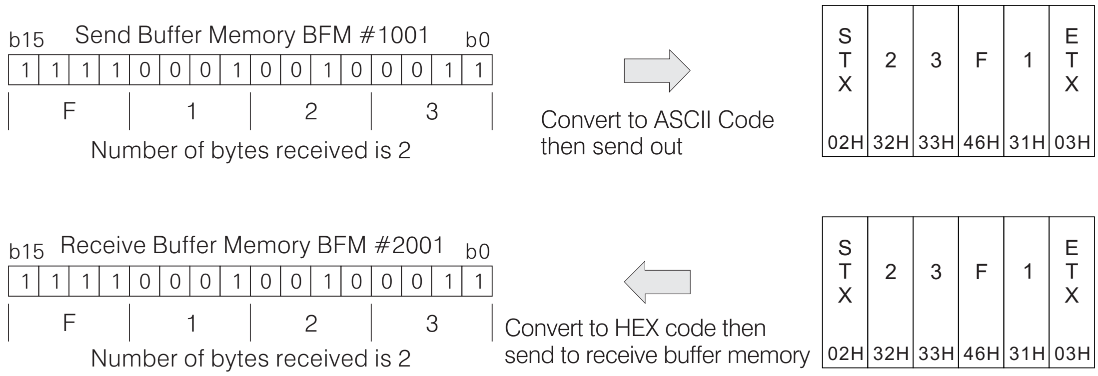
- Configuration Example of communication format (the format need to be configured based on the communication specifications of the corresponding equipment)



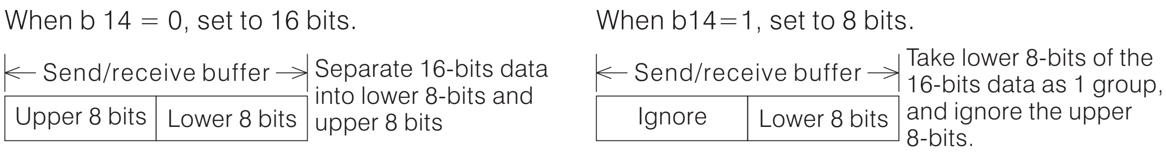
- The VB-1COM module can do the following 9 formats of serial communications

① Data
② Data | Terminator
③ Data | Terminator | CR
④ Data | Terminator | CR | LF
⑤ Header | Data | Terminator | Sum
⑥ Header | Data | Terminator | Sum | CR
⑦ Header | Data | Terminator | Sum | CR | LF
⑧ Data | CR
⑨ Data | CR | LF

- ASCII/HEX codes convert function
  Activate the ASCII/HEX codes convert function, will first convert the HEX codes (0 ~ F) in the send buffer memory to ASCII code then send out. And the received ASCII code data will also be converted to HEX code first then store in receive buffer memory. At this time, the sent/received byte number refers to the byte number of the HEX data.
  The following example demonstrates the converting process when the send/receive data is F123H, with start code STX and end code ETX.



- Data Length of the send/receive buffer



- BFM #1: Command

Bit	Name	Introduction
b0	Activate send/receive	When b0=ON, VB-1COM can send and receive data. When b0=OFF → ON, decide BFM#0 (communication format) and BFM #8 ~ #11 (receive start and end codes), and will clear BFM#28 b3 (error occurred) and BFM#29(error code). So the related data should be prepared before b0=OFFON.
b1	Start send	When b1=OFFON, will decide BFM#4 ~ 7 (send start and end codes), and start sending out the data in send buffer. And when the send completed the BFM#28 b0 (send complete) will be ON. Before giving the send start command next time, the BFM#28 b0 will be automatically cleared to OFF.
b2	Clear command when receive completed	When b2=ON, it decides BFM#8 ~ 11 (receive start code and end code), and clears BFM#28 b1 (receive completed) and the receive buffer. When data receive is completed (BFM#28 b1 turns ON), b2 must=ON, so that BFM#28 b1 will be cleared, otherwise VB-1 COM will not be able to receive the next data.
b3	Clear error	When b3=ON, BFM#28b3 (error occurs) and BFM#29 (error code) will both be cleared.

- BFM #2: the upper limit of the number of bytes received
  When the number of data bytes received in receive buffer become equal to the value of BFM #2, BFM #28 b1 (receive complete) will ON, indicates that receive completed.

- BFM #3: receive time-out time
  BFM #3 is used to set the maximum waiting time between 2 bytes in the data receive process. When the configured time past after a byte data is received, and the next byte of data has not arrived, BFM #28 b2 (receive time-out) will be ON, and the BFM #28 b1 (receive complete) will be ON too, indicate receive completed.

- BFM #5, #4: send start code
  VB-1COM can configure a 0 ~ 4 bytes send start code, when the setting value is 0, means no send start code, and this byte will not be sent. For the actual sending, the send sequence of start code is the 4th, 3rd, 2nd, 1st.
  The start code of this example is 02H (STX)

BFM #5 (upper 2 bytes)

b15															b0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4th		3rd	
0	0	0	0

BFM #4 (lower 2 bytes)

b15															b0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

2nd		1st	
0	0	0	2

- BFM #7, #6: send end code
  VB-1COM can configure a 0 ~ 4 bytes send end code, when the setting value is 0, means no send end code. The storage format of the send end code and sending sequence is the same as the send start code.
  User has to assign a 01H ~ 1FH ASCII code to the 1st byte of the send end code, and this rule does not apply to the 2nd~4th bytes.

- BFM #9, #8: receive start code
  VB-1COM can configure a 0 ~ 4 bytes receive start code, when the setting value is 0, means no receive start code.
  The storage format of the receive start code is the same as the send start code.
  For the actual receiving, the receive sequence of start code is the 4th, 3rd, 2nd, 1st.

- BFM #11, #10: receive end code
  VB-1COM can configure a 0 ~ 4 bytes receive end code, when the setting value is 0, means no receive end code.
  The storage format of the receive end code and receiving sequence is the same as the receive start code.
  In the data receiving process, if the end code set by BFM #11, #10 is received, BFM #28 b1 (receive complete) will be ON to indicate receive completed.

- BFM #13: number of data left for sending
  In the data sending process, the number of data byte waiting to be sent out in the send buffer.

- BFM #14: number of data byte received in the buffer
  In the data receiving process, the number of actual data byte received in the receive buffer.

- BFM #15: total checking code of the send data
  BFM #16: total checking code of the receive data
  The calculation method of the total checking code provided by this module is as below:

Start code											End Code	Total Checking code	
STX	0	0	5	1	0	0	8	1	0	2	ETX	F	4
02H	30H	30H	35H	31H	30H	30H	38H	31H	30H	32H	03H	46H	34H

After accumulating, take the last 2 digits (HEX) and convert to ASCII code.

30H + 30H + 35H + 31H + 30H + 30H + 38H + 31H + 30H + 32H + 03H = 1F4H

- BFM #28: status

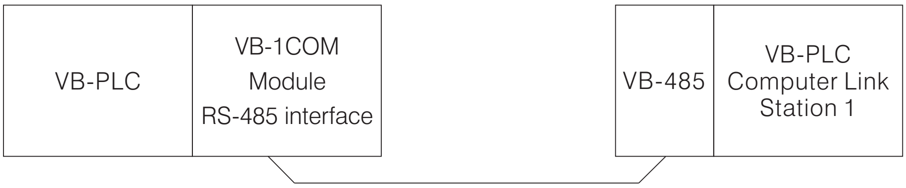
Bit	Name	Introduction
b0	Send complete	When the number of data sent out is equal to the set value of BFM#1000 (number of bytes sent), indicates send complete, and b0 turns ON. Before the next send start command (BFM#1 b1), b0 will be automatically cleared to OFF.
b1	Receive complete	Receive buffer has received the number of bytes as configured By BFM #2 (upper limit of number of bytes to receive), or the configured end code has been received, or time-out happens, VB-1COM takes as receive completes, b1 turns ON. After b1=ON, BFM #1 b2 (receive complete clear) has to be used to clear b1 to OFF. Otherwise, VB-1COM will not be able to receive the next data.
b2	Receive time-out	BFM #3 sets the receive time-out time. When receive time-out happens, b2 turns ON. And then BFM #28 b1 will turn ON to indicate receive completed. When BFM#1 b2 (receive complete clear) command is executed, b2 will be cleared to OFF as well.
b3	Error occur	If any error happens in the data sending/receiving process, b3 will be ON, and the error code will be store into BFM #29.
b6	Data sending	From the sending start command (BFM#1 b1) is given, until sending completed (BFM #28 b0 status turns ON), b6 will be ON.
b7	Data receiving	From the first character is received, until receiving completed (BFM #28 b1 turns ON), b7 will be ON.

- BFM #28: status

Num	Introduction	Possible Causes
0	No error	—
1	Receive parity error, overrun error,framing error	Invalid communication format causes control sequence error.
2	Undefined	—
3	Defective receive character	The data received is not ASCII code.
4	Receive sum check error	The total checking code received is not equal to the calculated total checking code (BFM#16).
5	Undefined	—
6	Baud rate setting error	Configured a non-existing baud rate.
7	Receive CR error	CR is not in the correct position.
8	Receive LF error	LF is not in the correct position
9	Send/receive initial terminator setting error	The 1st byte of the end code is not within 01H~1FH range.
10	Receive terminator error	The receive end code is not in correct position or is inconsistent with the configured value.

B-4-5 Programming Example

The VB-1COM communication module is normally used to connect with devices which do not have VIGOR “M, VB and VH communication protocol” like the market sold temperature controller, frequency converter, etc.
To make it easier to understand, here use “M, VB and VH communication protocol” as example, to introduce how to use VB-1COM module to connect to VB series PLC through proper program planning. Firstly, 2 VB-PLCs are connected as shown in the diagram, the left PLC uses the RS-485 interface of its VB-1COM to connect to the VB-485 interface of the VB-1COM of the right PLC. Set the CP2 application type of the right PLC to Computer Link. Then write communication program in the left PLC using “M, VB and VH communication protocol” format, send communication command to the right PLC through the RS-485 interface of VB-1COM module, and then read/write data from/to the right PLC.



- The right side PLC has to configure CP2 parameter through CP1 by Ladder Master.
  The application type of CP2 is set to Computer Link, the baud rate is set to 19200, and the communication station number is set to 1.

- This program example will give 2 application examples for user reference, and the execution results of these 2 programs are exactly the same.
  Program example 1: make no use of the start code, end code setting and calculate total checking Code, ASCII/HEX convert functions, treat the start code, end code and check codes of the communication format as parts of the data array, then use program to analyze the data array and read/write the transmission data. Since it is a common way of connecting to other devices using VB1COM, users need to understand this example thoroughly.
  Program example 2: activate the start code, end code settings and calculate total checking codes, ASCII/HEX code convert functions provided by VB-1COM. If the communication protocol format of the devices connected to VB-1COM corresponds with the auxiliary function definition, activate the auxiliary functions can help to improve the efficiency of the written communication program.

- The 2 PLC will execute the following actions as programmed when this application example executes.
  Left (master station) PLC: read value of VR1 and store into register D111, then show the content value of register D110 on the display screen.
  Write communication program according to the “M, VB and VH communication protocol”. Read the register D0 of the right (slave station 1) PLC, and store this value into register D110. Then write the content value of register D111 to the register D1 on the right (slave station 1) PLC.
  Right (slave station 1) PLC: read value of VR1 and store into register D0, then show the content value of register D1 on the display screen.

- Since the left (master station) PLC writes communication program based on the “M, VB and VH communication protocol” and transmit data, the following result will be generated.
  The read value of VR1 on the left (master station) PLC will be shown on the display screen of the right (slave station) PLC. Change the VR1 of left PLC, can see the changes on the display screen of the right PLC.
  The read value of VR1 on the right (slave station) PLC will be shown on the display screen of the left (master station) PLC. Change the VR1 of right PLC, can see the changes on the display screen of the left PLC.

- Below is a simplified introduction of the related instructions used in this application example of "M, VB and VH communication protocol". For the detailed content of the communication protocol, please refer to the specifications in "B-5 Communication Protocol of M, VB and VH Series".

  - Parameters of the M, VB and VH communication protocol
    Data length: 7 bits (ASCII) /Parity: EVEN/Stop bit: 1 bit

  - Calculation method of the checking code

STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
	0	1	5	1	1	C	0	0	0	2		0	0
02H	30H	31H	35H	31H	31H	43H	30H	30H	30H	32H	03H	30H	30H

After accumulating, take the last 2 digits (HEX) and convert to ASCII code.

30H+31H+35H+31H+31H+43H+30H+30H+30H+32H+03H=200H

  - Communication instruction to read the value of register D0 of slave station 1 (data address: 1C00H)

Data to send to slave station 1

STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
	0	1	5	1	1	C	0	0	0	2		0	0
02H	30H	31H	35H	31H	31H	43H	30H	30H	30H	32H	03H	30H	30H

Data responded by slave station 1

ACK	Station No.		Command		Error Code		Byte 1		Byte 2		ETX	Check Code	
	0	1	5	1	0	0	16^1	16^0	16^1	16^0		16^1	16^0
06H	30H	31H	35H	31H	30H	30H					03H		

	1C01		1C00	
D0	16^1	16^0	16^1	16^0

  - Communication instruction to write data into register D1 (data address: 1C02H) on slave station 1

Data to send to slave station 1

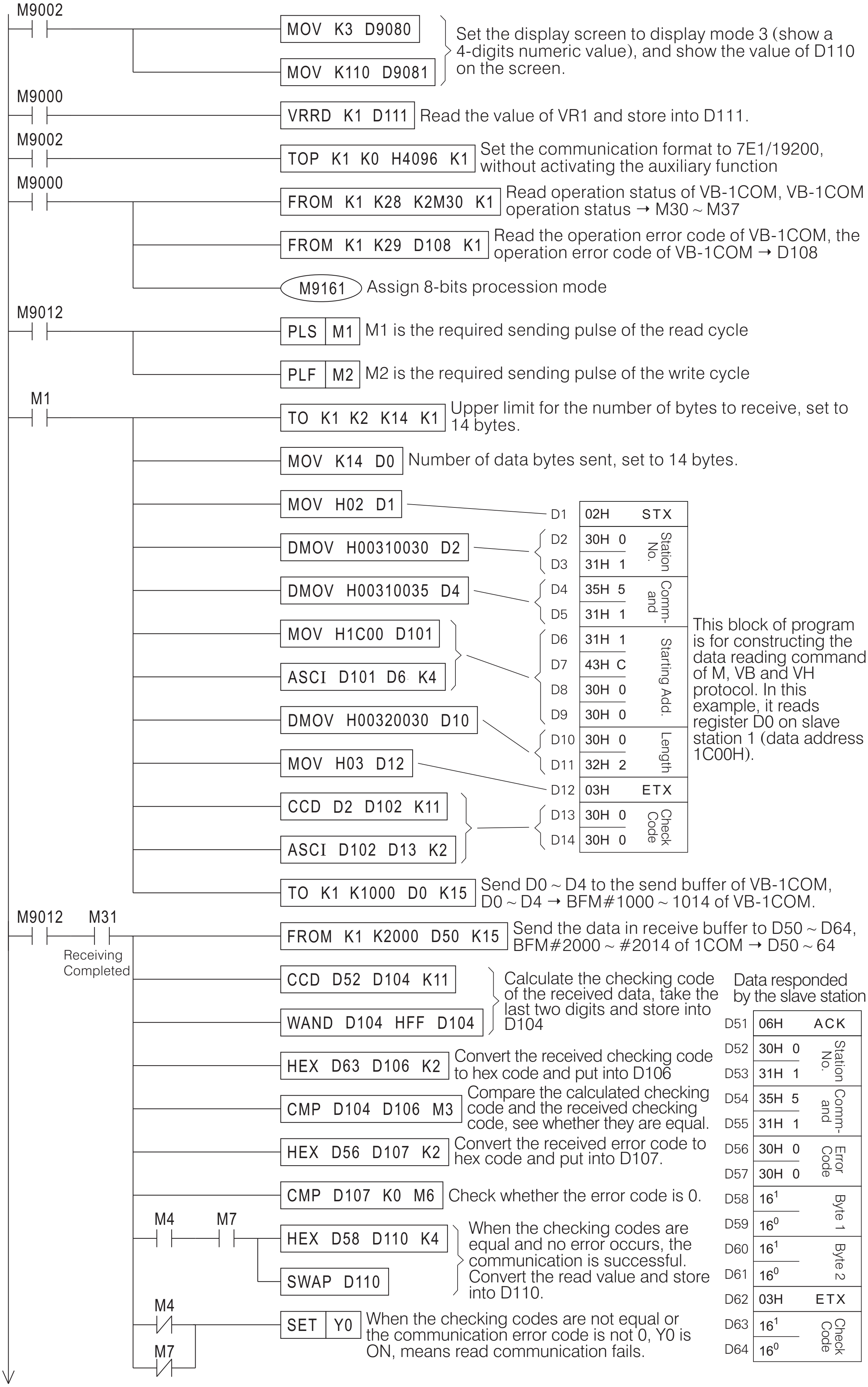
STX	Station No.		Command		Starting Add.				Length		Byte 1		Byte 2		ETX	Check Code	
	0	1	6	1	1	C	0	2	0	2	16^1	16^0	16^1	16^0		16^1	16^0
02H	30H	31H	36H	31H	31H	43H	30H	32H	30H	32H					03H		

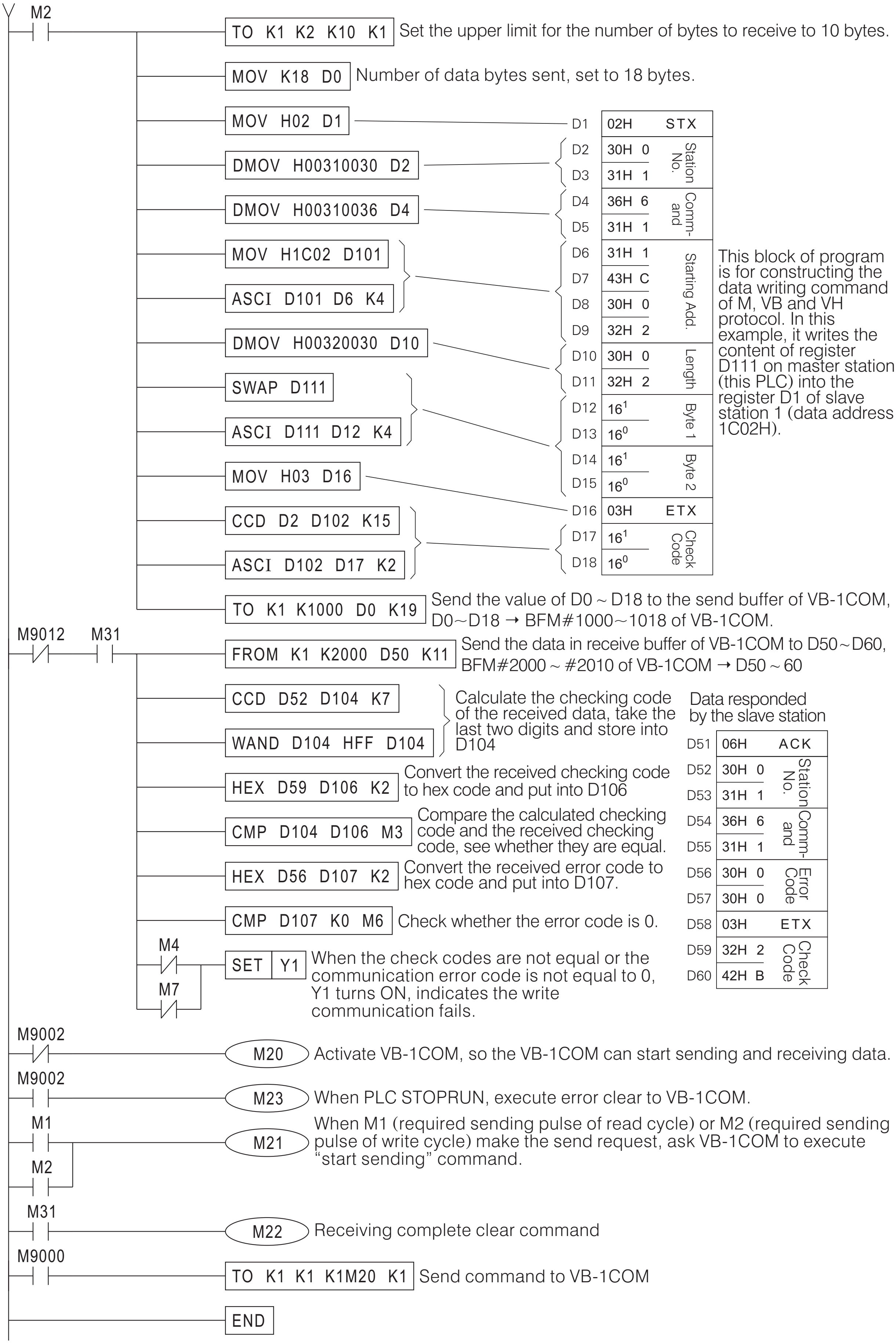
	1C03		1C02	
D1	16^1	16^0	16^1	16^0

Data responded by slave station 1

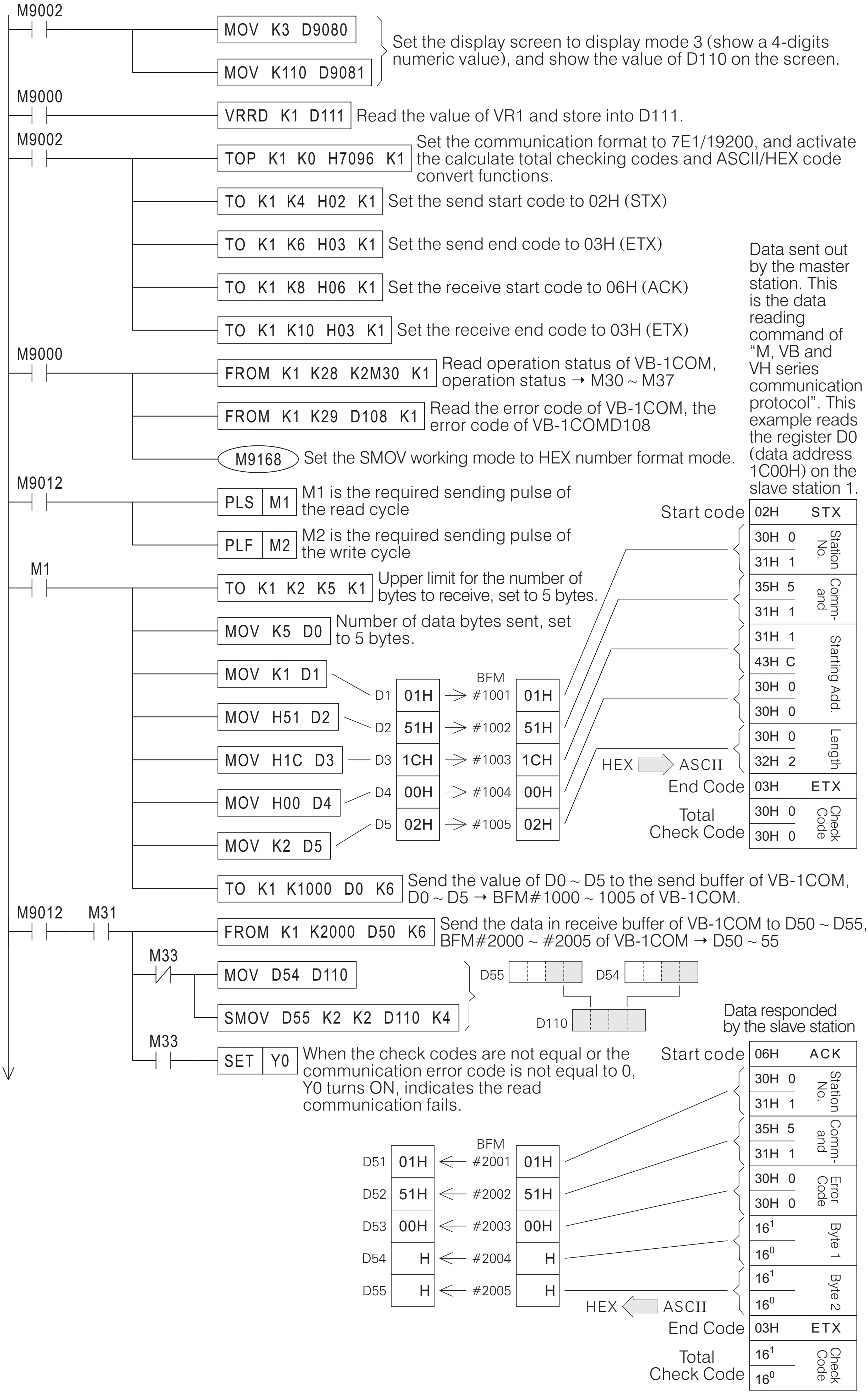
ACK	Station No.		Command		Error Code		ETX	Check Code	
	0	1	6	1	0	0		2	B
06H	30H	31H	36H	31H	30H	30H	03H	32H	42H

