VS Series PLC Programming Manual

Version: 2.1　　　　　　Date: October, 2019

Preface

The purpose of this book is to give programming inspires for the VS series PLC. For the installation, wiring, maintenance and safety precautions of VS series PLC, please refer to the VS Series PLC Product Manual.

Name of Manual	Content
VS Series PLC Programming Manual (This book)	• Descriptions of VS series PLC components • Functions of basic instructions and application instruction • Precautions regarding programming
VS Series PLC Product Manual	• Introduction to VS series PLC • Environment, wiring and installation cautions of VS series PLC • Precautions of environment, wiring and installment • Instructions of optional devices

Table of Contents

1. Introduction of VS Series PLC

1-1 The Basic Concept of PLC Users

1-1-1 Introduction of PLC

The programmable Logic Controller (PLC) is an industrial computer control system that is easy to maintain, low cost and saves space. In addition to its programmable features, because it is used in the field of industrial control, PLC is also required to have high reliability and resistance to climate.

The concept of PLC was introduced by General Motors in 1968. It has been developed by decades. The initial PLC only had the simple logic control ability, and thus was named as a programmable logic controller. Since Today's PLC functions have covered the scopes of mathematical operations, servo control and communication links, the more correct name for it should be programmable controller (referred to as PC), which means a controller that is programmable. Due to the fact that it shares the same English abbreviation with the personal computer (referred to as PC). Therefore, in order to distinguish, the programmable controller is referred to as PLC.

PLC has micro controller at its core. The rapid development of semiconductor technology has substantially increased PLC's computing speed, program capacity and communication connectivity and decreased its prices. Coupled with the demands on high reliability and resistance to climate the design requirements, PLC is widely used in the field of automatic control, and has become an extremely important part.

1-1-2 Configuration of PLC

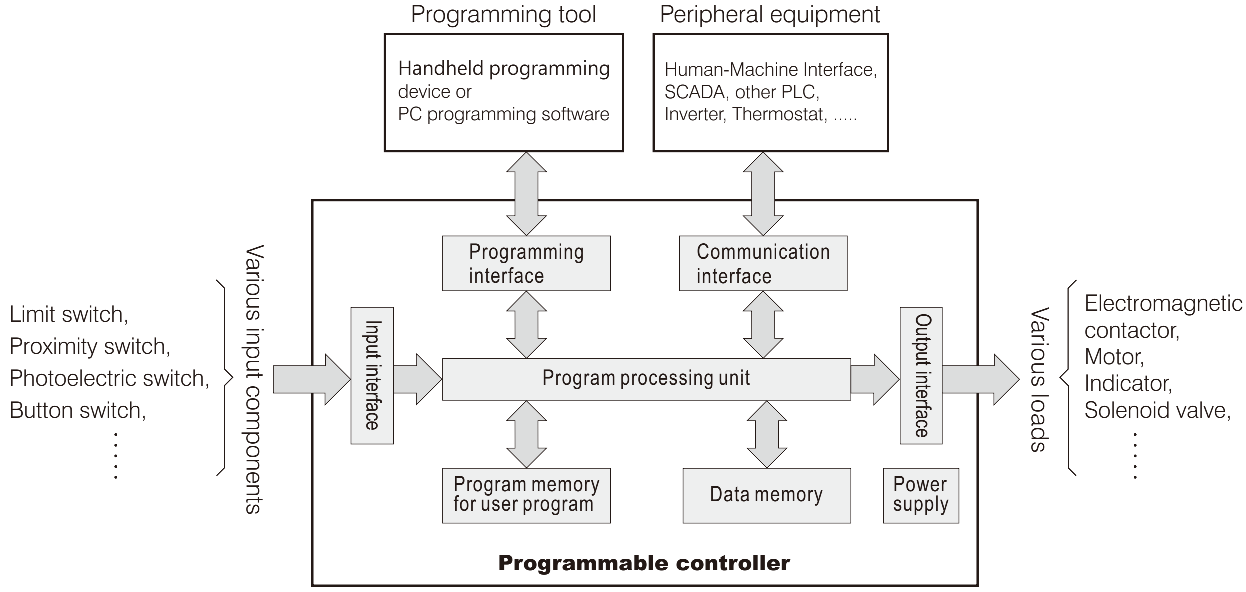


Fig.1-1 Processing Architecture of PLC

The configuration of programmable controller is shown in Fig. 1-1. Below describes the elements of a PLC briefly:

- Power supply —— Inside of a PLC consists of ordinary electronic circuit that must use low voltage DC power. Currently, a PLC manufacturer provides with the AC or DC power input model for the customer to choose from. The AC power model has a power supply circuit within the PLC to convert AC to DC. The DC power model needs to provide its power from an external DC source (usually DC 24V) then the PLC uses the internal DC to DC circuit to convert the power to supply the internal circuit require.

  In the past, user prefers to use the AC power input models. However, since there are many related equipments inside the control panel also required DC power, the use of an external DC power supply to provide all DC consumptions is popular in recent years.

  This is because the use of external DC power supply can avoid the event of unstable AC power input to damage the controller (this problem is easy to handle if an external DC power supply is used);This is because the use of external DC power supply can avoid the event of unstable AC power input to damage the controller (this problem is easy to handle if an external DC power supply is used); therefore, the use of external power supply increases the stability of control systems. On the other hand, it lowers the general cost of control systems. It is to kill two birds with one stone.

- Program processing unit —— The CPU is the core of a PLC, that is responsible for interpreting the user programs and calculating the results to achieve the purpose of control.

  Small PLCs usually use microcontrollers as the core. Now the semiconductor technology has been able to integrate CPU, memory and a variety of peripheral circuits into a SOC (System On Chip). This not only improves the overall performance, but also reduces costs. So now, even small PLCs use high-performance 32-bit CPU chip to produce controllers with fast execution, large program capacity and diverse functions, the real value for money.

- Program memory —— The memory for the PLC user to store the compiled control program, can let the CPU to interpret and compute then produces the control procedures.

  The program memory must have the ability to preserve its contents, in order to continue work after the power is resumption. At present, there are two ways.
  The first way is to use the SRAM (static random access memory) cooperated with a lithium battery. The advantages of this method are more flexible and less restrictive. But, the disadvantage is once the battery is exhausted, all the program and settings will disastrously disappear.
  The second method is to use non-volatile memory (such as EEPROM, Flash ROM, FRAM, etc.). By the no battery required characteristic of those semiconductor components to preserve the program. Its advantages are stable and reliable; but usually with higher cost, has the restriction on the number of writing times and slower in writing.
  A control system which is composed by a small PLC usually got less care from the owner. Therefore, it is a better choice to use battery-free non-volatile memory to store the user program.

- Data memory —— The data memory in PLC is responsible for storing the data.

  Both the operating system and the user program must have data workspace. Typically, SRAM (static random access memory) is used as the data work area.
  However, a part of the PLC data memory area must have latched feature to preserve the relevant parameter settings and specific operating results when power is missing. There are two components can cooperate with the SRAM to latch, one is the lithium battery, the other is the non-volatile (such as EEPROM, Flash ROM, FRAM, etc.) memory.
  Regarding their advantages and disadvantages, please refer to the description of program memory.

- Input interface —— The PLC reads external signals through its input interface to in order to operate as a reference for the program.

  In general, in order to prevent noise interference, the internal circuit of PLC inputs will isolate by photocouplers.
  The acceptable voltage of the input photocoupler is not high, so special attention should be paid to the external wiring. High voltage will damage the photocoupler and cause the input interface break down.

- Output interface —— The PLC sends the computed result to its output interface, then to drive components and reach the control action concretely.

  In order to avoid interference from noises, at the output interface of a PLC has the isolation circuit. Typically, the outputs are relays (by mechanically magnetic isolation) or transistors with photocouplers.
  The relay output is the dry contact of switch which with the advantage of intuitive no current direction limit and accepts AC or DC signal. The disadvantages are shorter life of contact, which is not a durable part and will affect the controller's life. Also, the movement of its contact is slow (about 5 ~ 10ms) which is not suited for the fast response output.
  The transistor is a semiconductor contactless switch, which has advantages and disadvantages just opposite to a relay.
  According to the characteristics of the control objectives, should choose an appropriate output type when designing the control system. In addition, for the purpose of the system reliability, may choose the transistor output type if it is possible.