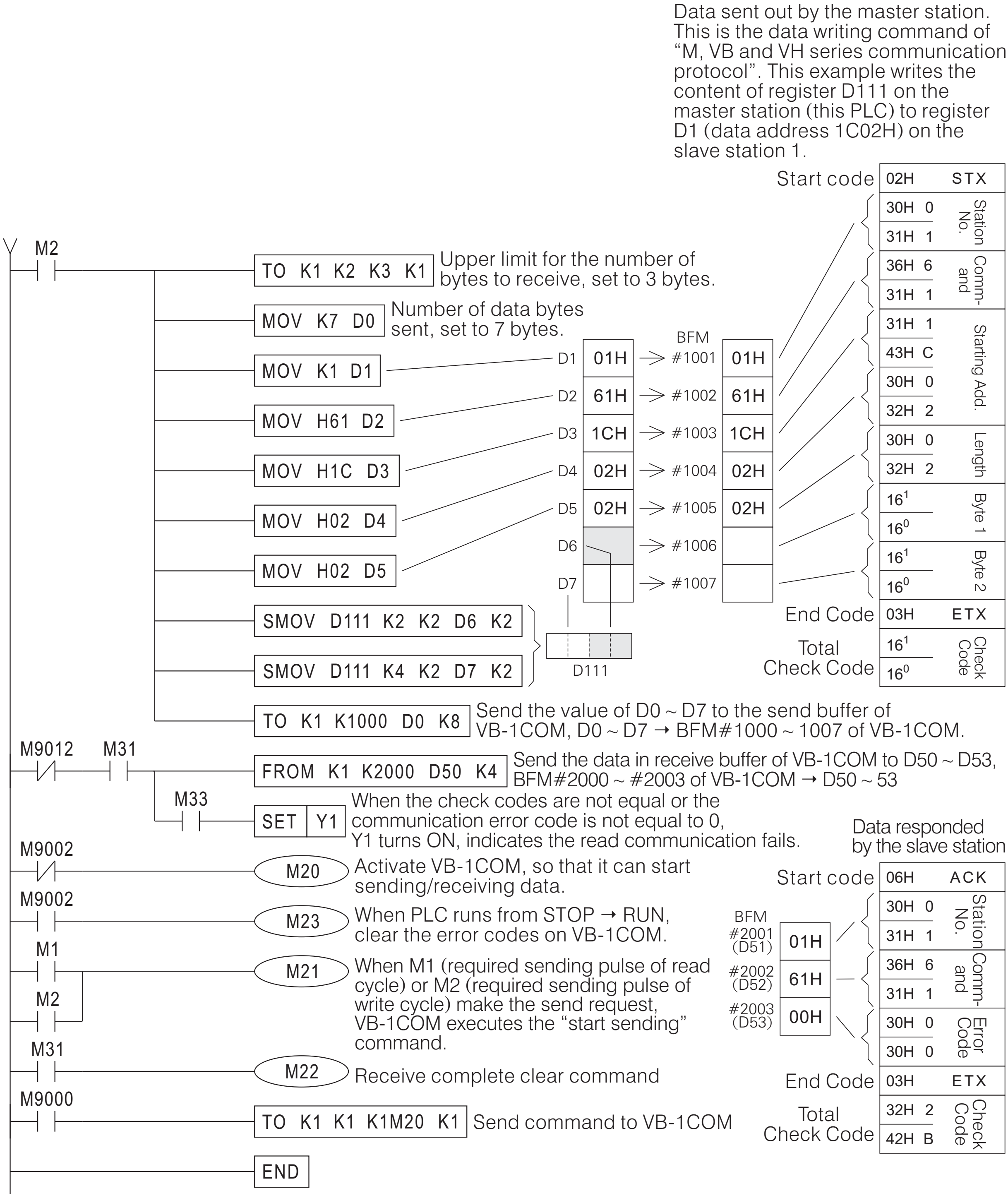
- Program example 1 of left PLC: without activating the auxiliary function



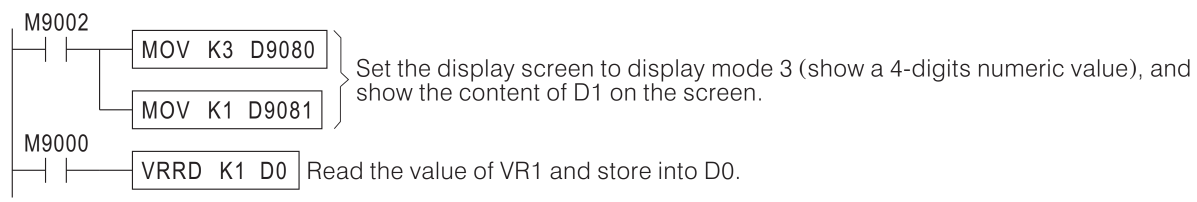

M2
TO K1 K2 K10 K1
Set the upper limit for the number of bytes to receive to 10 bytes.
MOV K18 D0
Number of data bytes sent, set to 18 bytes.
MOV H02 D1
DMOV H00310030 D2
DMOV H00310036 D4
MOV H1C02 D101
ASCI D101 D6 K4
DMOV H00320030 D10
SWAP D111
ASCI D111 D12 K4
MOV H03 D16
CCD D2 D102 K15
ASCI D102 D17 K2
D1 02H STX
D2 30H 0
D3 31H 1
Station No.
D4 36H 6
D5 31H 1
Comm-and
D6 31H 1
D7 43H C
D8 30H 0
D9 32H 2
Starting Add.
D10 30H 0
D11 32H 2
Length
D12 16^1
D13 16^0
Byte 1
D14 16^1
D15 16^0
Byte 2
D16 03H ETX
D17 16^1
D18 16^0
Check Code
This block of program is for constructing the data writing command of M, VB and VH protocol. In this example, it writes the content of register D111 on master station (this PLC) into the register D1 of slave station 1 (data address 1C02H).
TO K1 K1000 D0 K19
Send the value of D0 ~ D18 to the send buffer of VB-1COM, D0~D18 → BFM#1000~1018 of VB-1COM.
M9012 M31
FROM K1 K2000 D50 K11
Send the data in receive buffer of VB-1COM to D50~D60, BFM#2000 ~ #2010 of VB-1COM → D50 ~ 60
CCD D52 D104 K7
WAND D104 HFF D104
Calculate the checking code of the received data, take the last two digits and store into D104
Data responded by the slave station
D51 06H ACK
D52 30H 0
D53 31H 1
Station No.
D54 36H 6
D55 31H 1
Comm-and
D56 30H 0
D57 30H 0
Error Code
D58 03H ETX
D59 32H 2
D60 42H B
Check Code
HEX D59 D106 K2
Convert the received checking code to hex code and put into D106
CMP D104 D106 M3
Compare the calculated checking code and the received checking code, see whether they are equal.
HEX D56 D107 K2
Convert the received error code to hex code and put into D107.
CMP D107 K0 M6
Check whether the error code is 0.
M4
M7
SET Y1
When the check codes are not equal or the communication error code is not equal to 0, Y1 turns ON, indicates the write communication fails.
M9002
M20
Activate VB-1COM, so the VB-1COM can start sending and receiving data.
M9002
M23
When PLC STOPRUN, execute error clear to VB-1COM.
M1
M2
M21
When M1 (required sending pulse of read cycle) or M2 (required sending pulse of write cycle) make the send request, ask VB-1COM to execute "start sending" command.
M31
M22
Receiving complete clear command
M9000
TO K1 K1 K1M20 K1
Send command to VB-1COM
END

- Left PLC Program Example 2: activate auxiliary function





- Program of right side (slave station) PLC



MEMO

B-5 Communication Protocol of M, VB and VH Series

B-5-1 Communication Parameters

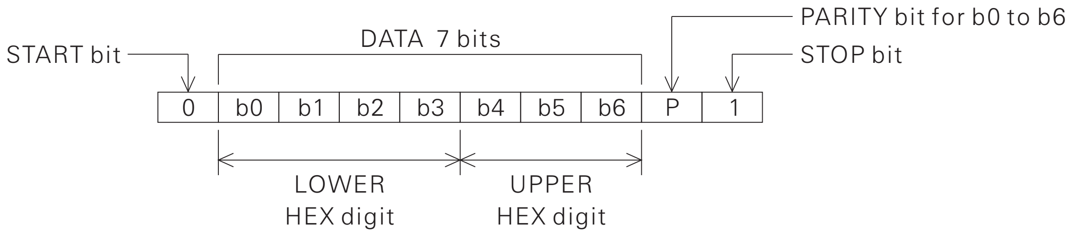
- Data length: 7 bits (ASCII)

  Parity: EVEN

  Stop bit: 1 bit

  Baud rate: the PLC built-in CP1 is fixed to 19200 bps.
  User can select any of 4800/9600/19200/38400 bps for CP2 by Ladder Master.
  CP3 is fixed to 19200 bps.

- Format of communication syntax



- This communication protocol use ASCII Code to transmit data, the table below lists the possible characters and the corresponding ASCII Codes.

Character	ASCII Code
STX	02H
ETX	03H
ACK	06H

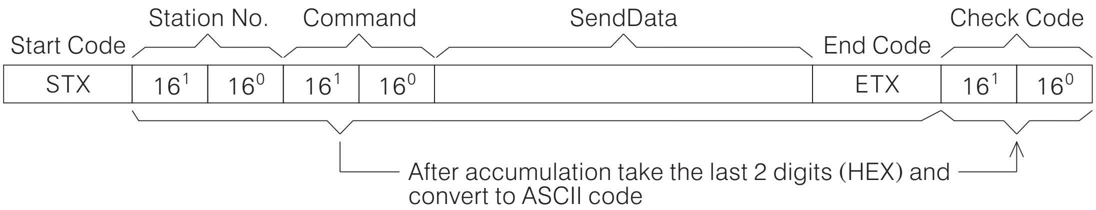
Character	ASCII Code
0	30H
1	31H
2	32H
3	33H
4	34H
5	35H
6	36H
7	37H

Character	ASCII Code
8	38H
9	39H
A	41H
B	42H
C	43H
D	44H
E	45H
F	46H

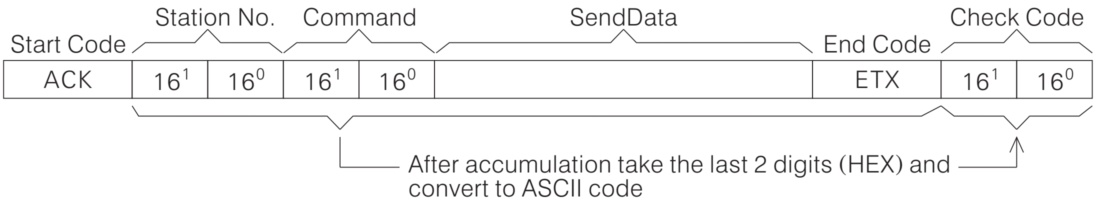
- Communication station number: CP1 is a stander build-in com port in a PLC, its default station number is 0.(In case need to change it, could use the Ladder Master to assign the station number between 0 ~ 255.)
  CP2 could use the Ladder Master to assign the station number between 0 ~ 255.
  CP3 is from the VB-CADP module, use two rotary switches on its left side to assign the station number between 00 ~ 99.

B-5-2 Communication Protocol Data Format

- The communication format to PLC



- The communication format to PLC



- Start code: starting character of data to transfer. The start code when send command to PLC is ASCII code STX (02H) and the start code when PLC send back data is ASCII code ACK (06H).
- Station Number: the identification number of the data transfer target. Every PLC in the communication circuit needs to have a station number. And when computer give communication command to PLC, it uses station number to identify which PLC is the target.
- Command: the computer command PLC to do the assigned tasks.

Command	Command code	Target Component	Introduction
Serial Data Read	51H	X、Y、M、S、T、C、D	Continuously read the bit component status or register value
Serial Data Write	61H	X、Y、M、S、T、C、D	Continuously write the bit component status or register value
Bit Component ON	70H	X、Y、M、S	Set the appointed component to ON
Bit Component OFF	71H	X、Y、M、S	Set the appointed component to OFF

- Data to Send: the content of the data to send. It may includes error code, data address, length of data to send, content of data to send, etc.
- End Code: the end bit of the data to send. The end code is ASCII code ETX (03H).
- Check Code: accumulate the data value from the station number until the end code, then take the last 2 digits (HEX) and convert to ASCII code as the checking code. Execute the same checking code processing operation at both the data sending side and the data receiving side, in order to ensure the transmit data is correct.

To PLC	STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
		0	0	5	1	0	0	8	1	0	2		F	4
	02H	30H	30H	35H	31H	30H	30H	38H	31H	30H	32H	03H	46H	34H

After accumulation take the last 2 digits (HEX) and convert to ASCII code

30H+30H+35H+31H+30H+30H+38H+31H+30H+32H+03H=1F4H

- Error Code: there will be an error code information in the data sent back by PLC to computer, and the table below lists the meaning of each error code.

Error Code	Details
00H	Communication no error
10H	ASCII converting error
11H	Communication SUM Check Error
12H	No such command
14H	Communication Error like STOP, Parity Error
28H	Data address exceeds range

B-5-3 Communication Instructions

- The table of component ID and the corresponding communication data addresses.

Component Name	Component ID	Data Address	Data Content							
			b7	b6	b5	b4	b3	b2	b1	b0
Input Relay X	X0～X7	0000	X7	X6	X5	X4	X3	X2	X1	X0
	≀	≀	≀							
	X770～X777	003F	X777	X776	X775	X774	X773	X772	X771	X770
Output Relay Y	Y0～Y7	0040	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
	≀	≀	≀							
	Y770～Y777	007F	Y777	Y776	Y775	Y774	Y773	Y772	Y771	Y770
Auxiliary Relay M	M0～M7	0080	M7	M6	M5	M4	M3	M2	M1	M0
	≀	≀	≀							
	M5112～M5119	02FF	M5119	M5118	M5117	M5116	M5115	M5114	M5113	M5112
Step Relay S	S0～S7	0300	S7	S6	S5	S4	S3	S2	S1	S0
	≀	≀	≀							
	S992～S999	037C	S999	S998	S997	S996	S995	S994	S993	S992
Timer Contact	T0～T7	0380	T7	T6	T5	T4	T3	T2	T1	T0
	≀	≀	≀							
	T248～T255	039F	T255	T254	T253	T252	T251	T250	T249	T248
Counter Contact	C0～C7	03A0	C7	C6	C5	C4	C3	C2	C1	C0
	≀	≀	≀							
	C248～C255	03BF	C255	C254	C253	C252	C251	C250	C249	C248
Special Relay M9000 ≀ M9255	M9000～M9007	03E0	M9007	M9006	M9005	M9004	M9003	M9002	M9001	M9000
	≀	≀	≀							
	M9248～M9255	03FF	M9255	M9254	M9253	M9252	M9251	M9250	M9249	M9248
Timer Coil	T0～T7	0780	T7	T6	T5	T4	T3	T2	T1	T0
	≀	≀	≀							
	T248～T255	079F	T255	T254	T253	T252	T251	T250	T249	T248
Counter Coil	C0～C7	07A0	C7	C6	C5	C4	C3	C2	C1	C0
	≀	≀	≀							
	C248～C255	07BF	C255	C254	C253	C252	C251	C250	C249	C248
Timer Current Value	T0	1400 1401	T0 MSB 16^1 16^0 16^1 16^0 LSB 1401 1400							
	≀	≀								
	T255	15FE 15FF								
Special register D9000 ≀ D9255	D9000	1600 1601	D9000 MSB 16^1 16^0 16^1 16^0 LSB 1601 1600							
	≀	≀								
	D9255	17FE 17FF								
C0 ≀ C199 Current Value	C0	1800 1801	C0 MSB 16^1 16^0 16^1 16^0 LSB 1801 1800							
	≀	≀								
	C199	198E 198F								
C200 ≀ C255 Current Value	C200	1A00 ≀ 1A03	C200 MSB 16^1 16^0 16^1 16^0 16^1 16^0 16^1 16^0 LSB 1A03 1A02 1A01 1A00							
	≀	≀								
	C255	1ADC ≀ 1ADF								
D0 ≀ D8191 Content Value	D0	1C00 1C01	D0 MSB 16^1 16^0 16^1 16^0 LSB 1C01 1C00							
	≀	≀								
	D8191	5BFE 5BFF								

- Command Number 51H: continuous data read command (can read 128 bytes at most)

To PLC	STX	Station No.		Command		Starting Add.				Length (Bytes)		ETX	Check Code	
		16^1	16^0	16^1	16^0	16^3	16^2	16^1	16^0	16^1	16^0		16^1	16^0

From PLC	ACK	Station No.		Command		Error Code		Byte 1 data		Byte 2 data		…	Last data Byte		ETX	Check Code	
		16^1	16^0	16^1	16^0	16^1	16^0	16^1	16^0	16^1	16^0		16^1	16^0		16^1	16^0

Example 1: read the status value of M8 ~ M23

Suppose that the status of M8 ~ M23 of the PLC are as below:

M23							M16	M15							M8
1	0	0	1	0	0	1	1	1	0	1	0	0	1	1	1
9				3				A				7			

To PLC	STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
		0	0	5	1	0	0	8	1	0	2		F	4
	02H	30H	30H	35H	31H	30H	30H	38H	31H	30H	32H	03H	46H	34H

From PLC	ACK	Station No.		Command		Error Code		Byte 1		Byte 2		ETX	Check Code	
		0	0	5	1	0	0	A	7	9	3		0	D
	06H	30H	30H	35H	31H	30H	30H	41H	37H	39H	33H	03H	30H	44H

Example 2: read the content value of D1, D2

Suppose the content value of D1 of the PLC is 1234H, and the content value of D2 is ABCDH.

To PLC	STX	Station No.		Command		Starting Add.				Length		ETX	Check Code	
		0	0	5	1	1	C	0	2	0	4		0	3
	02H	30H	30H	35H	31H	31H	43H	30H	32H	30H	34H	03H	30H	33H

From PLC	ACK	Station No.		Command		Error Code		Byte 1		Byte 2		Byte 3		Byte 4		ETX	Check Code	
		0	0	5	1	0	0	3	4	1	2	C	D	A	B		F	D
	06H	30H	30H	35H	31H	30H	30H	33H	34H	31H	32H	43H	44H	41H	42H	03H	46H	44H

- Command Number 61H: continuous data write command (can write 128 bytes at most)

To PLC

STX	Station No.		Command		Starting Add.				Length (Bytes)		Byte 1 data		Byte 2 data		...	Last data Byte		ETX	Check Code	
	16^1	16^0	16^1	16^0	16^3	16^2	16^1	16^0	16^1	16^0	16^1	16^0	16^1	16^0		16^1	16^0		16^1	16^0

From PLC

ACK	Station No.		Command		Error Code		ETX	Check Code	
	16^1	16^0	16^1	16^0	16^1	16^0		16^1	16^0

Example 1: write into Y30 ~ Y47

Suppose that the status of Y30 ~ Y47 of the PLC to be written are as below:

Y47							Y40	Y37							Y30
0	0	1	1	1	1	1	0	1	1	0	0	0	1	1	0
3				E				C				6			

To PLC

STX	Station No.		Command		Starting Add.				Length		Byte 1		Byte 2		ETX	Check Code	
	0	0	6	1	0	0	4	3	0	2	C	6	3	E		E	4
02H	30H	30H	36H	31H	30H	30H	34H	33H	30H	32H	43H	36H	33H	45H	03H	45H	34H

From PLC

ACK	Station No.		Command		Error Code		ETX	Check Code	
	0	0	6	1	0	0		2	A
06H	30H	30H	36H	31H	30H	30H	03H	32H	41H

Example 2: write A325H into the register D1 of the PLC

To PLC

STX	Station No.		Command		Starting Add.				Length		Byte 1		Byte 2		ETX	Check Code	
	0	0	6	1	1	C	0	2	0	2	2	5	A	3		D	D
02H	30H	30H	36H	31H	31H	43H	30H	32H	30H	32H	32H	35H	41H	33H	03H	44H	44H

From PLC

ACK	Station No.		Command		Error Code		ETX	Check Code	
	0	0	6	1	0	0		2	A
06H	30H	30H	36H	31H	30H	30H	03H	32H	41H

- Command Number 70H: bit component ON command
  Command Number 71H: bit component OFF command

To PLC	STX	Station No.		Command		Bit Component Address				ETX	Check Code	
		16^1	16^0	16^1	16^0	16^3	16^2	16^1	16^0		16^1	16^0

The bit component address consists of the data address and the big component position. Here use S20 as example to explain below:
Bit component is S20 (S)20÷8=2…4 。

The component position of S20 is 4
The data address of S0 is 0300H
The data address of S20 is 0300H + 2H = 0302H

Data Address / Component Position

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

1				8				1				4			
0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0

Bit Component Address

From PLC	ACK	Station No.		Command		Error Code		ETX	Check Code	
		16^1	16^0	16^1	16^0	16^1	16^0		16^1	16^0

Example 1: set M10 to ON

To PLC	STX	Station No.		Command		Bit Component Address				ETX	Check Code	
		0	0	7	0	0	4	0	A		9	F
	02H	30H	30H	37H	30H	30H	34H	30H	41H	03H	39H	46H

Calculate the bit component address of M10:
(M) 10 / 8 = 1…2
The data address of M0 is 0080H, and the data address of M10 is 0080H + 1H = 0081H
0080H+1H=0081H

0				0				8				1				2		
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0

0				4				0				A			
0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0

From PLC	ACK	Station No.		Command		Error Code		ETX	Check Code	
		0	0	7	0	0	0		2	A
	06H	30H	30H	37H	30H	30H	30H	03H	32H	41H

Example 2: set M1000 to OFF

To PLC	STX	Station No.		Command		Bit Component Address				ETX	Check Code	
		0	0	7	1	0	4	6	4		9	9
	02H	30H	30H	37H	31H	30H	34H	36H	34H	03H	39H	39H

Calculate the bit component address of M100:
(M) 10 / 8 = 12…4
The data address of M0 is 0080H, and the data address of M10 is 0080H + CH = 008CH

0				0				8				C				4		
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0

0				4				6				4			
0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0

From PLC	ACK	Station No.		Command		Error Code		ETX	Check Code	
		0	0	7	1	0	0		2	B
	06H	30H	30H	37H	31H	30H	30H	03H	32H	42H

MEMO

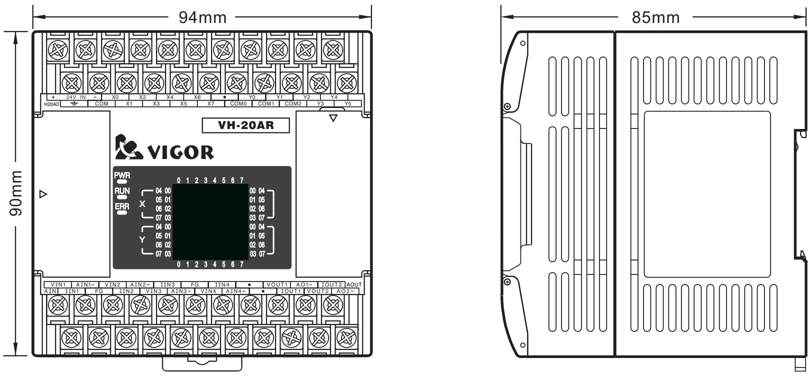
MEMO

C. VH-20AR Main Unit User Manual

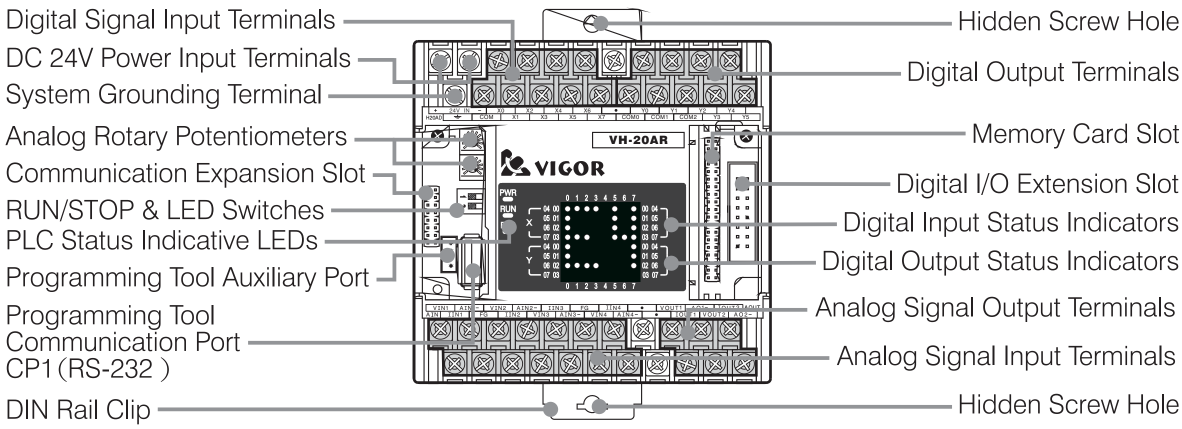
The VH-20AR Main Unit is a new model of the VH series PLC, it is not only supporting all functions of the original VH series PLC but also providing 4 channel analog inputs and 2 channel analog outputs. Which can extend the application of the VH series at analog controls.

C-1 Dimension and Component Designation

C-1-1 Dimension



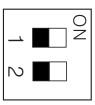
C-1-2 Component Designation



- The Programming Tool Communication Port (CP1, a USB A-type outlet) is using the RS-232 interface, that can not be directly connected with any equipment's USB port.
- Please use a VBUSB-200 adapter to connect between a PLC's Programming Tool Communication Port (CP1) and computer's USB port.
- Please use a MWPC-200 cable to connect between a PLC's Programming Tool Communication Port (CP1) and computer's RS-232 (Serial) port.
- Please use the BT-232 Bluetooth adapter at PLC's Programming Tool Communication Port (CP1) then by the wireless Bluetooth to connect with a computer or intelligent cellular phone.
- Usually, the Programming Tool Auxiliary Port (JST 4P outlet) is for connecting with a HMI or SCADA . The circuit of Programming Tool Auxiliary Port and Programming Tool Communication Port are parallel, either one of them can be use at same time.
- PLC Status Indicative LED

LED	Status	Comment
PWR (GREEN)	ON	Power in Supply
	OFF	Power Failure
RUN (GREEN)	ON	RUN
	OFF	STOP
ERR (RED)	ON	PLC System Error (Stop Running)
	Blinking	Abnormal State (Stop Running)
	OFF	Normal

- RUN/STOP & Indicating Section Switches



Number	Function	OFF	ON
1	RUN/STOP Switch	STOP	RUN
2	I/O Indicating Range Switch	X0～X37, Y0～Y37	X40～X77,Y40～Y77

C-2 VH-20AR Specification

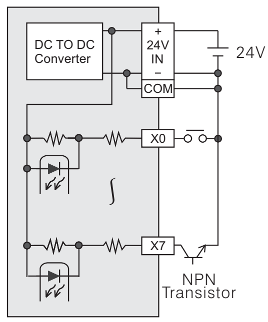
C-2-1 Performance Specification

Item			Specifications
Operation Control Method			Cyclic Operation by Stored Program
Programming Language Method			Electric Ladder Diagram + SFC
I/O Control Method			Batch Processing
Operation Processing Time	Basic Instruction		0.375 ~ 12.56 μs
	Applied Instruction		Several μs ~ Several 100 μs
Number of Instructions	Basic Instructions		27 (including: LDP,LDF, ANDP, ANDF, ORP, ORF and INV)
	Stepladder Instructions		2
	Applied Instructions		81
Memory Capacity (Flash ROM)	Program Capacity		Built-in 4 K Steps Flash ROM
	Comment Capacity		2730 comments (16 words for each comment)
	Program Comment Capacity		20,000 word
Max. Input / Output Points			128 points: X0 ~ X77, Y0 ~ Y77
Internal Relay	Auxiliary Relay (M)	General	384 points: M0 ~ M383
		Latched	128 points: M384 ~ M511
		Special	256 points: M9000 ~ M9255
	State Relay (S)	Initial	10 points: S0 ~ S9 (Latched)
		Latched	118 points: S10 ~ S127
Timer (T)		100ms.	63 points: T0 ~ T62 (Timer range: 0.1 ~ 3276.7 sec.)
		10ms.	31 points: T32 ~ T62 (When M9028 = "ON", Timer range: 0.01 ~ 327.67 sec.)
		1 ms.	1 points: T63 (Timer range: 0.001 ~ 32.767 sec.)
Counter (C)	16-bit Up	General	16 points: C0 ~ C15
		Latched	16 points: C16 ~ C31
High Speed Counter (C)	32-bit Bi-directional, Latched	1-phase Counter	11 points: C235 ~ C245 (Signal Frequency: 10 KHz Max.)
		2-phase Counter	5 points: C246 ~ C250 (Signal Frequency: 10 KHz Max.)
		A/B Phase Counter	4 points: C251 ~ C254 (Signal Frequency: 5 KHz Max.)
Data Register (D)		General	128 points: D0 ~ D127
		Latched	128 points: D128 ~ D255
		Special	256 points: D9000 ~ D9255
		Index	16 points: V0 ~ V7, Z0 ~ Z7
Pointer		Call Pointer (P)	64 points: P0 ~ P63
		Interrupt Pointer (I)	15 points: 6 points for external interrupt, 3 points for timer interrupt, and 6 points for counter interrupt
		Nest Pointer (N)	8 points: N0 ~ N7
Range of Constants	Decimal (K)	16 Bits	-32768 ~ 32767
		32 Bits	-2147483648 ~ 2147483647
	Hexadecimal (H)	16 Bits	0H ~ FFFFH
		32 Bits	0H ~ FFFFFFFFH
Pulse Output			1 point; Max. 7 KHz
Programming Device Link Interface CP1			RS-232C for directly connect to a PC, HMI or MODEM; with the BT-232 via Bluetooth wireless to connect to a PC or cellular phone
Communication Link Interface CP2 (Optional)			RS-232C, RS-422/485 or Ethernet
Communication Link Interface CP3 (Optional)			RS-485, for direct connect with a computer HMI
Real Time Clock (Optional)			To indicates year, month, day, hour, min., sec. and week
Error Code Display Function			Displays 109 error code (01~99 or E0~E9)
Analog Potentiometer			2 Analog Rotary Potentiometers, for values input (0 ~ 255 or 0 ~ 10)
Main Unit Built-in I/O		Digital Input	8 Points, X0 ~ X7
		Digital Output	6 Points, Y0 ~ Y5
		Analog Input	4 Points, 12 bit resolution, ±10V / 4 ~ 20mA / ±20mA
		Analog Output	2 Points, 12 bit resolution, ±10V / 4 ~ 20mA / ±20mA

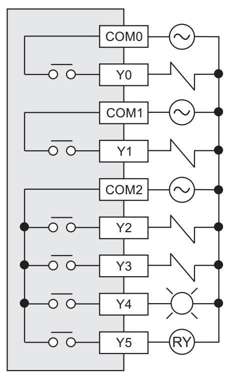
C-2-2 Power Specification

Item	Specifications
Power Input Require	DC24V, +20% / −15%
Input Frequency	—
Momentary Power Failure	Keep working at least 1 ms.
Power Fuse	250V; 0.5A
Power Consumption	5W (Main Unit Only)
Power Unit Output Current (Inner)	DC5V; 400mA
	DC12V; 530mA

C-2-3 Digital Input Specification

Item	Specifications
Power Input Require	DC24V, 15%
Input Signal Circuit	7mA / DC24V
Input ON Circuit	Above 3.5 mA
Input OFF Circuit	Below 1.7 mA
Input Resistance	3.3 kΩ approximately
Input Response Time	10 ms. approximately (X0 ~ X7 are variable, can be set between 0 ~ 15 ms.)
Input Signal Type	Dry Contact or NPN open collector transistor
Isolation Method	Photocoupler Isolation
Circuit Diagram	

C-2-4 Digital Output Specification

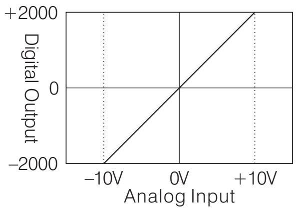
Item		Specifications
Output Type		Relay Output
Switched Voltages		≤ AC 250V / DC 30V
Rated Current	Resistive Load	2 A / point
	Inductive Load	80VA
	Lamp Load	100W
Open Circuit Leakage		—
Response Time		10 ms. approximately
Isolation Method		Mechanic Isolation (Relay)
Circuit Diagram		

C-2-5 Analog Input Specification

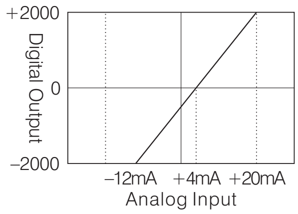
Item	Voltage Input	Current Input
	Voltage or Current Signal Inputs are Designated by D9090 and Different Terminals	
Analog Input Range	–10V ~ +10V	4 ~ 20mA / –20mA ~ +20mA
Digital Output Range	–2000 ~ +2000	0 ~ 2000 / –2000 ~ +2000
Input Resistance	200KΩ	250Ω
Resolution	5mV	20μA
Overall Accuracy	±1% (Max.)	
Conversion Speed	Data refresh at every Scan Time	
Isolation Method	Magnetic-coupler isolation between PLC and inputs; no isolation between analog input channels	
Max. Sustainable Input Range	±15V	±32mA

Curve diagram of A/D conversion characteristics (Designated by D9090)

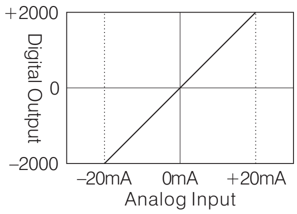
Mode **0**
(Voltage Input: –10V ~ +10V)



Mode **1**
(Circuit Input: +4mA ~ +20mA)



Mode **2**
(Circuit Input: –20mA ~ +20mA)

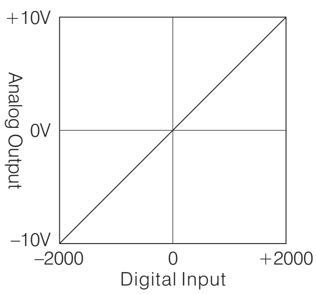


C-2-6 Analog Output Specification

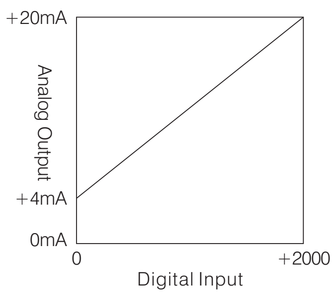
Item	Voltage Output	Current Output
	Voltage or Current Signal Outputs are Designated by D9095 and Different Terminals	
Analog Output Range	–10V ~ +10V	4 ~ 20mA / –20mA ~ +20mA
Digital Input Range	–2000 ~ +2000	0 ~ 2000 / –2000 ~ +2000
External Loading Resistance	500Ω ~ 1MΩ	Under 500Ω
Resolution	5mV	10μA
Overall Accuracy	±2% (Max.)	
Conversion Speed	Data refresh at every Scan Time	
Isolation Method	Magnetic-coupler isolation between PLC and outputs; no isolation between analog output channels	

Curve diagram of D/A conversion characteristics (Designated by D9095)

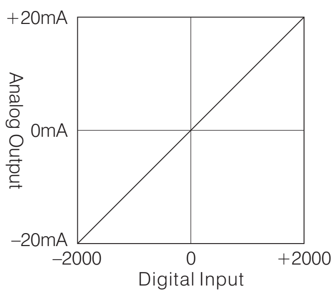
Mode **0**
(Voltage Output: –10V ~ +10V)



Mode **1**
(Circuit Output: +4mA ~ +20mA)



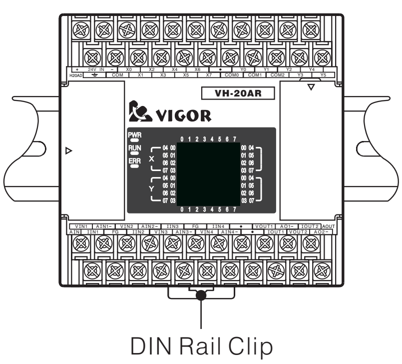
Mode **2**
(Circuit Output: –20mA ~ +20mA)



C-3 Installation

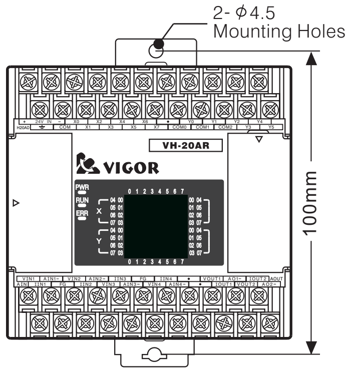
C-3-1 Installation Guides

- DIN Rail Installation



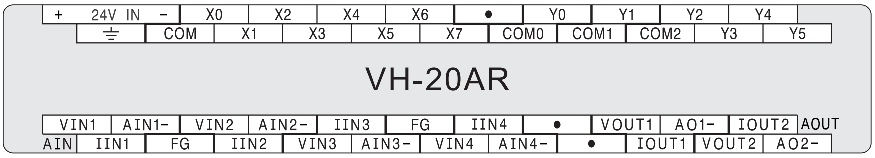
Install the PLC on a 35mm standard DIN rail; Pull the clip down can remove.

- Direct Screws Installation

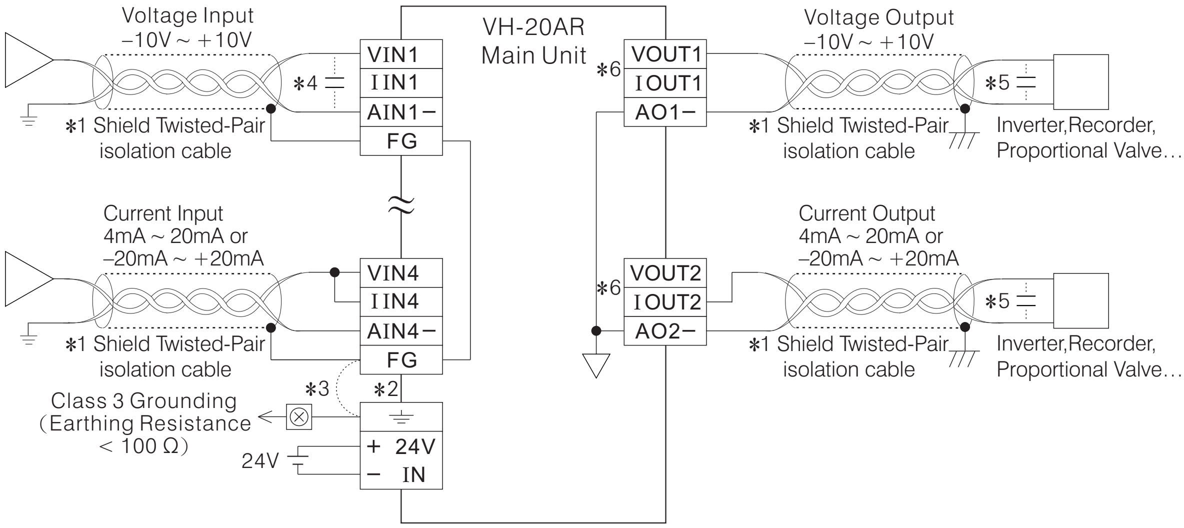


Pull out the hidden mounting holes (100mm distance) from the top and bottom of the PLC and install with screws.

C-3-2 Terminal Layouts



C-3-3 External Wiring



∗1 : Please use the Shield Twisted-Pair isolation cable for each analog input and output channel, and keep the cable away from the electromagnetic interference source (ex. power lines or any other lines which may induce electrical noise). Apply 1-point grounding at the load side of the output cable (Class 3 Grounding: Earthing Resistance < 100 Ω).

∗2 : Connect the ⏚ terminal to the grounding point and use the Class 3 Grounding for the system or connect it to the rack of distribution board.

∗3 : If there is excessive electrical noise, connect the FG frame ground terminal with the ⏚ terminal.

∗4 : If a voltage ripple occurs during input or there is electrically induced noise on the external wiring, please parallel connection a smoothing capacitor (0.1 ~ 0.47 μF, 25 V) between the input terminals.

∗5 : If electrical noise or a voltage ripple exists at the output signal to loader, plase parallel connection a smoothing capacitor (0.1 ~ 0.47 μF, 25 V) between the input terminals of loader.

∗6 : Use both (voltage and current) outputs from a channel is not allow.

C-4 Operating Instruction

Special data registers list for analog functions. They are not latched registers.

Register #	Special data registers list for analog functions. They are not latched registers.	
D9090	To organize the input modes of AIN1 ～ AIN4	
D9091	Averaged input value from AIN1	• Data values refresh at every Scan Time. • The contain values of D9091 ~ D9094 are averaged of 8 sampling times.
D9092	Averaged input value from AIN2	
D9093	Averaged input value from AIN3	
D9094	Averaged input value from AIN4	
D9095	To organize the output modes of AO1 and AO2	
D9096	Digital value for AO1 output	• Analog outputs refresh at every Scan Time. • The digital value of analog outputs will be reset when the PLC "STOP"
D9097	Digital value for AO2 output	

- For 4 analog value inputs, the value of D9090 switches the modes between voltage or current analog input on each channel. The D9090 uses a format of 4-digit hexadecimal number.
  The first hexadecimal digit will be the command for AIN1, and the second digit is for AIN2, and so forth.
  The numeric value of each digit respectively represent the following definitions:
  If the value of digit = 0 : Sets the channel to voltage input mode (–10 V ~ +10 V).
  If the value of digit = 1 : Sets the channel to current input mode (+4 mA ~ +20 mA).
  If the value of digit = 2 : Sets the channel to current input mode (–20 mA ~ +20 mA).
  If the value of digit = 3 : Disables the channel.

  Example: Let the D9090 = H3210
  AIN1 = 0 : Voltage output (–10 V ~ +10 V)
  AIN2 = 1 : Current output (+4 mA ~ +20 mA)
  AIN3 = 2 : Current output (–20 mA ~ +20 mA)
  AIN4 = 3 : Disabled.

b15 b0

Digit #4	Digit #3	Digit #2	Digit #1	D9090
AIN4	AIN3	AIN2	AIN1	

Input modes assign

- For 2 analog outputs, the value of D9095 switches the modes between voltage or current analog output on each channel. The D9095 uses a format of 2-digit hexadecimal number.
  The first hexadecimal digit will be the command for AO1, and the second digit is for AO2.
  The numeric value of each digit respectively represent the following definitions:
  If the value of digit = 0 : Sets the channel to voltage output mode (–10 V ~ +10 V).
  If the value of digit = 1 : Sets the channel to current output mode (+4 mA ~ +20 mA).
  If the value of digit = 2 : Sets the channel to current output mode (–20 mA ~ +20 mA).
  If the value of digit = 3 : Disables the channel.

  Example: Let the D9095 = H10
  AO1 = 0 : Voltage output (–10 V ~ +10 V)
  AO2 = 1 : Current output (+4 mA ~ +20 mA)

b15 b0

Digit #4	Digit #3	Digit #2	Digit #1	D9095
—	—	AO2	AO1	

Output modes assign

- Example Program

M9002
MOV H0210 D9090 (H0210)→D9090 to set the input modes
MOV H10 D9095 (H10)→D9095 to set the output modes
M9000
MOV D9091 D100 Moves the average value of AIN1 (by VIN1 & AIN1–) to D100
MOV D9092 D101 Moves the average value of AIN2 (by IIN2 & AIN2–) to D101
MOV D9093 D102 Moves the average value of AIN3 (by IIN3 & AIN3–) to D102
MOV D9094 D103 Moves the average value of AIN4 (by VIN4 & AIN4–) to D104
M9000
MOV D110 D9096 Moves the digital value of D110 to D9096 then convert to analog signal and output to AO1 (by VOUT1 & AO1–)
MOV D111 D9097 Moves the digital value of D111 to D9097 then convert to analog signal and output to AO2 (by IOUT2 & AO2–)

MEMO

MEMO

Z. Add-on notes:

Z-1 Newly added instructions

FNC No.	Instruction Title			Function	Applicable PLC Type		
	D		P		M	VB	VH
88		PID		PID control loop		V1.31	
92		TPID		Temperature PID Control		V1.70	
149		MBUS		MODBUS Communication		V1.31	V0.22
169		HOUR		Hour Meter		V1.30	
250	D	SCL	P	Scaling (Translated by Coordinate)		V1.70	
251	D	SCL2	P	Scaling II (Translated by Coordinate)		V1.70	
151	D	DVIT		One-speed Interrupt Constant Quantity Feed		VB1	
153	D	LIR		Relatively Linear Interpolation		VB1	
154	D	LIA		Absolutely Linear Interpolation		VB1	
188		CRC	P	Cyclic Redundancy Check - 16		V1.72	

	FNC 88 PID		PID (S1) (S2) (S3) (D)	PID control loop	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					
S2											○					
S3											○					
D											○					

- S3 occupies 25 consecutive registers

```
         (S1)(S2) (S3)  (D)
 X0
-| |----[PID D0 D1 D100 D200]
```

S1 : Set point value (SV)
S2 : Measured current value (PV)
S3 : The initiatory ID number of the parameters
D : Output value in destination device (MV)

- This instruction takes a current value from (S2) and compares it to a predefined set value (S1), then uses the parameters (initiated with (S3)) to process the PID operation. The calculate result will be stored to destination register (D) .
- When X0= "ON", this instruction starts to perform; When X0= "OFF", this process stops but the content value of D200 will be kept as the current value before X0= "OFF".
- The PID instruction's parameters are headed from (S3) , which require occupy 25 consecutive registers.
- When the control parametesr (S1) or (S3) +3 ~ (S3) +6 of setting values are changed, can rerun the PID instruction for the instant response of output value (D) .
- There's no limitation on the times used of the PID instruction.
- This instruction provided with the "Auto-tuning" function, it can help users to decide three of parameters in the PID instruction. (Please refer to follow pages.)
- Ecause the PID instruction uses the PLC's program Scan Time to accumulate the sampling time, to plan the program must pay attention on two following points:
  ① Even though, this instruction is allow to use it in a subroutine, interrupted subroutine, step ladder chart or conditional jump instructions, But at some of the PID instruction's processing duration, must make sure at every Scan Time of the program, it has been processed this instruction once. If this instruction has been processed more than once or had not been executed, it will cause some estimate error on the sampling time.
  ② When the sampling time is shorter than a Scan Time, it would make a PID process error. Then the PLC automatically sets the "sampling time = Scan Time" to execute the PID process.
- All the parameters must finished the settings before the PID instruction executes.