- Programming interface —— The communication interface for to link the programming device with the PLC.

  The user uses the programming device to write a control program and through the programming interface to load to the PLC. Furthermore, monitoring and debugging to complete the program's calibration test.
  Generally, the programming interface is a serial communication interface, the RS-232, RS-422 or RS-485 port is popular. Recently there have been some PLC introduced with the USB interface to provide faster communication link.

- Communication interface —— An interface which is responsible for linking PLC with the surrounding equipment for data transmission.

  Nowadays, the common communication interfaces are the RS-232, RS-422 and RS-485, also the use of Ethernet is gradually increased.
  In addition to protocol with their own products, the PLC manufacturers usually provide communication protocol with MODICON's MODBUS to facilitate connecting with peripherals.
  With the increasing complexity of automatic control systems, PLC manufacturers also pay attention to the communication capacity to link with more peripherals. Therefore, higher level of PLC can provide several communication interfaces to make it work with more peripherals and complete complex control demands.

- Programming device —— The PLC manufacturer provides a device for users to program and debug.

  Common programming devices include handheld programming device and PC programming software.
  The handheld programming device usually uses the simple-text of Instruction List (IL), in fact it is an important part in the programming development of a PLC. However, with the progress of the times and the popularity of personal computers. The software at PC which can provide graphical interface and has become the best choice for PLC programming.

- Peripheral equipment —— Various devices are linked with the PLC to complete the control system together.

  A PLC is often linked with peripheral equipments, such as the computer, human-machine interface (HMI), inverter, thermostat, servo drive, power meter and other PLC.

  The most commonly peripheral equipment which PLC operates with are:
  1. A computer or human-machine interface connected to a PLC for machine settings and monitoring
  2. A computer or human-machine interface connected with multiple PLCs that becomes a local network team and its monitor.
  3. Linking several PLCs for proceeding decentralized control.
  4 Linking PLC to a variety of peripheral equipment of specific use, for extending control areas.

1-1-3 PLC Operation and Scan Time

The PLC systems use microcomputer technology to achieve the purpose of simulating traditional relay control panel. The microcomputer first reads the external inputs and then sequentially scans and executes the user program, so as to calculate the control result that the user wants to achieve. Finally, it outputs the result to drive the external loads and performs the real control action. The PLC execution order is shown in Fig. 1-2.

The PLC will sequentially execute the cycle (① receive external inputs; ② process the user program; ③ output the computed results). Then, the time spent for one cycle is called a “Scan Time”.