The Equations of the PID Instruction

This instruction is according to the differential of speed, to operation the PID instruction, the equations are shown in the table below:

Direction of Operation	The Equations of the PID Instruction
"Forward" $PV_{nf} > SV$	$\triangle MV = K_P\ \{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} EV_n + D_n \}$ $EV_n = PV_{nf} - SV$ $PV_{nf} = \alpha\, PV_{nf-1} + (1-\alpha)PV_n$ $D_n = \frac{T_D}{T_S + K_D \cdot T_D}(-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \triangle MV$
"Reverse" $SV > PV_{nf}$	$\triangle MV = K_P\ \{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} EV_n + D_n \}$ $EV_n = SV - PV_{nf}$ $PV_{nf} = \alpha\, PV_{nf-1} + (1-\alpha)PV_n$ $D_n = \frac{T_D}{T_S + K_D \cdot T_D}(2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \Sigma \triangle MV$

EV_n : The current error value
EV_{n-1} : The previous error value
SV : The set point value (S_1)
PV_n : The current process value (S_2)
PV_{nf} : The calculated process value
PV_{nf-1} : The previous process value
PV_{nf-2} : The second previous process value
$\triangle MV$: The change in the output manipulation values
MV_n : The current output manipulation value (D)

D_n : The derivation value
D_{n-1} : The previous derivation value
K_P : The proportion constant
α : The constant of input filter
T_S : The sampling time
T_I : The integral time constant
T_D : The time derivative constant
K_D : The derivative filter constant

• The description of parameters (S3 ~ S3 +24)

Para-meter	Parameter Name/Function	Description		Setting range
S_3	Sampling time (Ts)	The time interval should longer than the Scan Time and the current Process Value of the system		1 ~ 32767mS
S_3+1	Direction of action-reaction and alarm control	b0	0: "Forward" operation	—
			1: "Reverse" operation	
		b1	0: Process Value (PV_{nf}) alarm disable	
			1: Process Value (PV_{nf}) alarm enable	
		b2	0: Output Value (MV) alarm disable	
			1: Output Value (MV) alarm enable	
		b3	Reserved	
		b4	0: Disable the Auto-tuning	
			1: Enable the Auto-tuning, it will reverting to 0 after the Auto-tuning is finished	
		b5	0: Disable the limit of the output range	
			1: Enable the limit of the output range	
		b6 ~ b15	Reserved	
S_3+2	Input filter (α)	Alters the effect of the input filter to smooth the changes of measured current value		0 ~ 99%
S_3+3	Proportional gain (K_P)	This is the P (Proportional) part of the PID loop		1 ~ 32767%
S_3+4	Integral time constant (T_I)	This is the I (Integral) part of the PID loop, (this parameter disables the I effect if it is set to "0")		(0 ~ 32767) ×100mS
S_3+5	Derivative gain (K_D)	This is a factor used to align the derivative output in a know proportion to the change in the Process Value (PV_{nf})		0 ~ 100%
S_3+6	Derivative time constant (T_D)	This is the D (Derivative) part of the PID loop, (this parameter disables the D effect if it is set to "0")		(0 ~ 32767) ×10mS
S_3+7 ~ S_3+19	Working space	Reserved for the internal processing of the PID instruction		—
S_3+20	Process Value (PV_{nf}) changed alarm (+)	Maximum limit of positive change (upper limit); Active when S_3+1's b1= "ON"(1)		0 ~ 32767
S_3+21	Process Value (PV_{nf}) changed alarm (–)	Maximum limit of negative change (lower limit); Active when S_3+1's b1= "ON"(1)		
S_3+22	Output Value (MV) changing alarm (+)	Maximum limit of positive change (upper limit); Active when S_3+1's b2="ON"(1)		0 ~ 32767
	The range limit of Output Value (MV) change (+)	The range limit of the Output Value (MV) maximum positive change (upper limit); Active when S_3+1's b5= "ON"(1)		– 32768 ~ 32767
S_3+23	Output Value (MV) changing alarm (–)	Maximum limit of negative change (lower limit); Active when S_3+1's b2= "ON"(1)		0 ~ 32767
	The range limit of Output Value (MV) change (–)	The range limit of the Output Value (MV) maximum negative change (lower limit); Active when S_3+1's b5= "ON"(1)		– 32768 ~ 32767
S_3+24	Alarm flags (for read only)	b0	High limit exceeded in Process Value (PV_{nf})	—
		b1	Below low limit for the Process Value (PV_{nf})	
		b2	Excessive positive change in Output Value (MV)	
		b3	Excessive negative change in Output Value (MV)	

• S3 +1's b2 and b5 should not be active at the same time.

• When any one of the S3 +1's b1, b2 or b5 is "ON", the parameters of the PID instruction of S3 will occupy S3 ~ S3 +24 total 25 consecutive registers.

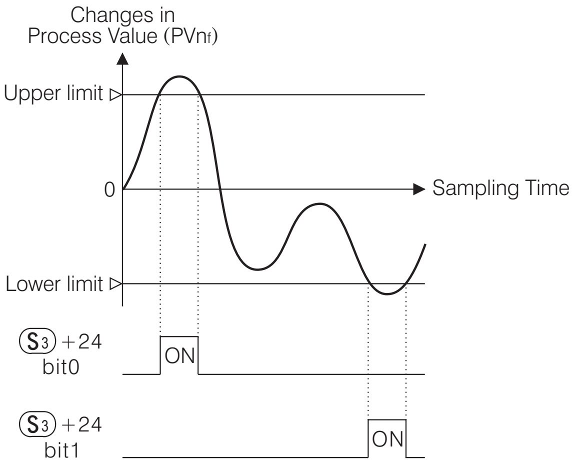
• When all of the S3 +1's b1, b2 and b5 are "OFF", the parameters of the PID instruction of S3 will occupy S3 ~ S3 +19 total 20 consecutive registers.

The Description of “Forward” and “Reverse” Operation

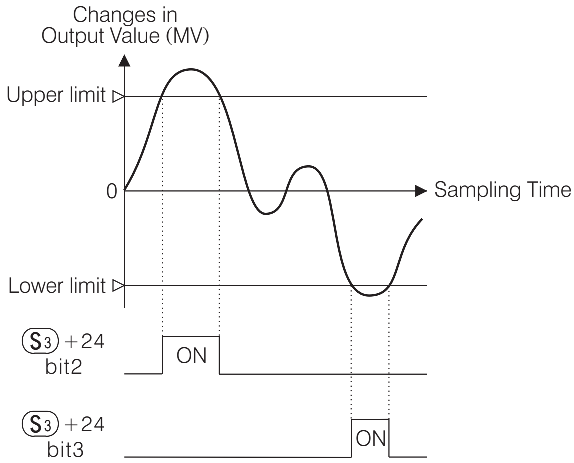
- If the parameter of S3 +1's b0=“OFF” then the PID instruction will process the forward operation;
  If the parameter of S3 +1's b0=“ON” then the PID instruction will process the reverse operation.
- When the calculated Process Value (PVnf) > the Set Point Value (SV), it will generate a positive deviation then the change to increase the effect is called forward operation.
  ex. A cooling air conditioning system: before the system turns on, usually the indoor temperature is higher than the set point value. (PVnf) > (SV), this is a typical forward operation control sample.
- When the calculated Process Value (PVnf) < the Set Point Value (SV), it will generate a negative deviation and increase the control effect is called “Reverse” operation.
  ex. An oven: before the heater of the oven turns on, usually the temperature of the oven is lower than the set point value. (PVnf) < (SV), this is a typical “Reverse” operation control sample.

The Description of Process Value (PVnf) Changed Alarm And Output Value (MV) Changing Alarm Functions

- If the S3 +1’s b1= “ON”, PID instruction provides the Process Value (PVnf) changed alarm. The parameters setting of the Process Value’s changed alarm are stored in S3 +20 and S3 +21 then the results will put in S3 +24’s b0 and b1. The content of S3 +21 is used as a negative value.
- If the S3 +1’s b2= “ON”, PID instruction provides the Output Value (MV) changing alarm. The parameters setting of the Output Value's changing alarm are stored in S3 +22 and S3 +23 then the results will put in S3 +24’s b2 and b3. The content of S3 +23 is used as a negative value.
- The definition of the change in Manipulation Values: Change = (Current value) – (Previous current value)
- The diagram of Process Value (PVnf) change:

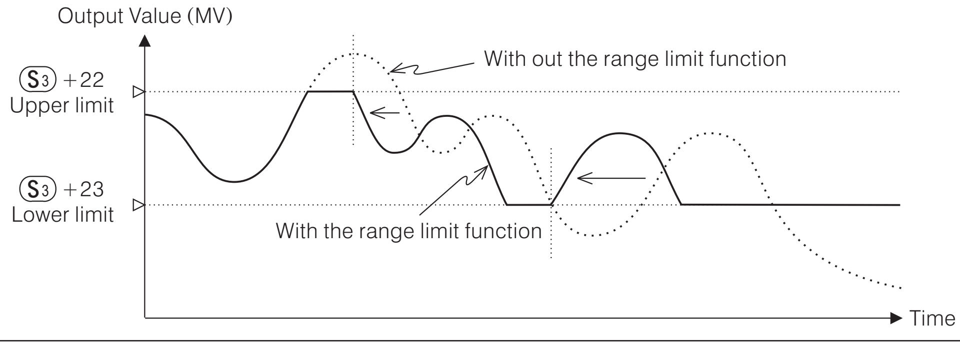


- The diagram of Output Value (MV) change:



The Description of Process Value (PVnf) Changed Alarm And Output Value (MV) Changing Alarm Functions

- If the parameter of S3 +1’s b5= “ON”, the PID instruction provides the range limit function of Output Value (MV) changing. The parameters setting of the Output Value's changing limits are store in S3 +22 and S3 +23.
- As a result both (limit and alarm) of the functions are occupy the same parameter registers S3 +22 and S3 +23. So, only one of the functions can be selected, the parameters in S3 +1’s b2 and b5 should not be “ON” at the same time.
- This function is very useful for limit the raise of the PID derivative value.
- The diagram of the range limit function of Output Value (MV) changing:



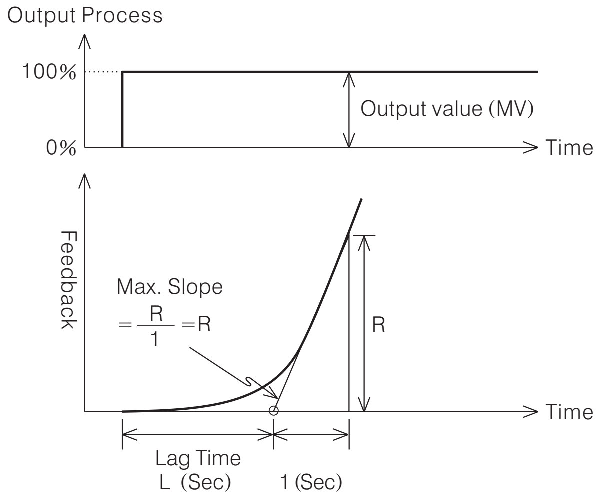
The Error Information of the PID Instruction

- If a setting value of parameter is not correct or the operation of a PID instruction occurs error, the special coil M9067 will be turned "ON". And the special register D9067 will store the error code.

Error Code	Error Occurrence	Treatment
6730	The setting value of Sampling Time (Ts) is beyond the range (Ts <1)	The PID instruction stops operation
6732	The setting value of Input Filter (α) is beyond the range ($\alpha <0$ or $\alpha >=0$)	
6733	The setting value of Proportion Constant (K_P) is beyond the range ($K_P<1$)	
6734	The setting value of Integral Time constant (T_I) is beyond the range ($T_I<0$)	
6735	The setting value of Derivative Filter Constant (K_D) is beyond the range ($K_D<0$ or $K_D>100$)	
6736	The setting value of Time Derivative Constant (T_D) is beyond the range ($T_D<0$)	
6740	The Sampling Time <= The Scan Time of PLC	The PID instruction keeps operation
6742	The variance of current Process Value is too large ($\triangle PV < -32768$ or $\triangle PV>32767$)	
6743	The variance of current Error Value is too large ($\triangle EV < -32768$ or $\triangle EV > 32767$)	
6744	The calculating value of Integral process exceeds –32768 ~ 32767	
6745	The value of Proportion Constant (K_P) is too large, it cause the calculating value of proportion which exceeds the range	
6746	The calculating value of Derivative process exceeds –32768 ~ 32767	
6747	The calculating result value of the PID instruction which exceeds –32768 ~ 32767	

The Method to Get The Parameters of a PID Instruction

- For a better control result of a PID instruction, we should get the correct parameters of the PID operation. It means we need to find the apropos values of Proportion Constant (K_P), Integral Time constant (T_I) and Time Derivative Constant (T_D).
- To get those three parameters, we have many different ways, usually the method of Process/Feedback Loop will be used. The following is the reference.
- The method of Process/ Feedback Loop gets the parameters is through step by step to control the system output between 0 ~ 100%. And then, observes the variation between processes and feedbacks, by those dynamic characteristics gets the parameters of PID.



Use the curve to get the PID's parameters

Control Method	Proportion Constant K_P(%)	Integral Time Constant T_I (× 100ms)	Time Derivative Constant T_D (× 10 ms)
P	$\frac{1}{RL}$ × Output value (MV)	—	—
PI	$\frac{0.9}{RL}$ × Output value (MV)	33L	—
PID	$\frac{1.2}{RL}$ × Output value (MV)	20L	50L

Auto-tuning Function

- The VB series provided the Auto-tuning function which can uses some PID correlative parameters from user (such as: the direction of action S3 +1, Sampling Time Ts, constant of Input Filter (α), Derivative Filter Constant K_D and Set Point Value S1) then via the PID instruction executes the Auto-tuning function, the system will get three important parameters of PID.
- The Auto-tuning function can help user to get those three important parameters of the PID then to simplify the operation of PID instruction.
- This instruction is using relay "ON"/"OFF" to execute the Auto-tuning function, then evaluates three important parameters of the PID: Proportional gain (K_P), Integral time constant (T_I), Derivative time constant (T_D).
- The steps to execute the Auto-tuning function:

① Input the direction of action S3 +1, Sampling Time Ts, constant of Input Filter (α), Derivative Filter Constant K_D and Set Point Value S1.

② Input the parameters S3 +14 and S3 +15.

Parameters	Parameter Name/Function	Description
S3 +14	The Max. Output Value	The output value when it is at 100% output operation
S3 +15	The Mini. Output Value	The output value when it is at 0% output operation

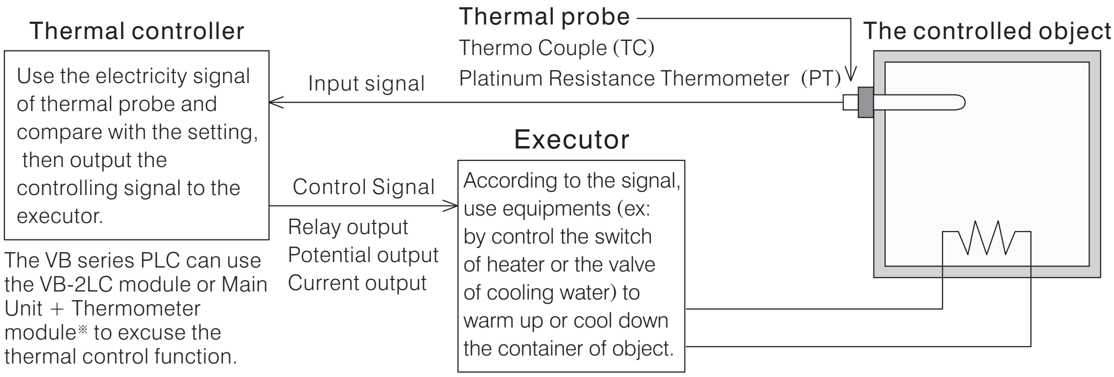
③ Let the parameter of S3 +1's b4= "ON", then it will start to execute the Auto-tuning operation.

④ When the Auto-tuning operation is finished, the parameter of S3 +1's b4 will automatically turned "OFF".

The General Idea of Thermal Control

Usually use the PID instruction contain in a PLC control system is for the thermal control. The following pages are the brief expositions about the thermal control.

- The construct of a thermal control system



※ The VB series PLC provide various thermometer module:

VB-8T ：8 points K or J type Isolated Thermo Couple input thermometer module.
VB-4T ：4 points K or J type Isolated Thermo Couple input thermometer module.
VB-4PT：4 points 3 wires PT-100 / 3850 ppm/°C input thermometer module.
VB-2PT：2 points 3 wires PT-100 / 3850 ppm/°C input thermometer module.

- The brief explanation of the thermal control

To set up the set piont value of thermal controller and let it functioning. The object may not steady changing the temperature immediately to the target temperature because the characteristic of the object.

In general, to expedite the responsive speed, it may cause overheat or waved temperature control. If want to reduce the those reaction, we should lower the volume of the response.

Some of the perform is like the Chart (1), which wants to control the temperature to the set point value as soon as possible. Under this condition, the temperature of object may overshooting the set point value, so it can be used only at the object is not concerned about overheat.

Some of the perform is like the Chart (2), which spends more time to get the smoothly thermal control. It is required the suppression of overshooting, so the longer time is required for stabilize temperature.

The Chart (3) is showing a compromise curve. That has an ideal responsive value, so it is the most popular type.

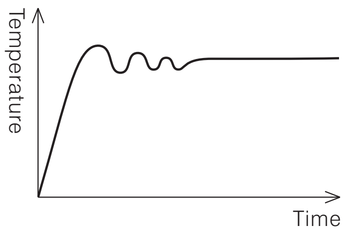


Chart (1), the Overshooting and Waving Response

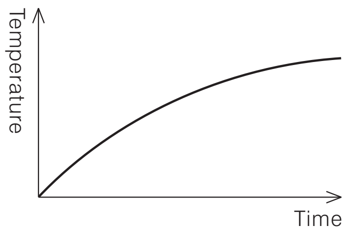


Chart (2), the Inert Response

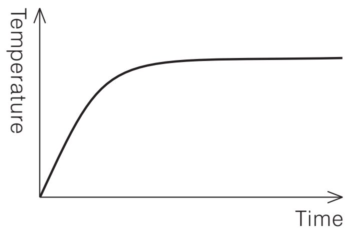


Chart (3), the Ideal Response

- The brief explanation of the thermal control

For the purpose of an ideal thermal control, when choose a thermal sensor and pick the controlling parameters, it is necessary to fully understand the characteristics of controlled object.

(1) Heat Capacity： How difficult to change the temperature, it may relate to the size of object.

(2) Heating Static Characteristics：It is indicate the capability of heating, which depends on the output capacity of heater.

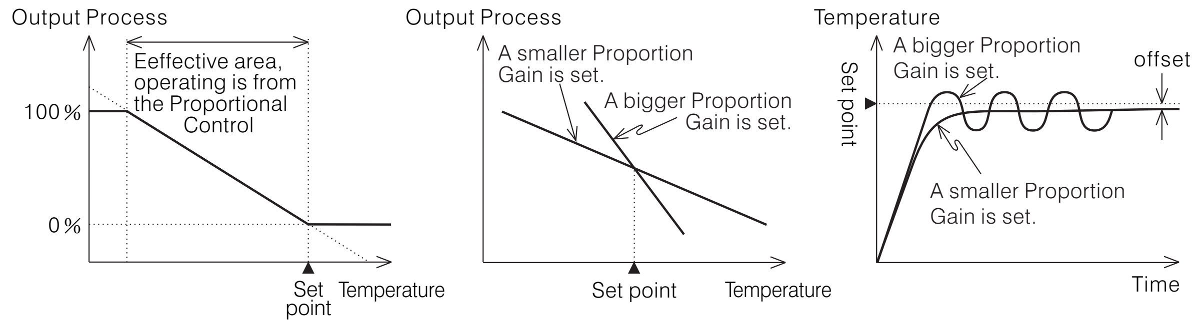
(3) Initially Dynamic Characteristic：At the beginning of heating, the characteristic of temperature changing which is complicated relationship with container and heater.

(4) External Disturbances : Some of the interference changes the temperature. ex. a door of the constant temperature furnace is opened.

- The PID Parametric Explanations

(1) P (Proportional Control) Action

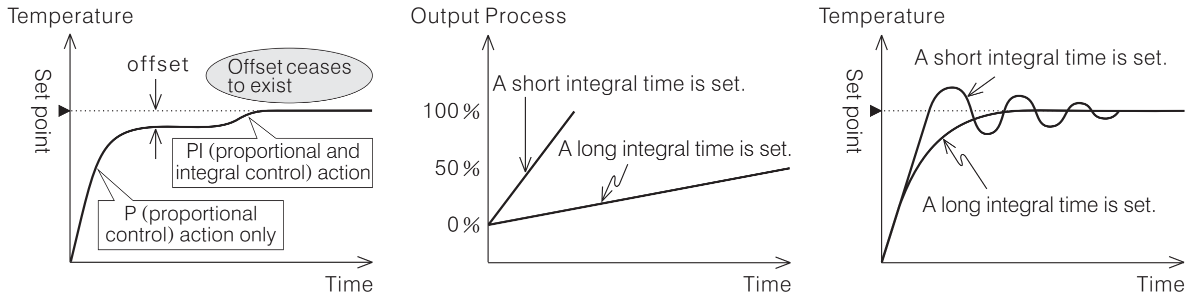
There is a value of difference between the set point (SV) and process value (PV), then the error value multiply by the Proportion Constant K_P(%) (or the Proportion Gain) will get a manipulative value to control the output.
So, if the output process value is generated by the error value or they have a specific direct ratio, the effect is called the Proportion operation.



(2) I (Integral Control) Action

It helps to achieve control at the set point and used for obtaining the output in proportion to the time integral value of the input.

P action causes an offset. Therefore, if proportional control action and integral control action are used in combination, the offset will be reduced as the time goes by until finally the control temperature will coincide with the set point and the offset will cease to exist.

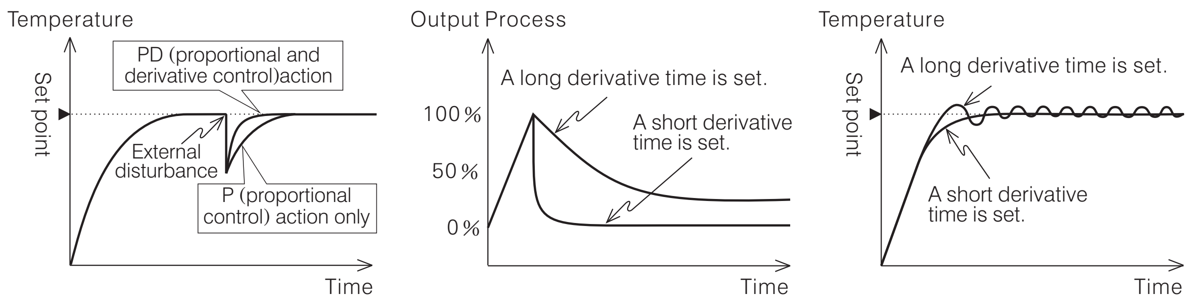


(3) D (Derivation or Rate Control) Action

D action (derivative or rate control action) is used for obtaining the output in proportion to the time derivative value of the input.

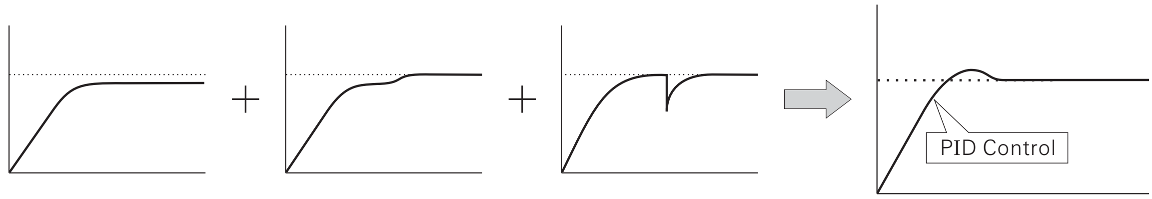
It provides a sudden shift in output level as a result of a rapid change in actual temperature. Proportional control action corrects the result of control and so does integral control action. Therefore, proportional control action and integral control action respond slowly to temperature change, which is why derivative control action is required. Derivative control action corrects the result of control by adding the control output in proportion to the slope of temperature change.

A large quantity of control output is added for a radical external disturbance so that the temperature can be quickly in control.



(4) PID Control

PID control is a combination of P(proportional), I (integral) and D (derivative) control actions, in which the temperature is controlled smoothly by proportional control action without hunting, automatic offset adjustment is made by integral control action, and quick response to an external disturbance is made possible by derivative control action.



- Control Cycle and Time-Proportioning Control Action

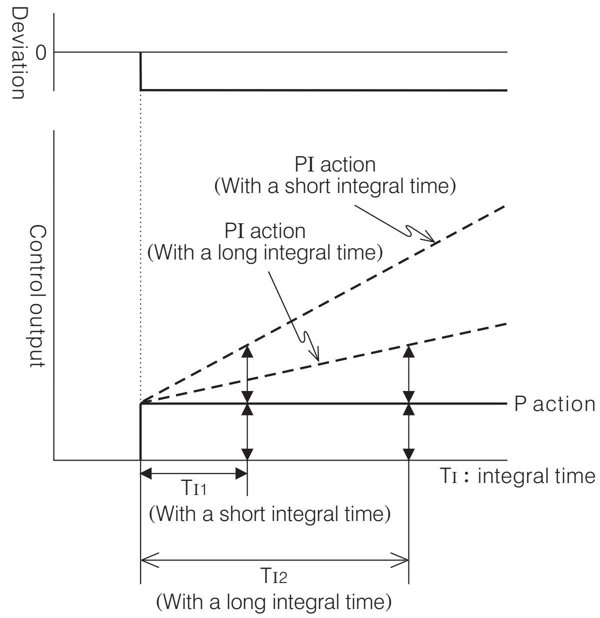
  When the temperature control is used with a relay or SSR to control the output, it will follow the premeditated timing cycle to turn “ON” or “OFF” a specified time intermittently. This preset cycle is called control cycle and this control method is called time-proportioning control action. A PLC system in the main unit is always using this method to procure temperature control.

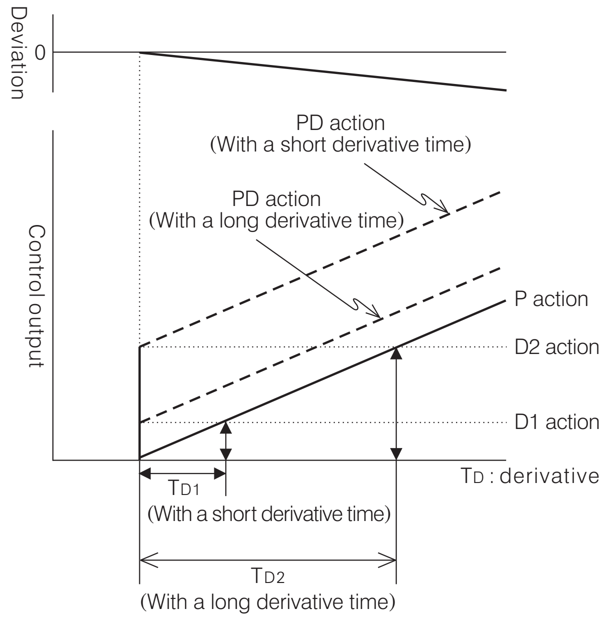
- The Definition of Integral Time

  Integral time is the period required for a step-type deviation in integral control (e.g., the deviation shown in the following graph) to coincide with the control output in proportional control action. The shorter the integral time is, the stronger the integral control action is. If the integral time is too short, it will cause a quick and huge correction then the temperature wave may result.

- The Definition of Derivative Time

  Derivative time is the period required for a ramp-type deviation in derivative control (e.g., the deviation shown in the following graph) to coincide with the control output in proportional control action. The longer the derivative time is, the stronger the derivative control action is.





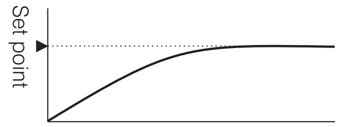
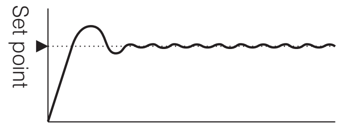
- Auto-Tuning

  All PID process/temperature controllers require the adjustment of the P, I, D and other parameters in order to allow accurate control of the load. There have been a variety of conventional methods but the Auto-tuning methods make it possible to obtain PID constants suitable to a variety of objects automatically.

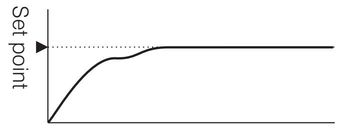
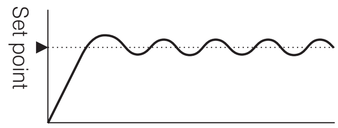
- Adjust the PID Parameters

  It is convenient while the PID constants calculated via the auto-tuning operation and normally they are more correct than tuning by manual. Usually, the auto-tuning do not cause problems and we will suggest using it to set up the parameters. Except for some particular applications if the more accurate constants is necessary. In which case, refer to the following to readjust the PID constants.

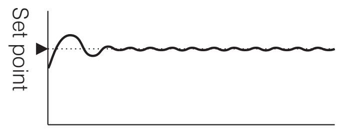
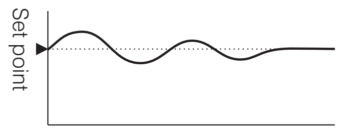
  Response to Change in Proportional Constant (Gain)

Smaller		It is possible to suppress overshooting although a comparatively long startup time and set time will be required.
Bigger		The process value reaches the set point within a comparatively short time and keeps the temperature stable although overshooting and waving will result until the temperature becomes stable.

Response to Change in Integral Time

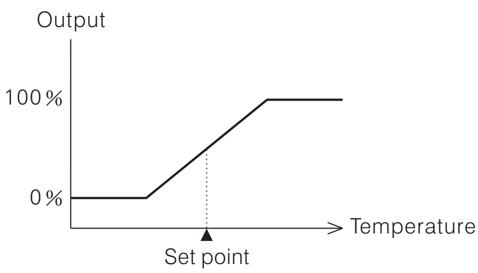
Wider		It is possible to reduce waving, overshooting and undershooting although a comparatively long startup time and set time will be required.
Narrower		The process temperature reaches the set point within a comparatively short time although overshooting, undershooting and waving will result.

Response to Change in Integral Time

Wider		The process value reaches the set point within a comparatively short time with comparatively small amounts of overshooting and undershooting although fine-cycle waving will result due to the change in process value.
Narrower		It will take a comparatively long time for the process value to reach the set point with heavy overshooting and undershooting.

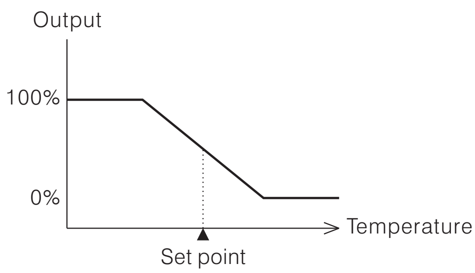
- Forward (Normal) Operation

  To increase the control output operation when the temperature of object is higher than the set point.



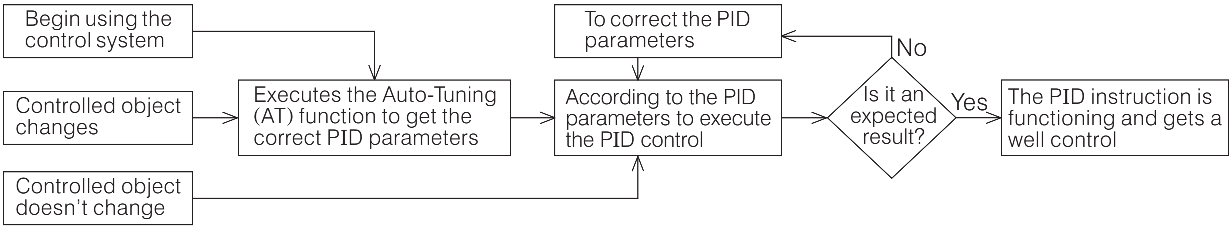
- Reverse Operation

  To increase the control output operation when the temperature of object is lower than the set point.

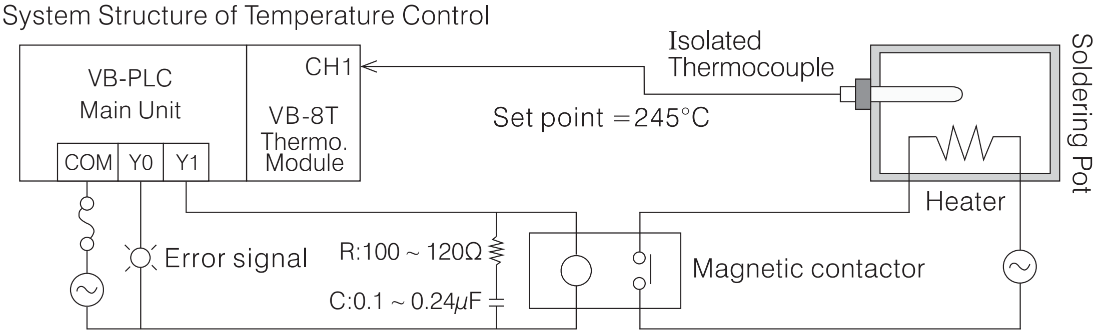


The Example of PID Temperature Control

- When design a PID temperature control program, the method below is the recommendable procedure to perform the PID instruction.

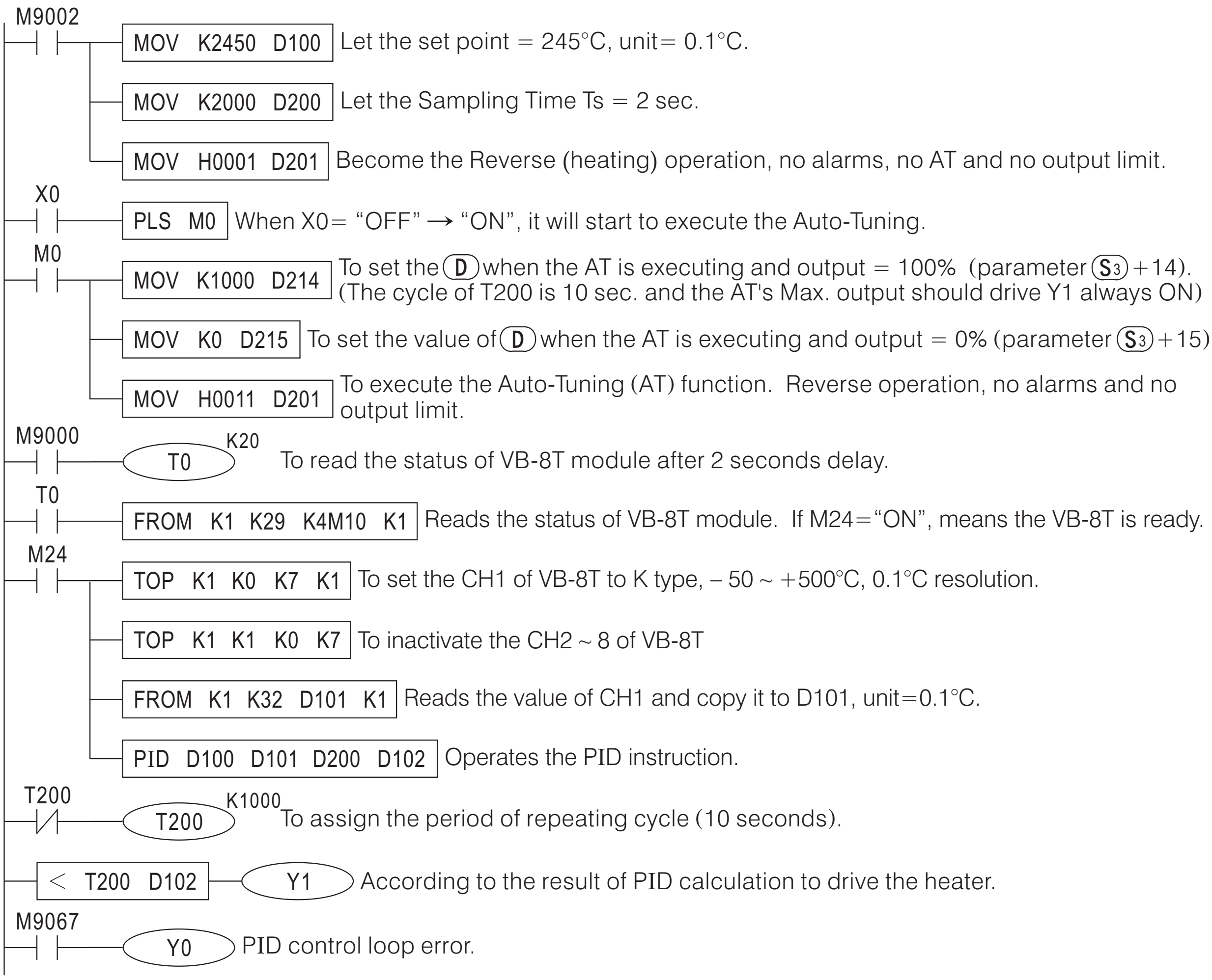


- The System Structure of Temperature Control



- Program Example

When X0="ON", it will executes Auto-Tuning function, and then starts the PID control; Otherwise, when X0= "OFF", it will executes the PID function directly.
This program is to control the "ON"/"OFF" length percentage in a specific time-span (10 seconds).
When this program starts at the first time, must let X0= "ON", then by the Auto-Tuning to get parameters of PID. Otherwise, the PID control will occur error because the related parameters aren't ready yet.



FNC 92 TPID	TPID (S1) (S2) (S3) (S4) (n)	Temperature PID Control	M: — / VB: ○ / VH: —

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					
S2											○					
S3											○					
S4											○					
n															○	

- S1 occupies n consecutive registers
- S2 occupies n consecutive registers
- S3 occupies (10×n)+10 consecutive registers
- S4 occupies 6×n consecutive registers
- $1 \leq n \leq 16$

```
          (S1)   (S2) (S3)  (S4)  (n)
  X0
--| |-----[TPID D7000 D0 D100 D7100 K16]
```

S1 : Head register ID of the Setting Value (SV) block

S2 : Head register ID of the Present Values (PV) block

S3 : Head register ID of the parameters & outputs

S4 : Head register ID of the parameters of PID & other setting values

n : Number of object channels need to control by this instruction

- This TPID instruction is especially for temperature application at the multi-object (1~16) PID control. The instruction provides temperature PID control, Auto-Tuning (AT), Auto/Manual control functions and alarms. So, the instruction can easily procure a smooth temperature control.
- Uses the difference between (S1) (one in the setting value block) and (S2) (correlated one in the present value block), then via the values of parameters in (S3) and (S4) to process the PID operate.
  The control result signal of coil ON/OFF will effect relative bit at (S3)+5. If the analog control output is required, the resulted value of PID will appear at correlated register of (S3).
- When X0= "ON", this instruction starts to perform; When X0="OFF", this process stops and all the output contacts at (S3)+5 will be turned "OFF" also all the analog output values in the (S3) will be reseted to "0".
- There's no limitation on the times used of the TPID instruction.
- This instruction provided with the "Auto-Tuning (AT)" function, it can help users to decide the parameters of P (K_P), I (T_I) and D (T_D) at the TPID instruction. (Please refer to following pages.)
- This instruction accumulates the values of difference between (S1) & (S2) block at every PLC Scan Time, those with parameters become parts of operand then effect the control output cycles. So, to use this instruction must pay attention to the suggestion below:
  This TPID instruction can be used in the SFC Ladder Chart, Subprogram, Interrupted Subprogram and the block of Jump instruction. If it has been pass over or process more than once, the input values may be calculated none or repetitiously. So, make sure the active instruction has been process once and only once at every Scan Time otherwise the result may be incorrect.
- The specification of Setting Value (SV) (S1) block
  By the content value of parameter (n) to establish the number of object channels then the (S1) block will occupy n registers.
  The content value of (S1) is the Setting Value (SV) for the first object channel; the content value of (S1)+1 is the Setting Value (SV) for the second object channel; and so on.
- The specification of Present Values (PV) (S2) block
  By the content value of parameter (n) to establish the number of object channels then the (S2) block will occupy n registers.
  The Present Value (PV) in (S2) is from the sensor of the first object channel; the Present Value (PV) in (S2)+1 is from the sensor of the second object channel; and so on.

- The specification of parameter block S_3

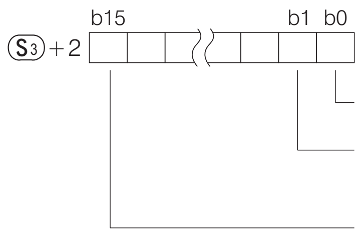
Para-meter	Parameter Name/Function	Description	Setting range
S_3	Control Cycle Setting	To assign the outputs period interval (the length of one ON/OFF cycle)	10~32767 ×10 ms.
S_3+1	Responsive Sensitivity	To assign the sensitive level of the instruction which is for all channels (“0”: sensitive - - “3”: insensitive)	0 ~ 3
S_3+2	Operational Direction	By relative bits at this register to assign the reacted direction of channels (“0”: “Reverse” / “1”: “Forward”)	H0000 ~ HFFFF
S_3+3	Auto/Manual Select	By relative bits at this register to assign the control method of channels (“0”: Automatic / “1”: Manual)	H0000 ~ HFFFF
S_3+4	AT Command	By relative bits at this register to start the Auto-Tuning (AT) function of channels (“1”: start AT; reset to “0” when AT has been finished)	H0000 ~ HFFFF
S_3+5	Outputs	Output the control signals for object channels by relative bits	—
S_3+6	Limitation Alarm Status	Display the limitation alarms for object channels by relative bits	—
S_3+7	Deviation Alarm Status	Display the deviation alarms for object channels by relative bits	—
S_3+8 S_3+9	System Operating Area	Reserved for the internal processing of the TPID instruction	—
S_3+10	The First Object Analog Output	Display the analog output value of the first object channel	0 ~ 1000 ×0.1%
S_3+11 S_3+19	The First Object Operating Area	Reserved for the internal processing of the TPID instruction	—
S_3+20	The Second Object Analog Output	Display the analog output value of the second object channel	0 ~ 1000 ×0.1%
S_3+21 S_3+29	The Second Object Operating Area	Reserved for the internal processing of the TPID instruction	—
⋮	⋮	⋮	⋮

- The values in S_3~S_3+9 are the common parameters for all object of this instruction.
  And, to add any object channel will occupy extra 10 registers form S_3+10 to $S_3+(10×n)+9$.

- The parameter at S_3 is the control output period setting for this instruction.
  Usually, the length of control period depends on the type of loading.
  If the the equipment is driven by a Magnetic Contactor (MC), to set the value bigger than 1000 (1000×10ms. = 10 Sec.) is recommend that is for extend its lifespan.
  If the the equipment is driven by a Solid State Relay (SSR), can set the value to 200 (200×10ms.=2 Sec.)

- The parameter at S_3+1 is to set up control sensitivity for the response of this instruction.
  The value in S_3+1 will affect all object channels in the instruction.
  To control the temperature of a system, always expect its response as soon as possible but in some condition the quick response will cause temperature waving then occur a unsuccessful control.
  Therefore, could adjust its level of sensitivity to get a better control. To input the value equal to “0” is the fast response; “1” is medium; “2” is slow.

- Each bit at S_3+2 is for set up control direction of every single object channel.
  When the measured Present Value (PV) < the Setting Value (SV), it will generate a negative deviation and increase the control effect, that is called the “Reverse” operation. Ex. An oven: before the heater of the oven turns on. Usually the temperature of the oven is lower than the setting value. (PV) < (SV), this is a typical “Reverse” operation control sample.
  When the measured Present Value (PV) > the Setting Value (SV), it will generate a positive deviation and increase the control effect, that is called the “Forward” operation. Ex. An air conditioning system: before the system turns on. Usually the indoor temperature is higher than the setting value. (PV) > (SV), this is a typical “Forward” operation control sample.



“0” stands for the “Reverse” (heating) control;
“1” stands for the “Forward” (cooling) control

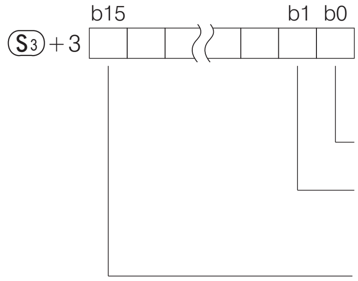
To assign the direction of the first object channel

To assign the direction of the second object channel

⋮

To assign the direction of the sixteenth object channel

- Each bit at (S3)+3 is for set up Auto/Manual control of every single object channel.



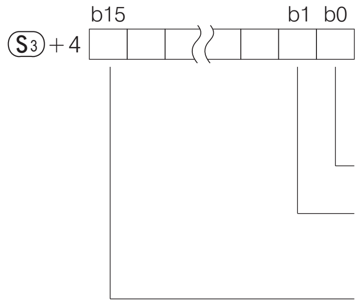
"0" stands for PID automatic control; "1" stands for manual control. When using the manual control mode, should input the expected output value (0 ~ 1000) directly to relative register.

To assign the A / M method of the first object channel

To assign the A / M method of the second object channel

To assign the A / M method of the sixteenth object channel

- Each bit at (S3)+4 is for trigger the Auto-Tuning (AT) function of every single object channel.



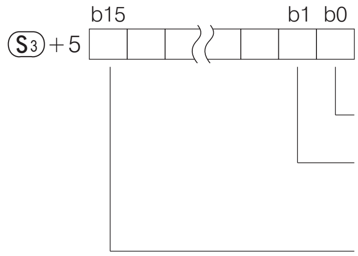
To trigger the b0 "ON", the first object channel will start to process the AT function; when its AT function finished, the b0 will be reset to "OFF" and the parameters P (K_P), I (T_I) and D (T_D) will put into relative registers in (S4) block.

To trigger the AT function of the first object channel

To trigger the AT function of the second object channel

To trigger the AT function of the sixteenth object channel

- Each bit at (S3)+5 is for output the "ON" / "OFF" control signal of every single object channel.



By the (S3) period and the calculated result of TPID instruction to produce outputs (proportional "ON" / "OFF" rate in every period).

To output the control signal of the first object channel

To output the control signal of the second object channel

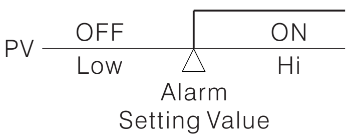
To output the control signal of the sixteenth object channel

Each object channel will also generate an analog PID output value.
The results are output to (S3)+10×m (m=1 ~ n). Ex. The (S3)+10 is the output value of the first object channel; the (S3)+20 is the output value of the second object channel; and so on. Those output values can be used for the digital-analog (D/A) convert circuits to perform the analog control outputs.
This (S3)+5 outputs are using those values in (S3)+10×m (m=1 ~ n) to produce proportional "ON" / "OFF" output signals.
But at the manual control method, should put the expected output values (0 ~ 1000) into relative registers.

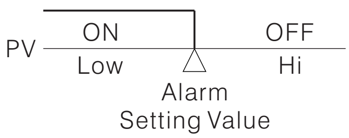
- This instruction provides two alarm signals for each object channel. See the illustrations below.

① Limitation Alarm

When a object channel uses the "Reverse" operation, the Limitation Alarm will "ON" if the PV is higher then the alarm setting value.

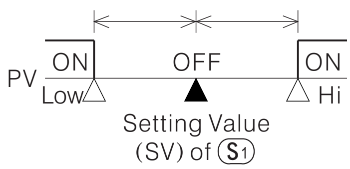


When a object channel uses the "Forward" operation, the Limitation Alarm will "ON" if the PV is lower then the alarm setting value.

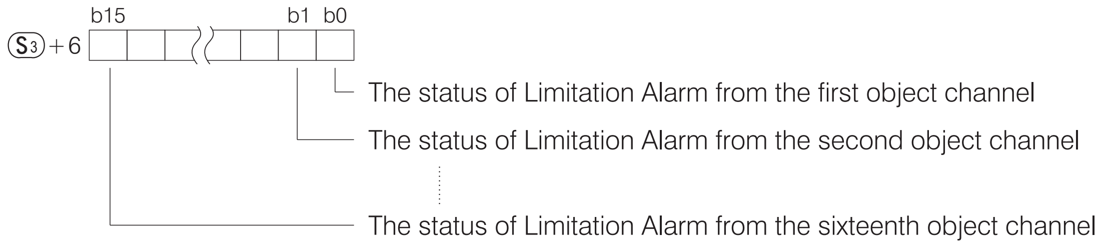


② Deviation Alarm

Allowably (+/–) deviant value.

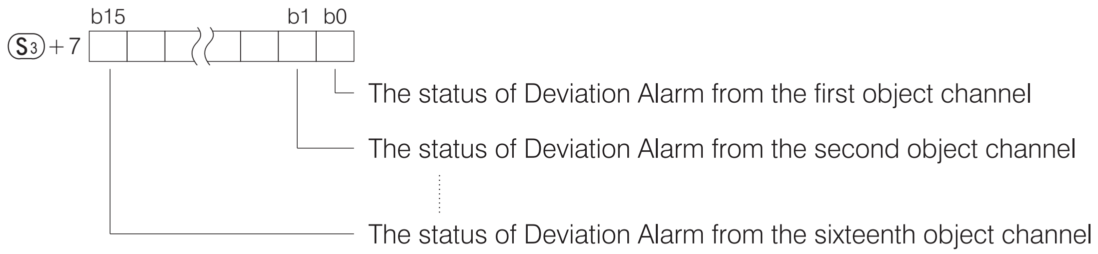


- Each bit at S_3+6 is for storage the status of Limitation Alarm of every single object channel.



A object channel has a setting value of Limitation Alarm which is put in S_4+6m+4 (m=0 ~ n–1); the S_4 +4 is for the first object channel; the S_4+10 is for the second object channel; and so on.

- Each bit at S_3+7 is for storage the status of Deviation Alarm of every single object channel.



An object channel has a setting value of Deviation Alarm which is put in S_4+6m+5 (m=0 ~ n–1); the S_4 +5 is for the first object channel; the S_4+11 is for the second object channel; and so on.