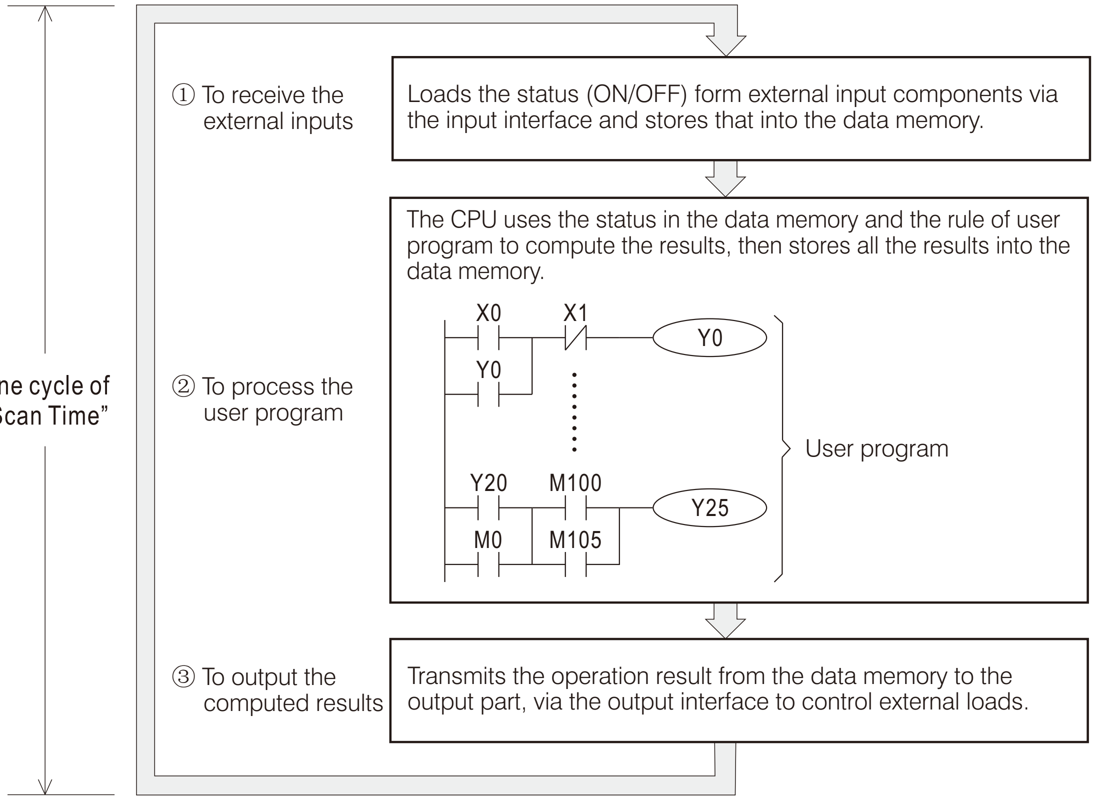


Fig. 1-2 PLC operation order and scan time

The PLC processes user program by the Sequential Scan (in the Ladder Diagram scans from left to right then from top to bottom), as shown in Fig. 1-3. Therefore, must pay extra attention to the procedural order of the program.

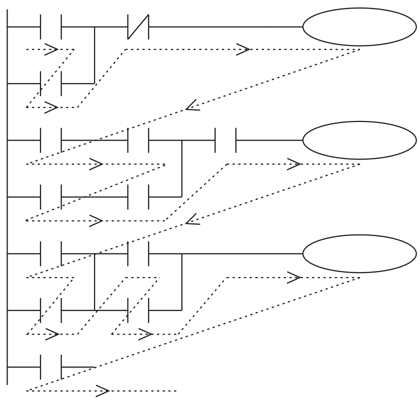


Fig. 1-3 PLC makes the sequential scan at the Ladder Diagram

The “Scan Time” and “Sequential Scan” are the basic and the most important concepts. A PLC user must fully understand those meanings, and pay attention about the influences when designing the PLC program.

1-1-4 PLC Input Signals

PLC's input endpoint is a window for PLC to accept control signals from outside, and is used to interface with a variety of switches and sensing elements. In recent years, the PLC functions have been tending to more developed diversification, detecting elements connected with the input points are also more diverse. Therefore, it is necessary for users to further understand the interface of PLC input point. Here are a few things to note:

1. As the PLC's work environment is often filled with a variety of noises and interfering sources, in order to work properly, photocouplers are often used at input endpoint to isolate noises.
2. The reaction speeds of different photocouplers are varied. The High-speed photocouplers can transmit signals faster, but the cost is higher. Therefore, high-speed photocouplers are often used at the input points of PLC which need high-speed responses. In the rest of case, general photocouplers are used.
3. In order to prevent noise interferences, the input circuit of a PLC in addition to photocouplers, also added with filters. Filters are divided into analog filters (composed of RC circuit) and digital filters (filter time is adjustable).
   According to functional requirements the PLC inputs can be divided into high-speed and general inputs. The general inputs are usually used in receiving signals which are not too fast, and often connected to the mechanical switchs (signal may have bounce in action). Thus, approximately 10ms filter circuits are added with in the general inputs.
   The high-speed inputs are usually designed for multiple functional inputs, those can be used as general inputs (need longer filter time) and can also process high-speed signals (need tiny filter time). Therefore, digital filters are used with high-speed inputs, they adjust filter time according to different needs (higher speed does not always mean better).
4. There is a time delay between the PLC's CPU received and the external signal input. The delay time is accumulated with the aforementioned photocoupler and added filter.
5. In order to meet multiple requirements, the current PLCs are usually designed with some high-speed input endpoints to execute some functions which require high speeds, such as high-speed counting, external interruption, pulse capture, frequency meter and pulse measurement.
   These signals usually require rapid responses and are relatively susceptible to noises. Therefore, in use, extra attention should be paid to its wiring. To avoid interfering sources, isolation cables can be used to avoid any interference.
6. There is certain amount of drive current at the PLC input endpoint, when using sensing elements, one must pay attention, especially when using two-wire sensing elements.
7. PLC cannot read input signals which change too fast. For PLC, regardless of input signals, ON or OFF, its duration must be longer than the scan time; otherwise, PLC is likely failed to read correct signals, as shown in Fig.1-4.
   When input signal changes too fast to be read normally, the interrupt input function or pulse wave capture function can be employed.