- The exposition of parameter block S_4 (K_P, T_I, T_D & other setting values)

Para-meter	Parameter Name/Function	Description	Setting range
S_4	Proportional Gain (K_P) of the First Object Channel	The P (Proportional) part of the PID loop	1～32767 ×0.01
S_4+1	Integral Time Constant (T_I) of the First Object Channel	The I (Integral) part of the PID loop, (this parameter disables the I effect if it is "0")	0～32767 ×100 ms.
S_4+2	Derivative Time Constant (T_D) of the First Object Channel	The D (Derivative) part of the PID loop (this parameter disables the D effect if it is "0")	0～32767 ×10 ms.
S_4+3	Overshoot Repression Value of the First Object Channel	To set this repression deviation appropriately could repress the overshoot at the beginning	0～32767
S_4+4	Limitation Alarm Setting Value of the First Object Channel	For the "Reverse" operation: Limitation Alarm "ON" if PV > this setting value. For the "Forward" operation: Limitation Alarm "ON" if PV < this setting value.	–32768～32767
S_4+5	Deviation Alarm Setting Value of the First Object Channel	Deviation Alarm "ON" if PV > (SV + this setting value) or PV < (SV - this setting value)	–32768～32767
S_4+6	Proportional Gain (K_P) of the Second Object Channel	The P (Proportional) part of the PID loop	1～32767 × 0.01
S_4+7	Integral Time Constant (T_I) of the Second Object Channel	The I (Integral) part of the PID loop, (this parameter disables the I effect if it is "0")	0～32767 ×100 ms.
S_4+8	Derivative Time Constant (T_D) of the Second Object Channel	The D (Derivative) part of the PID loop, (this parameter disables the D effect if it is "0")	0～32767 ×10 ms.
S_4+9	Overshoot Repression Value of the Second Object Channel	To set this repression deviation appropriately could repress the overshoot at the beginning	0～32767
S_4+10	Limitation Alarm Setting Value of the Second Object Channel	For the "Reverse" operation: Limitation Alarm "ON" if PV > this setting value. For the "Forward" operation: Limitation Alarm "ON" if PV < this setting value.	–32768～32767
S_4+11	Deviation Alarm Setting Value of the Second Object Channel	Deviation Alarm "ON" if PV > (SV + this setting value) or PV < (SV - this setting value)	–32768～32767
⋮	⋮	⋮	⋮

- The register block starting from S_4 is for storage every channel's K_P, T_I, T_D parameters, starting Overshoot Repression and two alarm setting values.
  Every channel will occupy 6 sequential registers. The S_4~S_4+5 keep parameters for the first channel; the S_4+6 ~S_4+11 keep parameters for the second channel; and so on.

- Registers for the block of S_4 are usually assigned to latched registers.

- Every channel's K_P, T_I and T_D parameters could use the Auto-Tuning (AT) function to get the values, also available given by user.

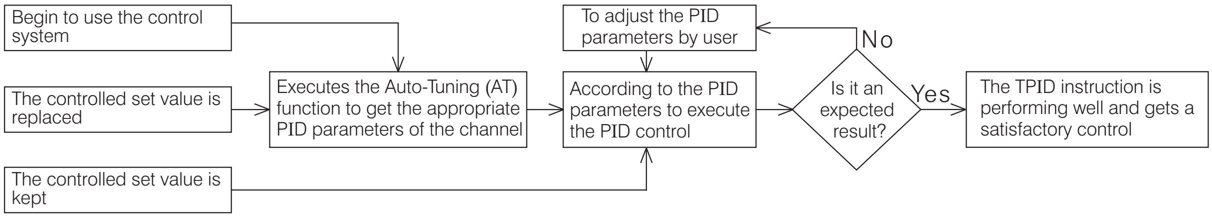
- The unit of the Overshoot Repression follows the SV value. If the unit of SV is 0.1°C (usually), then to use the function of starting Overshoot Repression, its unit is equal to 0.1 °C also.
  If the application of temperature control which is sensitive to the starting overshoot, the channel could use this function and appropriately set the deviation value then it can effectively repress starting overshoot. To get this repressive value, could observe the maximum overshoot at AT processing. Or, approximately preset a value (10.0 ~ 20.0 °C) to do an experiment then use the result to adjust the repressive value.

- To read the statuses of alarms which are appointed by the parameter block S_4 , please refer to the instruction of S_3+6 and S_3+7.

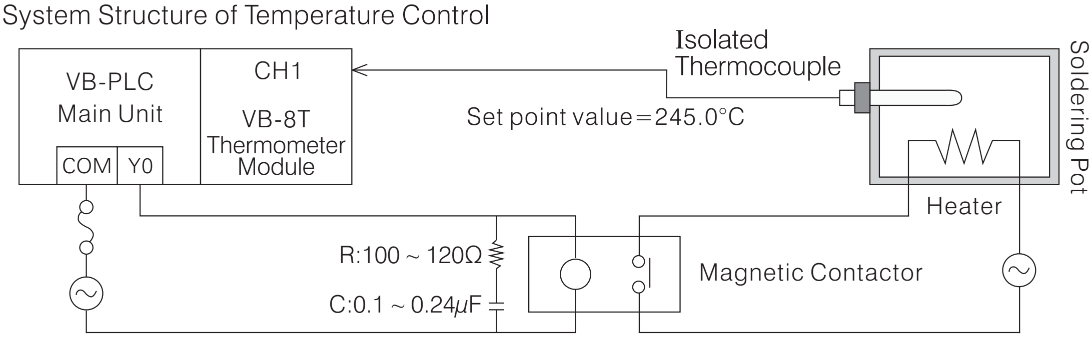
- This instruction will be valid if a VB series V1.70 or later is used.

TPID Instruction Temperature Control Example I

- When design a PID temperature control program, the method below is the recommendable procedure to perform the TPID instruction.

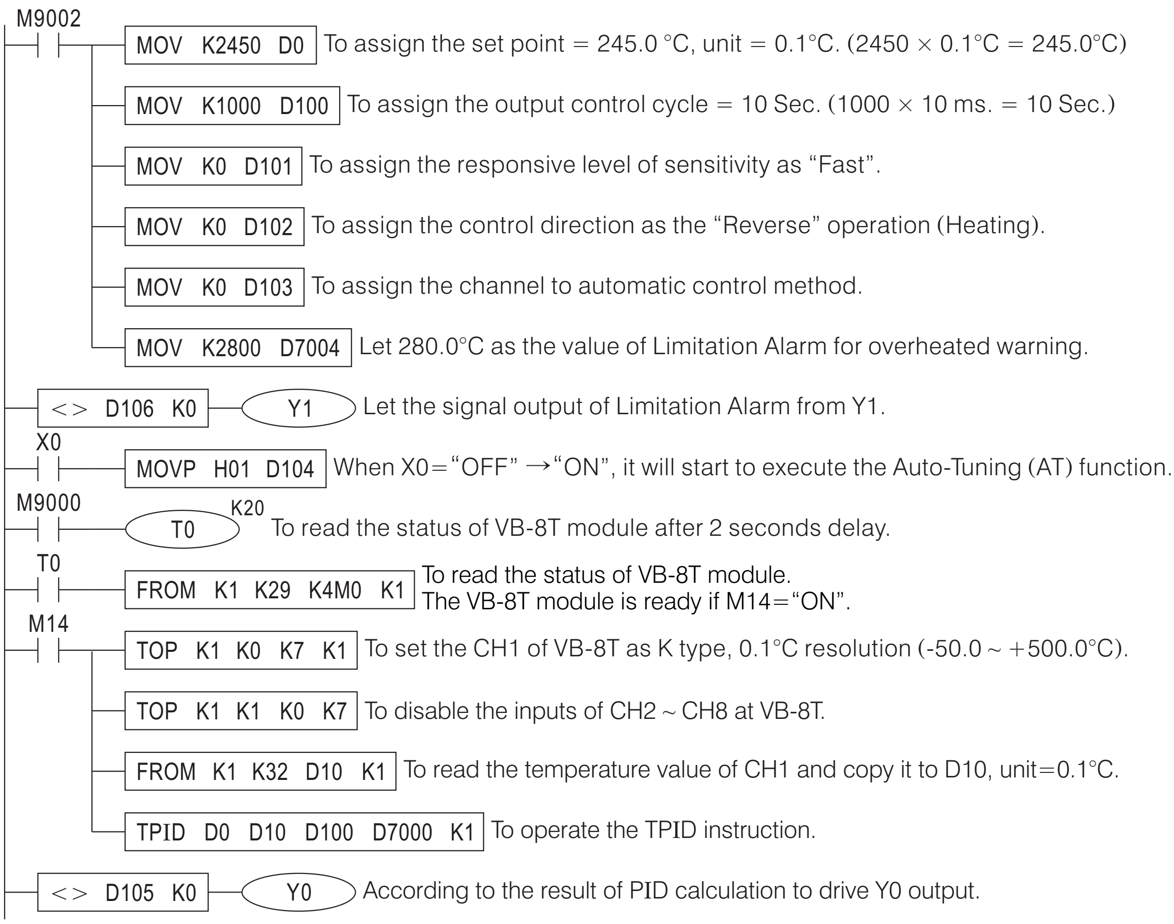


- The System Structure of Temperature Control



- Program Example

When X0="ON", it will execute the Auto-Tuning (AT) function first and then start the PID control; Otherwise, when X0="OFF", it will execute the PID operation directly.
Must be trigger the X0="ON" once if this program is started at the first time, then by the Auto-Tuning (AT) function to get the P, I and D parameters of the channel. Otherwise, the PID control will occur error because the related parameters are not ready yet.



TPID Instruction Temperature Control Example II

This is a 16 channels temperature control example, which needs a 32 points VB series Main Unit and two VB-8T modules also a HMI (Human Machine Interface) is required for data settings and statuses display.

- The components list at this example:

Controlled CH #	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Auto/Manual Select	M2015	M2014	M2013	M2012	M2011	M2010	M2009	M2008	M2007	M2006	M2005	M2004	M2003	M2002	M2001	M2000
AT Command	M35	M34	M33	M32	M31	M30	M29	M28	M27	M26	M25	M24	M23	M22	M21	M20
AT Status	M55	M54	M53	M52	M51	M50	M49	M48	M47	M46	M45	M44	M43	M42	M41	M40
Output Point	Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Limitation Alarm Status	M75	M74	M73	M72	M71	M70	M69	M68	M67	M66	M65	M64	M63	M62	M61	M60
Deviation Alarm Status	M95	M94	M93	M92	M91	M90	M89	M88	M87	M86	M85	M84	M83	M82	M81	M80
Temp. Setting Value (SV)	D7015	D7014	D7013	D7012	D7011	D7010	D7009	D7008	D7007	D7006	D7005	D7004	D7003	D7002	D7001	D7000
Temp. Present Value (PV)	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Parameter of P Phase (K_P)	D7190	D7184	D7178	D7172	D7166	D7160	D7154	D7148	D7142	D7136	D7130	D7124	D7118	D7112	D7106	D7100
Parameter of I Phase (T_I)	D7191	D7185	D7179	D7173	D7167	D7161	D7155	D7149	D7143	D7137	D7131	D7125	D7119	D7113	D7107	D7101
Parameter of D Phase (T_D)	D7192	D7186	D7180	D7174	D7168	D7162	D7156	D7150	D7144	D7138	D7132	D7126	D7120	D7114	D7108	D7102
Overshoot Repression Value	D7193	D7187	D7181	D7175	D7169	D7163	D7157	D7151	D7145	D7139	D7133	D7127	D7121	D7115	D7109	D7103
Limitation Alarm Value	D7194	D7188	D7182	D7176	D7170	D7164	D7158	D7152	D7146	D7140	D7134	D7128	D7122	D7116	D7110	D7104
Deviation Alarm Value	D7195	D7189	D7183	D7177	D7171	D7165	D7159	D7153	D7147	D7141	D7135	D7129	D7123	D7117	D7111	D7105

Besides the components on the table above, this instruction will occupy the registers D100 ~ D269. When actually use this instruction, some unnecessary control items (Ex. Auto/Manual control selection) could remove from the program then those items would not occupy components.

- Program Example

M9002
MOV K1000 D100 — To assign the output control cycle = 10 Sec. (1000 × 10 ms. = 10 Sec.)
MOV K0 D101 — To assign the responsive level of sensitivity for CH1 ~ CH16 as "Fast".
MOV H0 D102 — To assign the control direction of CH1 ~ CH16 as the "Reverse" operation (Heating).

M9000
MOV K4M2000 D103 — To setup Auto/Manual control method of CH1 ~ CH16 by Latched M2000 ~ M2015.
WOR K4M20 D104 D104 — Use the M20 ~ M35 to trigger the AT function of CH1 ~ CH16. At this application must use the WOR instruction, can't use the MOV instruction.
MOV D104 K4M40 — Use the M40 ~ M55 to display the status of AT function of CH1 ~ CH16. If a coil "ON", the corresponded channel processing the AT function.
MOV D105 K4Y0 — To assign the Y0 ~ Y17 become the control outputs of CH1 ~ CH16, by the output signals to drive the loaders.
MOV D106 K4M60 — To assign the M60 ~ M75 become the Limitation Alarm outputs of CH1 ~ CH16.

M9013
C0 K3600 — After one hour, C0 turns "ON" to start the Deviation Alarm monitors, that could ensure the system is processing under normal temperature.

C0
MOV D107 K4M80 — To assign the M80 ~ M95 become the Deviation Alarm outputs of CH1 ~ CH16.

M9000
T0 K20 — To read the status of VB-8T modules after 2 seconds delay.

T0
FROM K1 K29 K4M100 K1 — To read the status of the first VB-8T module. The first VB-8T module is ready if M114="ON".
FROM K2 K29 K4M120 K1 — To read the status of the second VB-8T module. The second VB-8T module is ready if M134="ON"

M114
TOP K1 K0 K7 K8 — To set all the channels of the first VB-8T as K type, 0.1°C resolution (–50.0 ~ +500.0°C).
FROM K1 K32 D0 K8 — To read the temperature values of CH1 ~ CH8 from the first VB-8T and copy it to D0 ~ D7, unit=0.1°C.

M134
TOP K2 K0 K7 K8 — To set all the channels of the second VB-8T as K type, 0.1°C resolution (–50.0 ~ +500.0 °C).
FROM K2 K32 D8 K8 — To read the temperature values of CH9 ~ CH16 from the second VB-8T and copy it to D8 ~ D15, unit=0.1°C.

M114 M134
TPID D7000 D0 D100 D7100 K16 — To operate the TPID instruction.

FNC 149 MBUS	MBUS (S1) (S2)	MODBUS Communication	M	VB	VH
				○	

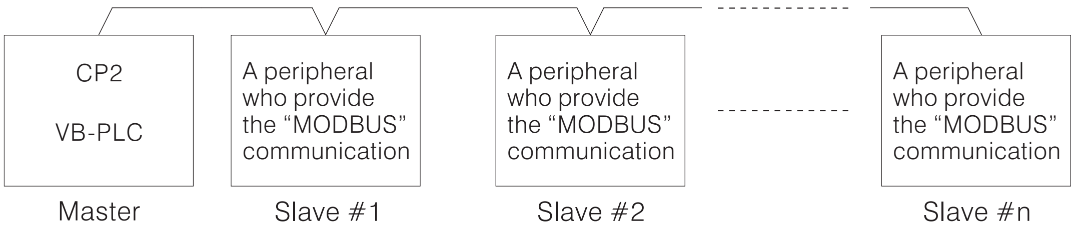
Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					
• S2 occupies 4 consecutive registers																

X20 — MBUS D1000 D100 [(S1) = D1000, (S2) = D100]

S1 : To indicate the head ID number of receiving/sending data registers

S2 : Instruction working area, occupies 4 consecutive registers

- This section is for VB series PLC only; for the MBUS instruction in a VH series PLC, please revere to page 359.
- When a VB Series Main Unit has been installed a communication card (VB-232R or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between the PLC and a device who has MODBUS communication protocol.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 would like to use for this instruction, the communication type of CP2 should chose the "MODBUS". To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master "System---2nd COM Port Setting..."to get the right setting.
- As the diagram below, use the CP2 to connect the PLC and other peripherals, use the program develop devices (e.g. Ladder Master) to set the "MODBUS" communication mode and the communication parameters. Then, to properly finish all the setting of station IDs (the range of station ID number is 1 ~ 247, but when this system link is used the RS-232, there is only one slave available) and parameters for slaves (or peripherals). Write the data transmission/receiving command to the PLC (Master station), to drive the data transmission between PLCs or peripherals.



- When X20="ON", the MBUS instruction will start to be performed. Based on the designated register string(which initiating from D1000), to process writes/reads data into/from an appointed Slave PLC or peripheral. At the same time, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by (S1) is duly completed, the M9199 will be "ON" for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20="ON" → "OFF", the instruction will be stopped and the data transmission/receiving will be discontinued immediately.
- The MBUS instruction is for the Master PLC, it can be used once only and do not use the LINK or RS instruction in the program.
- For avoid the corresponding breakup, when the MBUS instruction sends a communication request to a particular Slave, if the respondent time of the Slave exceeds the Time-out duration (designated by D9129), the MBUS instruction will stops communication from the specific Slave and operates next communication command.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) ×10ms. When D9129=0 (the default value), the Time-out duration is 100 ms.
- Most of the applied situation is not necessary to change the Time-out duration. But, if an equipment in the communication link, its response is very slow, then the longer Time-out duration is necessary.

- The register headed with S_1 is used to describe the data transmission/receiving information:

S_1	Content Value	Description
D1000	1 ~ 255	To designate the number of transferred and received data sets. Each data transmission/receiving set should be described with 7 registers.
D1001	1 ~ 247	Designates the Slave station ID number, to proceed data transmission/receiving for the particular Slave station
D1002	1 ~ 3	Instruction commend. 1: read data from the Slave station; 2: write a series of data into the Slave station; 3: write one device's data into the Slave station.
D1003	1 ~ 64	Length of data transferred or received. If the instruction code (S_1+2)= 3, this data will be ignored.
D1004	1 ~ 6 10,11,13	Designates the device type of the Master station 1: Input Contact X 2: Output Contact Y 3: Auxiliary Coil M 4: State Coil S 5: Timer Contact T 6: Counter Contact C 10: The Present-value Register of the Timer 11: 16-bit Counter, Present-value Register 13: Data Register D
D1005		Designates the initial component ID number of the Master station device
D1006	0,1,3,4	Designates the device type of the Slave station 0: A readable/writable bit device 1: A readable only bit device 3: A readable only 16 bits data Register 4: A readable/writable 16 bits data Register
D1007	0 ~ 32767	Designates the initial component data ID number of the Slave station device
D1008	1 ~ 247	Designates the Slave station ID number
D1009	1 ~ 3	Instruction commend
D1010	1 ~ 64	Length of data transferred/received
D1011	1 ~ 6 10,11,13	Designates the device type of the Master station
D1012		Designates the initial component ID number of the Master station device
D1013	0,1,3,4	Designates the device type of the Slave station
D1014	0 ~ 32767	Designates the initial component data ID number of the Slave station device
⋮	⋮	

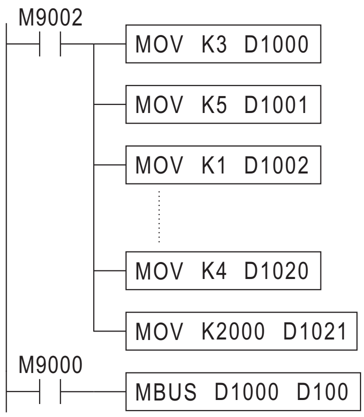
Description of the 1st data transmission/receiving operation (D1001 ~ D1007)

Description of the 2nd data transmission/receiving operation (D1008 ~ D1014)

- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.
- The instruction working area headed with S_2 :

S_2		Description
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
	Upper 8 bits	Instruction working status 0: Normal data transmission/receiving 2: Error of the length of the transferred/received data (unequal to 1 ~ 64) 4: Error of the designated device type 5: Error of the designated device ID number 6: The characteristic of devices between the Master and Slave stations are different A: Normal communications but no response from Slave stations B: Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

Description by an Example (For the VB series only)



To create a communication table.
This block can be replaced by the function of File Registers

MODBUS communication instruction

There are totally 3 transmission/receiving data sets in this example.

① To read the data in 40000 ~ 40009 of Slave station #5 and put they to D2000 ~ D2009 of the Master station.

② To write the data in D2010~D2014 of the Master station into 41000 ~ 41004 of Slave station #2.

③ To write the data in D2015 of the Master station into 42000 of Slave station #3.

S1	Content Value	
D1000	3	Three transmission/receiving data sets
D1001	5	Designates Slave station #5
D1002	1	Reads data from the Slave station
D1003	10	Length of the data to be read
D1004	13	Designates the device in the Master station which headed with D2000
D1005	2000	
D1006	4	Designates the device in the Slave station which headed with 40000
D1007	0	
D1008	2	Designates Slave station #2
D1009	2	Write a series of data into the Slave station
D1010	5	Length of the data to be written
D1011	13	Designates the device in the Master station which headed with D2010
D1012	2010	
D1013	4	Designates the device in the Slave station which headed with 41000
D1014	1000	
D1015	3	Designates Slave station #3
D1016	3	Write the device's data to the Slave station
D1017	1	This information will be ignored
D1018	13	Designates the data in the Master station D2015
D1019	2015	
D1020	4	Designates the data in the Slave station 42000
D1021	2000	

D1001 ~ D1007: The first transmission/receiving data sets:
40000 ~ 40009 of Slave station #5
↓
D2000 ~ D2009 of the Master

D1008 ~ D1014: The second transmission/receiving data sets:
D2010 ~ D2014 of the Master
↓
41000 ~ 41004 of Slave station #2

D1015 ~ D1021: The third transmission/receiving one data set:
D2015 of the Master
↓
42000 of Slave station #3

- Use the File Registers to set up the communication table
  In the VB series PLC, the File Registers are read only registers and the their contents are assumed as a part of program.
  When a user copy or access the program file, the program itself and the File Registers will be handled together. Since the File Registers have this characteristic, use they to store the communication table were suitable. They are not only to copy the data of File Registers easily but also can minimize the program size. Please refer to CH 2-9 “File Register (D)” for more information about the File Register.
  To plan the contents of File Registers, which can use the programming tool software “Ladder Master”, it provide the edit tool “System ---- File Register Edit....”, easily to set the data in the registers.

- Edit Communication Table
  In addition to the File Registers' layout function; and further, the Ladder Master provides more user friendly and easily of data input interface, it provide the user to create and edit the Communication Table List.
  Please select the Ladder Master's “Tools ---- Edit Communication Table ....” function to start the Communication Table List document edit window. By the interlocutory pop-up window, user can easily create and edit the communication table step-by-step. After the Communication Table has been finished, the user can put the communication data into the designated File Registers then this communication table is completed. And also, this function provides user to retrieve, access and edit the Communication Table back from the File Registers.
  For the VB series PLCs, the File Register is read-only, and its value will be treated as a part of the user program. When user copy or save program file, the File Register together with the program itself will be copied or saved. This feature makes the File Register very suitable for communication table storing; it can be easily copied from and helps to save PLC program space. For detailed introduction on the File Register, please refer to the section “2-9 File Register (D)”.

- Communication Table example :

M9000

MBUS D1000 D100

Instruction: MBUS ▼ Start of File Reg: D1000 Length of Reg: 22

Number	Command	Master Data		Slave ID	Slave Data Type	Slave Data #	Length	Word / Bit
1	Read	D2000	<--	5	4	0	10	W
2	Write	D2010	-->	2	4	1000	5	W
3	Single Write	D2015	-->	3	4	2000	1	W

There are totally 3 transmission/receiving data sets in this Communication Table example.

(1) To read the data in 40000 ~ 40009 of Slave station #5 and put they to D2000 ~ D2009 of the Master station.

(2) To write the data in D2010 ~ D2014 of the Master station into 41000 ~ 41004 of Slave station #2

(3) To write the data in D2015 of the Master station into 42000 of Slave station #3.

The “Slave Data Type” and “Slave Data No.” in the communication table refers to the component ID number of the slave station equipment.
For example, there is a MODBUS component:

4 0 0 0 0

The component data ID No.

The component data type 0:Writable & Readable Bit Component

1:Read Only Bit Component

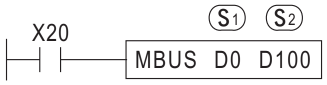
3:Read Only Data Register (16 bits)

4:Writable & Readable Register (16 bits), the most often type.

FNC 149 MBUS	MBUS (S1) (S2)	MODBUS Communication	M	VB	VH
					○

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S1											○					○
S2											○					

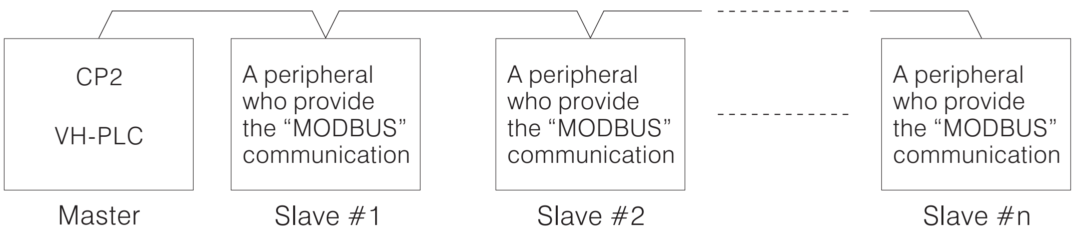
• S2 occupies 4 consecutive registers



S1 : To indicate a virtual register for the communication table

S2 : Instruction working area, occupies 4 consecutive registers

- This instruction is for the VH series PLC only. The MBUS instruction for VB series, please referee to page 355
- When a VH Series Main Unit has been installed a communication card (VB-232R or VB-485) or a communication module (VB-485A, VB-CADP etc.), the Main Unit will have the CP2 (2nd Communication Port). Then, via this instruction to proceed data transfer between the PLC and a device who has MODBUS communication protocol.
- The CP2 is a multi-functional expanded communication port, it can be used for multiplex communication types. When the CP2 would like to use for this instruction, the communication type of CP2 should chose the “MODBUS”. To select and relative parameters setting about the manipulation type of CP2, please use the option in the programming tool Ladder Master “System---2nd COM Port Setting...”to get the right setting.
- As the diagram below, use the CP2 to connect the PLC and other peripherals, use the program develop devices (e.g. Ladder Master) to set the “MODBUS” communication mode and the communication parameters. Then, to properly finish all the setting of station IDs (the range of station ID number is 1 ~ 247, but when this system link is used the RS-232, there is only one slave available) and parameters for slaves (or peripherals). Write the data transmission/receiving command to the PLC (Master station), to drive the data transmission between PLCs or peripherals.



- When X20=“ON”, the MBUS instruction will start to be performed. Based on the designated Comm Table string , to process writes/reads data into/from an appointed Slave PLC or peripheral. At the same time, D100 ~ D103 store the status of the instruction execution.
- Every time the transmission/receiving operation which designated by(S1)is duly completed, the M9199 will be “ON” for a scan time. And then, it will repeat the data transmission/receiving processes from the first data again.
- When X20=“ON” → “OFF”, the instruction will be stopped and the data transmission/receiving will be discontinued immediately.
- The MBUS instruction is for the Master PLC, it can be used once only and do not use the LINK or RS instruction in the program.
- For avoid the corresponding breakup, when the MBUS instruction sends a communication request to a particular Slave, if the respondent time of the Slave exceeds the Time-out duration (designated by D9129), the MBUS instruction will stops communication from the specific Slave and operates next communication command.
- The setting value of the Time-out duration is restored in D9129. The Time-out duration = (the content value of D9129) ×10ms. When D9129=0 (the default value), the Time-out duration is 100 ms.
- Most of the applied situation is not necessary to change the Time-out duration. But, if an equipment in the communication link, its response is very slow, then the longer Time-out duration is necessary.
- The attributes of the devices designated in a data transmission/receiving operation should be the same. For example, if the device designated by the Master station is a bit device, then the designated device of the Slave station should be also a bit device.

- The instruction working area headed with S2 :

S2		Description
D100	Lower 8 bits	The Slave station ID number when a communication error occurs
	Upper 8 bits	Instruction working status 0: Normal data transmission/receiving 2: Error of the length of the transferred/received data (unequal to 1 ~ 64) 4: Error of the designated device type 5: Error of the designated device ID number 6: The characteristic of devices between the Master and Slave stations are different A: Normal communications but no response from Slave stations B: Abnormal communications
D101 ~ D103	The working area required when the instruction is performed	

- Edit Communication Table
  In addition to the File Registers' layout function; and further, the Ladder Master provides more user friendly and easily of data input interface, it provide the user to create and edit the Communication Table List.
  Please select the Ladder Master's "Tools ---- Edit Communication Table ...." function to start the Communication Table List document edit window. By the interlocutory pop-up window, user can easily create and edit the communication table step-by-step. After the Communication Table has been finished, the contents will become a part of the user program. The communication commands in the table will go with the user program and keep in VH PLC's system process area. And also, this function provides user to retrieve, access and edit the Communication Table.

- Communication Table Example:

```
M9000
-| |----[MBUS  D0  D100]
```

Instruction: MBUS ▼ Length of Reg: 22

Number	Command	Master Data		Slave ID	Slave Data Type	Slave Data #	Length	Word / Bit
1	Read	D200	<--	5	4	0	10	W
2	Write	D210	-->	2	4	1000	5	W
3	Single Write	D215	-->	3	4	2000	1	W

This example is for communication table to execute 3 data receiving/transmitting operations.

(1) To read the data in 40000 ~ 40009 of Slave station #5 and put they to D200 ~ D209 of the Master station.

(2) To write the data in D210 ~ D214 of the Master station into 41000 ~ 41004 of Slave station #2

(3) To write the data in D215 of the Master station into 42000 of Slave station #3.

The "Slave Data Type" and "Slave Data No." in the communication table refers to the component ID number of the slave station equipment.
For example, there is a MODBUS component:

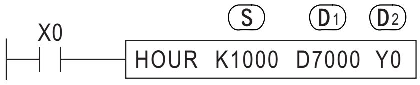
4 0 0 0 0

- 0000 — The component data ID No.
- 4 — The component data type
  - 0:Writable & Readable Bit Component
  - 1:Read Only Bit Component
  - 3:Read Only Data Register (16 bits)
  - 4:Writable & Readable Register (16 bits) , the most often type.

D	FNC 169 HOUR		⊣⊢ D HOUR (S) (D1) (D2)	Hour Meter	M	VB	VH
						○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○	○		○	○	○
D_1											○	○				○
D_2		○	○	○												○

- 16-bit instruction, D_1 occupies 2 consecutive devices; 32-bit instruction, D_1 occupies 3 consecutive devices



S : The period of time in which D2 will turns “ON” (the unit of (S) is hour)

D_1 : The current value of the time meter (the unit of (D1) is hour)

D_2 : The output device of the time meter

- This instruction is used one hour as the unit of timer.
- The timer counts the time by up counting clock pulse. When the current value of the time meter (D1) ≥ the period setting value (S), the contact of the time meter (D2) = “ON”.
- The real setting period of the time meter = One hour * the setting value of (S).
- (D1) stores integer number of the current value (in hours); The register next to (D1) stores the current value which is less than 1 hour (in seconds).
- As the diagram above

  When X0= “ON”, the current value of the register (D1) will begin to do the cumulatively up counting (hourly). If the current value of D7000=K1000 (1000 hours), the contact of output device Y0= “ON”.

  When X0= “OFF”, this instruction will provides retentive function for the current value of time meter, the current value of register D7000 will be retain.
- Mostly, this instruction is used to monitor the lifespan of a component or to remind the regularly maintenance. For retain the register's current value of time meter during power failure, please assign (D1) to a latched register. If assign (D1) to a general register, when the power failure or the PLC states “STOP” → “RUN”, the content value of (D1) will reset to “0”.
- After the output device of time meter (D2) = “ON”, the current value of time meter (D1) will continuously execute the up counting.
- When the current value of time meter (D1) reaches the maximum value of a 16-bit or 32-bit register, the counting will be stopped.
- If the PLC is “RUN” and the (D1) > = (S) , the output of (D2) will be “ON”.

D	FNC 250 SCL	P	─┤├──[DSCLP S1 S2 D]	Scaling (Translated by Coordinate)	M	VB	VH
						○	
D	FNC 251 SCL2	P	─┤├──[DSCL2P S1 S2 D]	Scaling II (Translated by Coordinate)	M	VB	VH
						○	

Operand	Devices															
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S1					○	○	○	○	○	○	○	○		○	○	○
S2											○					○
D						○	○	○	○	○	○	○		○		○

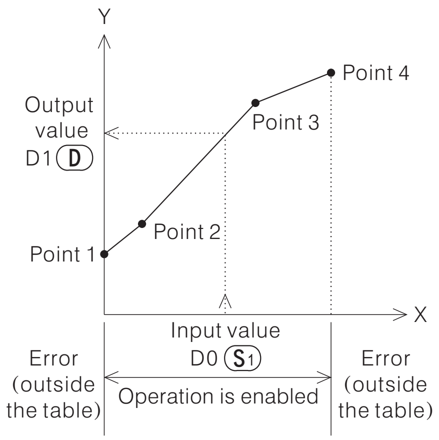
```
      X0              S1   S2    D
├──┤ ├──────[ SCL  D0  D1000  D1 ]
```

S1 : the source value (device) of scaling (X coordinate)

S2 : the head device of the conversion data table

D : the device stores scaling output value (Y coordinate)

- These instructions will be valid if a VB series V1.70 or later is used.
- To perform this instruction, the input value that specified in S1 (X coordinate) is processed by scaling for the specifically conversion characteristics and get the result (Y coordinate) stored into a specified device in D. The characteristics of scaling is assigned by the conversion data table that is storage at specified devices S2 and later.
- When X0= "ON" , this instruction uses the value at D0 and the conversion data table (which is started from D1000) to execute translate then put the scaling result into D1.



S2 Conversion data table setting (for 16-bit scaling)

Coordinate #		S2	D1000*
Point 1	X_1	S2+1	D1001
	Y_1	S2+2	D1002
Point 2	X_2	S2+3	D1003
	Y_2	S2+4	D1004
Point 3	X_3	S2+5	D1005
	Y_3	S2+6	D1006
Point 4	X_4	S2+7	D1007
	Y_4	S2+8	D1008

S2 Conversion data table setting (for 32-bit scaling)

Coordinate #		S2+1, S2
Point 1	X_1	S2+3, S2+2
	Y_1	S2+5, S2+4
Point 2	X_2	S2+7, S2+6
	Y_2	S2+9, S2+8
Point 3	X_3	S2+11, S2+10
	Y_3	S2+13, S2+12
Point 4	X_4	S2+15, S2+14
	Y_4	S2+17, S2+16

* D1000 = K4 in this case, shown in the left figure

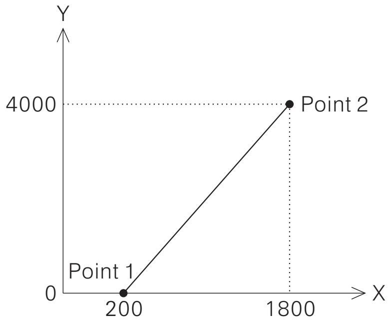
- This instruction could easily transfer between the value of analog I/O and the quantity under expected unit (Ex: weight, distance... etc.)
- If the output data is not an integer, it will be rounded off to an integer result.
- If in the conversion table have two or more points at the same X coordinate and the input value S1 is equal to this value, the value of second Y coordinate will output to the D .
- An operation error is caused in the following cases; The error flag M9067 turns "ON", and the error code K6706 is stored in D9067.
  ① When the data of Xn is not set by the ascending order in the table (Xn+1 smaller than Xn)
  ② When S1 is outside the data table
  ③ When the differential value between contiguous points (including X or Y coordinate) exceeds the 16-bit data range (K65535)
  ④ When the number of coordinate points at S2 is < K1.

- Example: To get and translate the positional data from a Linear Potential-Meter

  In this case, a 500 mm stroke Linear Potential-Meter and a VB-4AD analog input module work together to measure the current position. Its adjustable position will lie in between 50 to 450 mm. Then, use 0.1 mm as a unit to display the moving distance 0.0 ~ 400.0 mm.

  Since a Linear Potential-Meter will transfer the position of 0 ~ 500 mm into 0 ~ 10 V potential and output to a VB-4AD then could get a number that is between 0 to 2000. By those characteristics, the possible position at the machine is 50 ~ 450 mm will get a data of 200 ~ 1800.

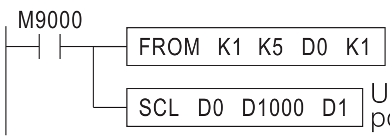
  By the plan above to work out a conversion chart and table below.



The conversion data table of SCL instruction

	Item		Registers #	Content value
S2	Number of coordinate points		D1000	2
S2 +1	Point 1	X_1	D1001	200
S2 +2		Y_1	D1002	0
S2 +3	Point 2	X_2	D1003	1800
S2 +4		Y_2	D1004	4000

To set the input value into the X coordinate
Then the Y coordinate is the output value that represents the current position (unit: 0.1 mm)



Let the output of a Linear Potential-Meter connect to a VB-4AD and the read value store at D0

Use the SCL instruction and the content of D0 to get the current position (unit:0.1 mm) and store at D1

- The SCL2 instruction having the same function but uses different configuration of data table.
  There is the configuration of the conversion data table for the SCL2 instruction below. (The number of coordinate points for this example is K4.)

For 16-bit SCL2 instruction

Number of coordinate points		S2
X coordinate	X_1	S2 +1
	X_2	S2 +2
	X_3	S2 +3
	X_4	S2 +4
Y coordinate	Y_1	S2 +5
	Y_2	S2 +6
	Y_3	S2 +7
	Y_4	S2 +8

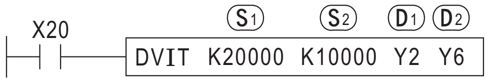
For 32-bit SCL2 instruction

Number of coordinate points		S2 +1, S2
X coordinate	X_1	S2 +3, S2 +2
	X_2	S2 +5, S2 +4
	X_3	S2 +7, S2 +6
	X_4	S2 +9, S2 +8
Y coordinate	Y_1	S2 +11, S2 +10
	Y_2	S2 +13, S2 +12
	Y_3	S2 +15, S2 +14
	Y_4	S2 +17, S2 +16

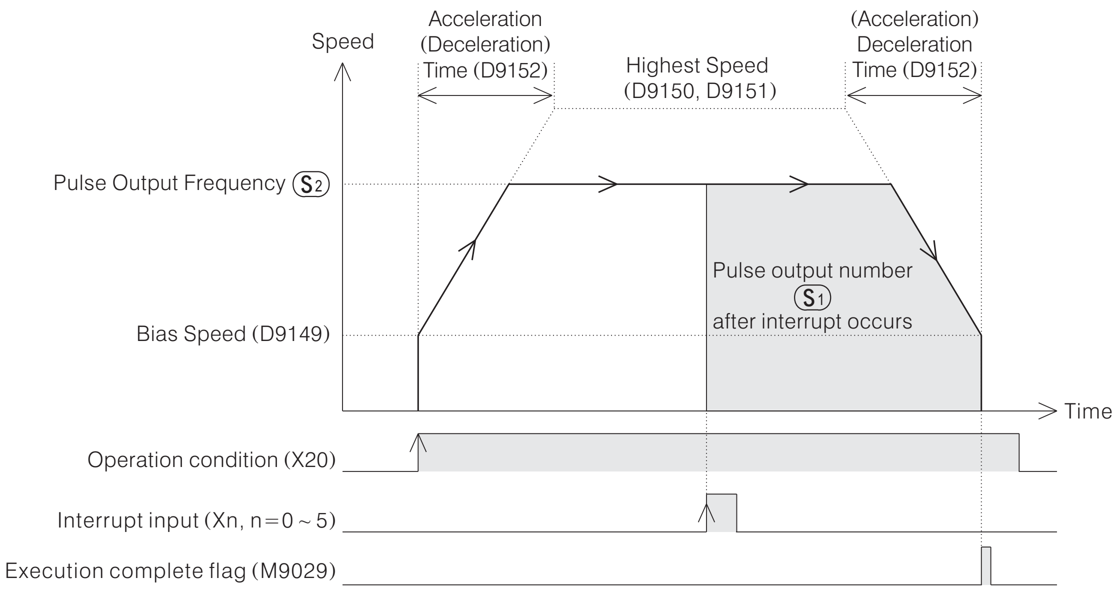
D	FNC 151 DVIT		DDVIT S1 S2 D1 D2	One-speed Interrupt Constant Quantity Feed	M	VB1	VH
						○	

Operand	Devices														
	X	Y	M	S	K*n*X	K*n*Y	K*n*M	K*n*S	T	C	D	SD	P	V,Z	K,H	VZ index
S_1					○	○	○	○	○	○	○	○		○	○	○
S_2					○	○	○	○	○	○	○	○		○	○	○
D_1		○														○
D_2		○	○	○												○

- D_1 = Y0 ~ Y3
- 16-bit instruction, S_1 = −32,768 ~ 32,767 (S1≠0);
  32-bit instruction, S_1 = −2,147,483,648 ~ 2,147,483,647 (S1≠0)
- When D_1=Y0 or Y1, S_2 = 10 ~ 20,000
- When D_1=Y2 or Y3, 16-bit instruction, S_2 = 10 ~ 32,767; 32-bit instruction, S_2 = 10 ~ 200,000



S_1 : pulse output number after interrupt occurs
S_2 : pulse output frequency
D_1 : pulse output point
D_2 : direction signal output point



- This instruction will be valid if a VB1 series V1.72 or later is used.
- When X20 = "ON", the Y2 generates 10 KHz pulses continuously. When the first interrupt condition occurs, will reset the D9145, D9144 (CV of Y2) and the Y2 generates specified 20, 000 pulses (10 KHz) then stop. The Y6 = ON if it's a positive(forword) rotation.
- When X20 = "OFF" → "ON", the DVIT instruction decides the one-speed interrupt constant quantity feed position control, it is according to the D9148 interruption devices allocation, D9149 Bias Speed, (D9151, D9150) Highest Speed (≦ 20, 000 if D1 = Y0 or Y1) , D9152 Acc./Dec. time, M9141 ~ M9144 interrupt signal logic reverse flags, S1 and S2 .
  During it is executing, to change any parameter will be ineffective. So, must finish all the parameters (D9148 ~ D9152 and M9141 ~ M9144) before it starts.
- Every pulse output point has its own interrupt signal logic reverse flag. By the individual status of M9141 ~ M9144 to assign which is the normal or reverse interrupt logic of Y0 ~ Y3.
  If its flag is "OFF" (normal edge logic), turning "ON" (OFF → ON) the input will accept the interrupt.
  If its flag is "ON" (reverse edge logic), turning "OFF" (ON → OFF) the input will accept the interrupt.

Pulse output point	D1 =Y0	D1 =Y1	D1 =Y2	D1 =Y3
Interrupt signal logic reverse flag	M9141	M9142	M9143	M9144

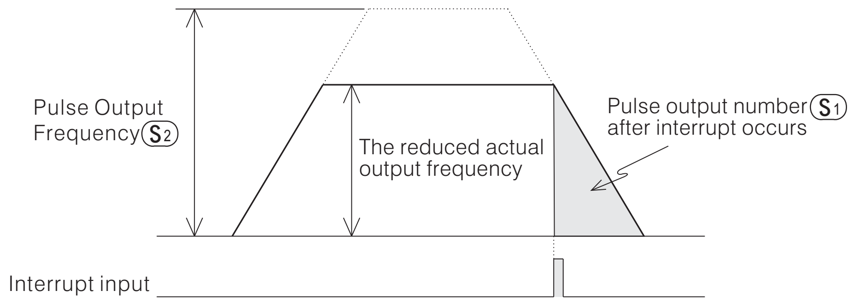
- To select the input point of interrupt signal is by the contents of the D9148. (Default value is H3210)

D9148=H○○○○ (contents setting by hexadecimal method) ＊Ex.: D9148 = H5421

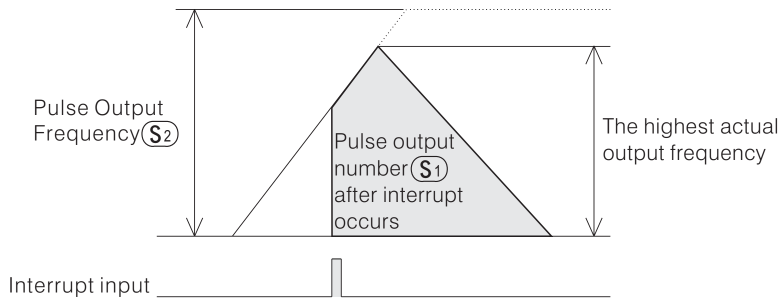
1st digit (right)	For the interrupt input point of Y0	The interrupt input point of Y0 is X1
2nd digit	For the interrupt input point of Y1	The interrupt input point of Y1 is X2
3rd digit	For the interrupt input point of Y2	The interrupt input point of Y2 is X4
4th digit (left)	For the interrupt input point of Y3	The interrupt input point of Y3 is X5

Contents of ○	Description of setting	Available input component and notes
0	To assign the X0 as the interrupt input point	• The interrupt input point can use X0~X5, otherwise the instruction wouldn't execute correctly.
1	To assign the X1 as the interrupt input point	
≀	≀	• Any two of the interrupt input points at D9148 can't assign to the same point.
5	To assign the X5 as the interrupt input point	

- After the interrupt occurs and output point generates the specified S1 number of pulses, the operation will be stopped and the Execution Complete flag M9029 will be "ON" for a scan time.
- When the condition contact X20 turns "OFF" during the pulse output, the operation will be decelerated to stop , but the Execution Complete flag M9029 will not take action then.
- Please check the pulse output monitor flag (M9149 ~ M9152) of the output point D1 before running this instruction. If the corresponding flag signal is "ON", that means another pulse output instruction still using this D1 point then the instruction will not start.
- The D2 forward/reverse direction signal is decided by the +/– sign of pulse output number S1 's value.
  If the value of S1 > 0, it's a forward rotation. The D2 will "ON" and the value of CV registers will increase.
  If the value of S1 < 0, it's a reverse rotation. The D2 will "OFF" and the value of CV registers will decrease.
- If the D1 is assigned toY0 (or Y1) (its frequency is up to 20 KHz), the available range of S2 is 10 ~ 20, 000.
- If the D1 is assigned toY2 (or Y3) (its frequency is up to 200 KHz), the available range of S2 is 10 ~ 32,767 (16-bit operation) or 10 ~ 200,000 (32-bit operation).
- If the specified S1 is not large enough to finish its deceleration (related to S2 and D9152), then the actual highest speed will be reduced (< S2). So, via the slower speed output that can smoothly slow down and stop within the specified number of pulses.



- If【 S1 < (the required pulse number for acceleration + deceleration) 】and the interruption condition turns "ON" during the acceleration, the decelerating operation may start earlier (before the output speed reach the highest setting) as the figure below.



- If to execute the DVIT instruction but its interrupt condition is"ON" already, the operation of this DVIT instruction will be performed in the same way as the DRVI instruction.
- Must input the interruption signal before the number of pulse output is increased to 4,294,967,296. If the number is reached (without its interrupt), the operation will be stopped and the Execution Complete flag (M9029) will be turned "ON" once.

D	FNC 153 LIR		DLIR (S) (D)	Relatively Linear Interpolation	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D															○	
• 16-bit instruction, S occupies 9 consecutive registers; 32-bit instruction, S occupies 18 consecutive registers																
• D = K0 or K1 (if D = K0, occupies Y0, Y1, Y4 and Y5; if D = K1, occupies Y2, Y3, Y6 and Y7)																

X20 — LIR D1000 K1 (S) (D)

S : the head register of the parameter data table

D : output points setting parameter

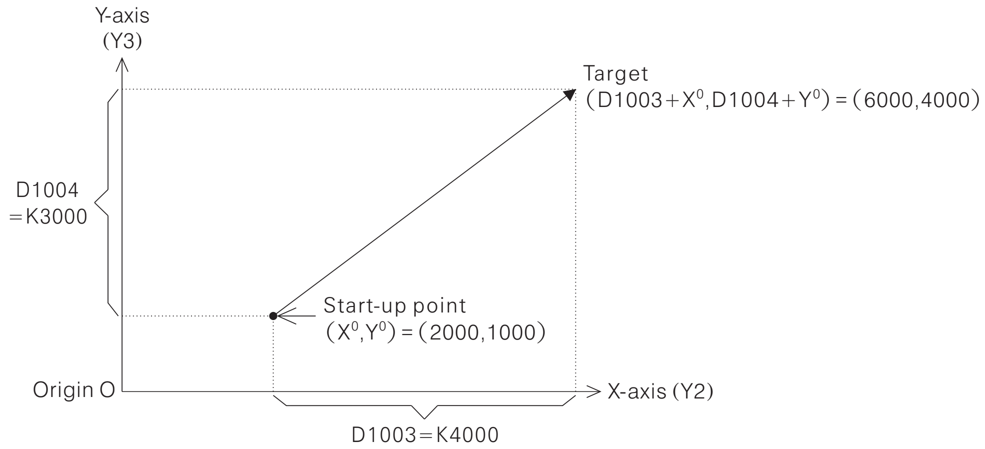
- This instruction will be valid if a VB1 series V1.72 or later is used.
- The LIR instruction simultaneously controls with two axes by two pulse and two direction output points to move the position at an X-Y table. The parameters are two composite speeds, the Acceleration/Deceleration time and two individual target values.
- When X20 = "ON", simultaneously the Y2 (X-axis) and Y3 (Y-axis) output points simultaneous generate pulses also the Y6 (X-axis) and Y7 (Y-axis) output moving direction signals. They separately output signals by using the parameters of D1000 (Composite Initial speed), D1001 (Composite Maximum speed), and D1002 (Acceleration/Deceleration time). The location in a coordinate (X,Y) is from the start-up point (X^0,Y^0) to the target point (D1003+X^0, D1004+Y^0)
- The definition list about the output points parameter(D):

Content value of (D)	The pulse output point of X-axis (the CP of X-axis)	The pulse output point of Y-axis (the CP of Y-axis)	The direction signal output of X-axis	The direction signal output of Y-axis	Pulse output stop control coil
K0	Y0 (D9141,D9140)	Y1 (D9143,D9142)	Y4	Y5	If either one of the M9145 or M9146 is "ON", both axes stop output pulses.
K1	Y2 (D9145,D9144)	Y3 (D9147,D9146)	Y6	Y7	If either one of the M9147 or M9148 is "ON", both axes stop output pulses.