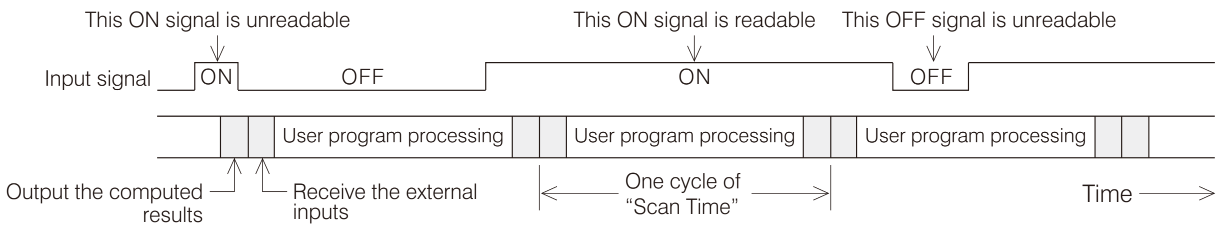


Fig. 1-4 PLC cannot read input signal which is swap over too fast

1-1-5 PLC Output Signals

The PLC's output endpoint is a window for PLC to send out the computed results, and is used to interface with a variety of loads. Since the output signals are to drive external loads and to complete machine's actions, must pay extra attention about the safety. Please note the following points:

1. Never ignore the possibility of the PLC failure. It is required to design an external safety circuit and safety mechanism to avoid accidents.
2. PLCs typically push external loads through relays or transistors. The difference between the characteristics of two is great, with each advantages and disadvantages. In designing, one should select properly.

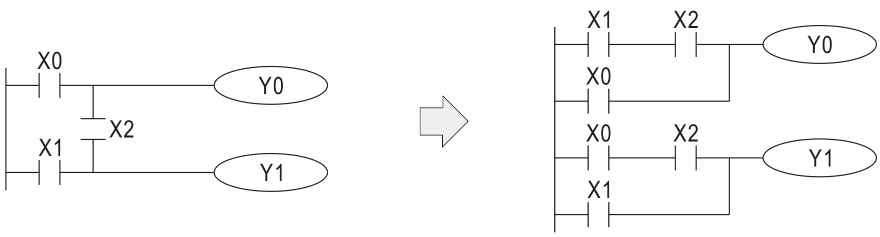
   The advantages of using a relay are that its contact switch has no current polarity and promotes more current (about 2A), and AC or DC power supply can be used. Its shortcomings are that the contact switch is mechanical, with mechanical life and electrical life limits. Basically, the relay is not a durable product, and becomes the unreliable factor in a control system.

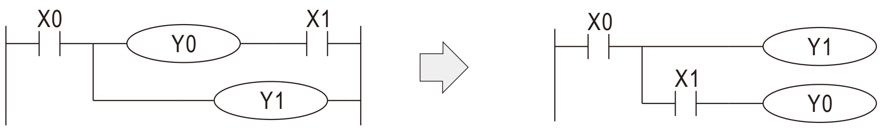
   The advantages of using a transistor are since it switches by the semiconductor. The number of swap times is unlimited and the speed of switch is fast, depending on demands could reach hundreds kHz. Its shortcomings are has current polarity, can only be used for DC load, below 30V and with less current (about 0.1A ~ 0.5A).