- The parameter data table of block(S):

16-bit instruction	32-bit instruction	Description and available setting range	Fill up method
(S)	(S)+1 、(S)	Composite Initial speed [(D)= K0, 0 ~ 20,000 (Hz);(D)= K1, 0 ~ 32,767 (Hz)]	Designated by user program or communication
(S)+1	(S)+3 、(S)+2	Composite Maximum speed [16-bit:(D)=K0,10~20,000 (Hz);(D)=K1,10~32,767 (Hz); 32-bit: 10~200,000 (Hz)]	
(S)+2	(S)+5 、(S)+4	Acceleration/Deceleration time [0 ~ 5,000 (ms.)]	
(S)+3	(S)+7 、(S)+6	Target pulse number (X-axis) [16-bit: -32,768 ~ 32,767 (pulses); 32-bit: -2,147,483,648 ~ 2,147,483,647 (pulses)]	
(S)+4	(S)+9 、(S)+8	Target pulse number (Y-axis) [16-bit: -32,768 ~ 32,767 (pulses); 32-bit: -2,147,483,648 ~ 2,147,483,647 (pulses)]	
(S)+5	(S)+11 、(S)+10	Initial speed of X-axis (Hz)	Storage area of calculated results, they are produced by the executed LIR instruction
(S)+6	(S)+13 、(S)+12	Initial speed of Y-axis (Hz)	
(S)+7	(S)+15 、(S)+14	Maximum speed of X-axis (Hz)	
(S)+8	(S)+17 、(S)+16	Maximum speed of Y-axis (Hz)	

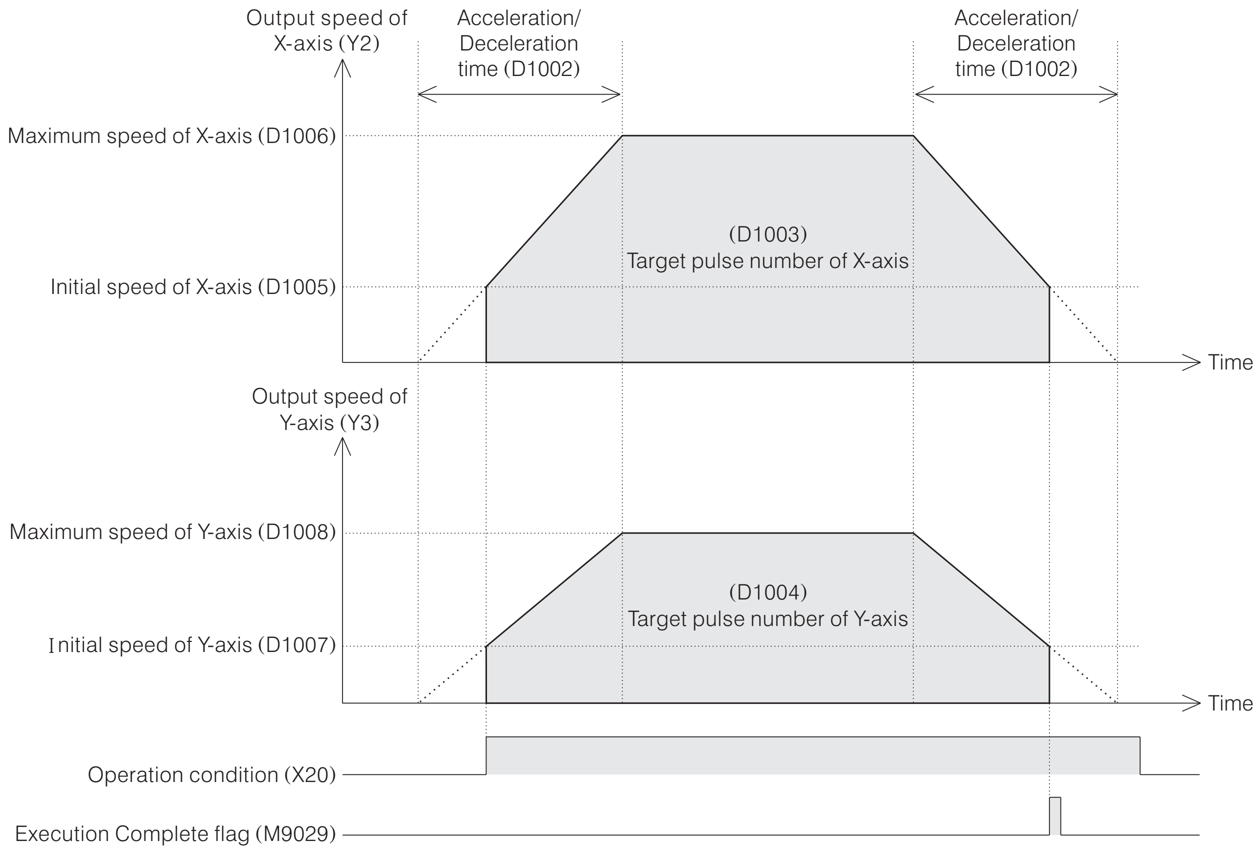
- For example: Before the instruction executes, the start-up current position CP is (X^0,Y^0) = (2000,1000), D1000 = K1000 (Composite Initial speed Hz), D1001 = K3000 (Composite Maximum speed Hz), D1002 = K300 (Acceleration/Deceleration time ms.), D1003 = K4000 (X-axis target pulse number) and D1004 = K3000 (Y-axis target pulse number). When X20 = “OFF”→“ON”, the LIR instruction will compute the further parameters and fill D1005 ~ D1008 up then all the X–axis, Y–axis and direction signals start to output. Those will follow the parameter data table and move the positioning path from (2000,1000) to (6000,4000) as shown below.



The formulas to get the D1005 ~ D1008 and relationship diagram between X-axis, Y-axis and timing:

$$D1005 = D1000 \times \frac{(D1003)}{\sqrt{(D1003)^2 + (D1004)^2}} \; ; \; D1007 = D1000 \times \frac{(D1004)}{\sqrt{(D1003)^2 + (D1004)^2}}$$

$$D1006 = D1001 \times \frac{(D1003)}{\sqrt{(D1003)^2 + (D1004)^2}} \; ; \; D1008 = D1001 \times \frac{(D1004)}{\sqrt{(D1003)^2 + (D1004)^2}}$$



- During this instruction is in execute, to change its parameter will be ineffective. So, must finish all the configuration of correlative parameters (data table S) before this instruction is executed.
- When the positioning target is reached, the operation will be stopped and the Execution Complete flag M9029 will be "ON" for a scan time.
- When the condition contact X20 turns "OFF" during the pulse outputs, the operations will be decelerated to stop, but the Execution Complete flag M9029 will not take action then.
- When the setting of D is K0 and any one of the M9145, M9146 turns "ON" during the pulse outputs, both of the Y0 (X-axis) and Y1 (Y-axis) immediately stop pulse outputs, but the Execution Complete flag M9029 will not take action then; When the setting of D is K1 and any one of the M9147, M9148 turns "ON" during the pulse outputs, both the Y2 (X-axis) and Y3 (Y-axis) immediately stop pulse outputs, but the Execution Complete flag M9029 will not take action then.
- Please check the pulse output monitor flags (M9149 ~ M9152) of D 's related output points before running this instruction. If any one of the corresponding flag signal is "ON" (M9149 or M9150 for D =K0;M9151 or M9152 for D = K1), that means another pulse output instruction still using the point(s) then the instruction will not start.
- For every single axis, its forward/reverse direction signal is decided by the positive/negative sign of the target pulse number.
  If the axis's target pulse number ≥ 0, that is a forward rotation. The direction signal is "ON" and the value of the current value registers will be increased.
  If the axis's target pulse number < 0, that is a reverse rotation. The direction signal is "OFF" and the value of the current value registers will be decreased.
- The Composite Initial speed must be equal to or less than the Composite Maximum speed.
- Since the output frequency rate of Y0 or Y1 is 20 KHz at the most, when the content value of D is K0, the configuration range of the Composite Initial speed is 0 ~ 20,000 (Hz) and the Composite Maximum speed is 10 ~ 20,000 (Hz).
- Since the output frequency rate of Y2 or Y3 is 200 KHz at the most, when the content value of D is K1, the configuration range of the Composite Initial speed is 0 ~ 32,767 (Hz) and the Composite Maximum speed is 10 ~ 32,767 (Hz) for 16-bit or 10 ~ 200,000 (Hz) for 32-bit.
- If the calculated result of maximum speed is less than 1 (Hz), the axis will not generate a pulse.
- The content values of D9149 ~ D9152 will not affect the pulse output of this instruction.
- If both of the pulse output numbers are equal to 0, this instruction will not execute.

D	FNC 154 LIA		DLIA (S) (D)	Absolutely Linear Interpolation	M	VB1	VH
						○	

Operand	Devices															
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S											○					○
D															○	
● 16-bit instruction, S occupies 9 consecutive registers; 32-bit instruction, S occupies 18 consecutive registers																
● D = K0 or K1 (if D = K0, occupies Y0, Y1, Y4 and Y5; if D = K1, occupies Y2, Y3, Y6 and Y7)																

```
        (S)    (D)
X20
─┤ ├──[LIA  D1000  K1]
```

S : the head register of the parameter data table

D : output points setting parameter

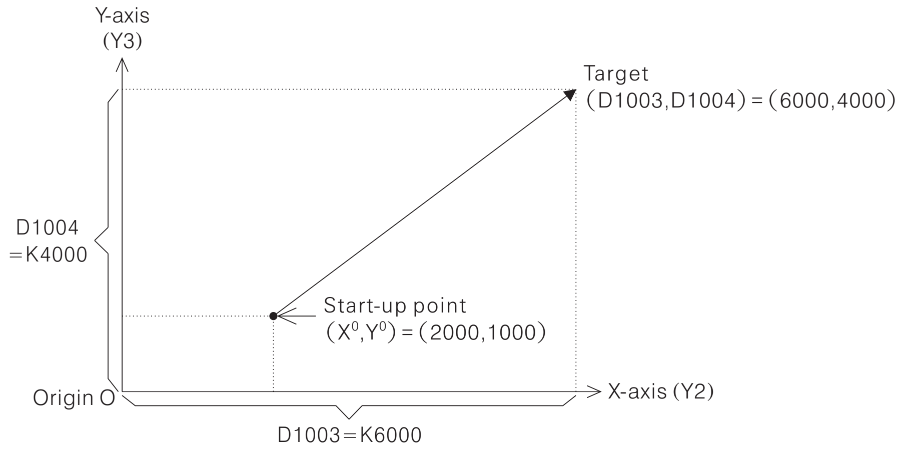
- This instruction will be valid if a VB1 series V1.72 or later is used.
- The LIA instruction simultaneously controls with two axes by two pulse and two direction output points to move the position at an X-Y table. The parameters are two composite speeds, the Acceleration/Deceleration time and two individual target points.
- When X20 = "ON", simultaneously the Y2 (X-axis) and Y3 (Y-axis) output points simultaneous generate pulses also the Y6 (X-axis) and Y7 (Y-axis) output moving direction signals. They separately output signals by using the parameters of D1000 (Composite Initial speed), D1001 (Composite Maximum speed), and D1002 (Acceleration/Deceleration time). The location in a coordinate (X,Y) is from the start-up point (X^0,Y^0) to the absolutely target point (D1003, D1004).
- The definition list about the output points parameter (D):

Content value of (D)	The pulse output point of X-axis (the CP of X-axis)	The pulse output point of Y-axis (the CP of Y-axis)	The direction signal output of X-axis	The direction signal output of Y-axis	Pulse output stop control coil
K0	Y0 (D9141,D9140)	Y1 (D9143,D9142)	Y4	Y5	Both X and Y axes stop pulse outputs if either one of the M9145 or M9146 is "ON"
K1	Y2 (D9145,D9144)	Y3 (D9147,D9146)	Y6	Y7	Both X and Y axes stop pulse outputs if either one of the M9147 or M9148 is "ON"

- The parameter data table of block (S):

16-bit instruction	32-bit instruction	Description and available setting range	Fill up method
(S)	(S)+1 、(S)	Composite Initial speed [(D)= K0, 0 ~ 20,000 (Hz);(D)= K1, 0 ~ 32,767 (Hz)]	Designated by user program or communication
(S)+1	(S)+3 、(S)+2	Composite Maximum speed [16-bit:(D)=K0,10~20,000 (Hz);(D)=K1,10~32,767 (Hz); 32-bit: 10~200,000 (Hz)]	
(S)+2	(S)+5 、(S)+4	Acceleration/Deceleration time [0 ~ 5,000 (ms.)]	
(S)+3	(S)+7 、(S)+6	Target point at X-axis [16-bit: -32,768 ~ 32,767 (pulses); 32-bit: -2,147,483,648 ~ 2,147,483,647 (pulses)]	
(S)+4	(S)+9 、(S)+8	Target point at Y-axis [16-bit: -32,768 ~ 32,767 (pulses); 32-bit: -2,147,483,648 ~ 2,147,483,647 (pulses)]	
(S)+5	(S)+11 、(S)+10	Initial speed of X-axis (Hz)	Storage area of calculated results, they are produced by the executed LIA instruction
(S)+6	(S)+13 、(S)+12	Initial speed of Y-axis (Hz)	
(S)+7	(S)+15 、(S)+14	Maximum speed of X-axis (Hz)	
(S)+8	(S)+17 、(S)+16	Maximum speed of Y-axis (Hz)	

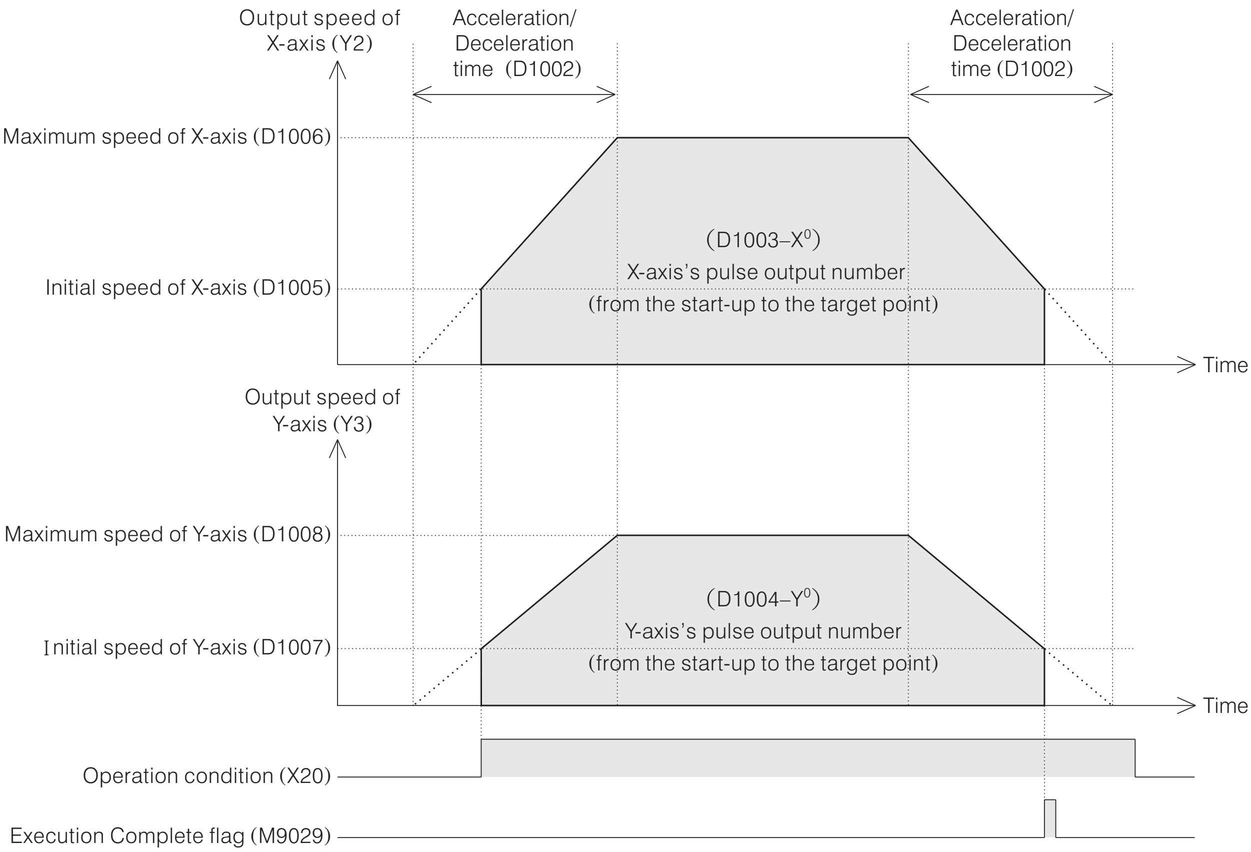
• For example: Before the instruction executes, the start-up current position CP is (X^0,Y^0) = (2000,1000), D1000 = K1000 (Composite Initial speed Hz), D1001 = K3000 (Composite Maximum speed Hz), D1002 = K300 (Acceleration/Deceleration time ms.), D1003 = K6000 (Target point at X-axis) and D1004 = K4000 (Target point at Y-axis). When X20 = "OFF"→"ON", the LIA instruction will compute the further parameters and fill D1005 ~ D1008 up then all the X-axis, Y-axis and direction signals start to output. Those will follow the parameter data table and move the positioning path from (2000,1000) to (6000,4000) as shown below.



The formulas to get the D1005 ~ D1008 and relationship diagram between X-axis, Y-axis and timing:

$$D1005 = D1000 \times \frac{(D1003-X^0)}{\sqrt{(D1003-X^0)^2+(D1004-Y^0)^2}} \ ; D1007 = D1000 \times \frac{(D1004-Y^0)}{\sqrt{(D1003-X^0)^2+(D1004-Y^0)^2}}$$

$$D1006 = D1001 \times \frac{(D1003-X^0)}{\sqrt{(D1003-X^0)^2+(D1004-Y^0)^2}} \ ; D1008 = D1001 \times \frac{(D1004-Y^0)}{\sqrt{(D1003-X^0)^2+(D1004-Y^0)^2}}$$



- During this instruction is in execute, to change its parameter will be ineffective. So, must finish all the configuration of correlative parameters (data table S) before this instruction is executed.
- When the positioning target is reached, the operation will be stopped and the Execution Complete flag M9029 will be "ON" for a scan time.
- When the condition contact X20 turns "OFF" during the pulse outputs, the operations will be decelerated to stop, but the Execution Complete flag M9029 will not take action then.
- When the setting of D is K0 and any one of the M9145, M9146 turns "ON" during the pulse outputs, both of the Y0 (X-axis) and Y1 (Y-axis) immediately stop pulse outputs, but the Execution Complete flag M9029 will not take action then; When the setting of D is K1 and any one of the M9147, M9148 turns "ON" during the pulse outputs, both the Y2 (X-axis) and Y3 (Y-axis) immediately stop pulse outputs, but the Execution Complete flag M9029 will not take action then.
- Please check the pulse output monitor flags (M9149 ~ M9152) of D 's related output points before running this instruction. If any one of the corresponding flag signal is "ON" (M9149 or M9150 for D = K0;M9151 or M9152 for D = K1), that means another pulse output instruction still using the point(s) then the instruction will not start.
- For every single axis, its forward/reverse direction signal is decided by the positive/negative sign of the pulse output number (to subtract the start-up point from the target point).
  If the axis's pulse output number $\geq$ 0, that is a forward rotation. The direction signal is "ON" and the value of the current value registers will be increased.
  If the axis's pulse output number $<$ 0, that is a reverse rotation. The direction signal is "OFF" and the value of the current value registers will be decreased.
- The Composite Initial speed must be equal to or less than the Composite Maximum speed.
- Since the output frequency rate of Y0 or Y1 is 20 KHz at the most, when the content value of D is K0, the configuration range of the Composite Initial speed is 0 ~ 20,000 (Hz) and the Composite Maximum speed is 10 ~ 20,000 (Hz).
- Since the output frequency rate of Y2 or Y3 is 200 KHz at the most, when the content value of D is K1, the configuration range of the Composite Initial speed is 0 ~ 32,767 (Hz) and the Composite Maximum speed is 10 ~ 32,767 (Hz) for 16-bit or 10 ~ 200,000 (Hz) for 32-bit.
- If the calculated result of maximum speed is less than 1 (Hz), the axis will not generate a pulse.
- The content values of D9149 ~ D9152 will not affect the pulse output of this instruction.
- If both of the pulse output numbers are equal to 0, this instruction will not execute.

FNC 188 CRC	P	CRCP (S) (D) (n)	Cyclic Redundancy Check - 16	M	VB	VH
					○	

Operand	Devices														
	X	Y	M	S	KnX	KnY	KnM	KnS	T	C	D	SD	P	V,Z	K,H	VZ index
S					○	○	○	○	○	○	○					○
D									○	○	○					○
n											○				○	

- When S is designated to K*n*X, K*n*Y, K*n*M or K*n*S, it should be designated to K4X, K4Y, K4M or K4S.
- n = 1 ~ 256 • If the M9161 = "ON" (8-bit mode), the D occupies 2 consecutive points.

```
        (S) (D)   (n)
 X20
 |-| |----CRC D0 D100 K7
```

S : Head ID number of data source

D : Position where the CRC result is stored

n : Number of 8-bit data (Unit: byte)

- This instruction will be valid if a VB series V1.72 or later is used.
- To calculate the CRC-16 (Cyclic Redundancy Check) code for the content of (n) byte (8-bit) data headed with (S) , the result is stored in the designated device (D) .
- When the instruction is used for communication, the CRC-16 is applied to ensure and check the accuracy of the data transmission. The polynomial of the CRC-16 code: $X^{16}+X^{15}+X^{2}+1$.
- When X20 = "ON", it calculates 7 consecutive 8-bit data headed with D0, the CRC-16 code is stored in D100 (if the M9161 = "ON", two 8-bit codes are stored in D100 and D101).
- The instruction has two operation modes depending on the status of M9161:

M9161 = "OFF" (16-bit mode)

- This mode will separate the Upper 8 bits and Lower 8 bits of each device as two 8-bit data. The instruction uses (n) (= K7) 8-bit data (started by (S)) to calculate the CRC-16 code and stores to (D) (by a 16-bit value).

	Device	Content value	
(S)	D0 Lower 8 bits	H01	(n)=K7
	D0 Upper 8 bits	H03	
	D1 Lower 8 bits	H04	
	D1 Upper 8 bits	HED	
	D2 Lower 8 bits	H85	
	D2 Upper 8 bits	HA3	
	D3 Lower 8 bits	H28	
(D)	D100	H58A6	

M9161 = "ON" (8-bit mode)

- This mode will take the Lower 8 bits of each device as an 8-bit data (while ignore its Upper 8 bits). The instruction uses (n) (= K7) 8-bit data (started by (S)) to calculate the CRC-16 code and stores to (D) and (D) +1 (by two 8-bit values).

	Device	Content value	
(S)	D0 Lower 8 bits	H01	(n)=K7
	D1 Lower 8 bits	H03	
	D2 Lower 8 bits	H04	
	D3 Lower 8 bits	HED	
	D4 Lower 8 bits	H85	
	D5 Lower 8 bits	HA3	
	D6 Lower 8 bits	H28	
(D)	D100	HA6	
	D101	H58	