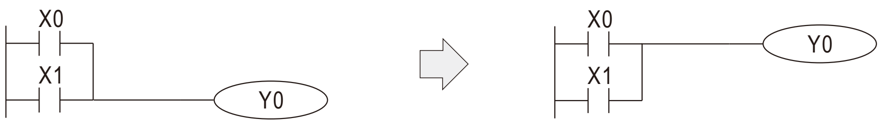
   When using a relay output, must pay attention to the frequency of action then to calculate the machine's life. That is for to avoid controller breakdown due to the damage of the relay. Therefore, most control panels take the approach of attaching a relay externally. Using external relays to promote loads can be considered as the most correct way to assemble panels. At this point, a transistor output controller can be used to enhance the panel's reliability.
3. In order to prevent interference from noises, the PLC outputs must take isolation measures. The relay output is magnetically isolated and this can cause an output delay of about 10 ms. The transistor output is isolated by a photocoupler, the output delay is below 1ms for the general output but only several μs delay for high-speed output.
4. There is a specification of output capability for the load and should be taken seriously. Do not use excessively. In general, the output specification of the relay is AC220V 2A or less; the transistor is DC30V 0.1A ~ 0.5A or less. For the unsuitable load to push excessive output at a short time may not have a problem. However, it will certainly affect the service life of PLC.

1-1-6 Some Improper Diagrams at a PLC Program

Some conventional circuits of relay switchboards cannot be directly replaced by the Ladder Diagrams at PLC. The following diagrams indicate such circuit loops on the left and the alternative for PLC are on the right.





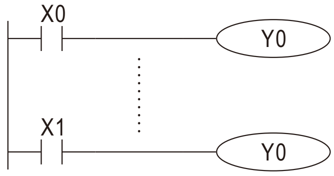


1-1-7 Double Coil Output

The PLC program with the following characteristic will affect the operation result, please pay special attention.

1. When executing the user program PLC follows sequential scan (from left to right, up to down).
2. In the PLC executing process, only the contents of the data memory will be used and changed. Actually to drive the external loads are performed after the execution of the program and there is a procedure to output the computing results.

As the figure below shows, Y0 is set as OUT twice, which is referred to as double coil output In this case, it is the state of X1 that actually drives external loads.



To resolve the confusion of double coil output, methods are recommended as follows:

- Parallel connect the driving conditions and then output once.
- Use the SET, RST instructions.
- Use the CJ instruction.
- Use the Step Ladder (STL) instruction or Sequential Function Chart (SFC)

1-1-8 Conclusion

A PLC usually performs the important part in the automatic control system. Appropriate protection of the PLC can improve the reliability of the system.

Traditional PLCs which we are familiar with usually use AC power supply, relay output and terminal block wiring. The AC power supply unit is susceptible to damage due to the instability of the power system (especially in non-industrial advanced countries). The relay output has the lifespan limit because of mechanical contact. Also, to disassemble the wiring at terminal blocks is time-consuming, inconvenient to fault repair (although has separable terminal products but with higher price).

The best configuration for the new generation of PLCs should be the combination of DC power supply, transistor output and connector wiring to exclude factors which affect the reliability from PLC systems as far as possible. In addition, this way makes it possible to reduce wiring time, reduce costs and improve maintenance convenience. It would be more p erfect if they are equipped with a memory card to store programs and data . When a PLC fails, the really important and difficult part is to transfer the program (which need system-related professionals to deal with), rather than the work of dissembling a terminal block (what a general electromechanical person can handle). With a memory card, the PLC replacement work can become easy.

Before using a PLC, you should really understand the specifications of each elements and avoid excessive use (with particular attention to power capacity and output specifications). Also, choose the right products according to your needs in order to design a high cost-effective control system.

PLC is the core of a control system!

Having correct knowledge and ideas helps you establish a stable and reliable control system.

1-2 The VS Series PLC Products Overview

VS Series Controller Provides [Comprehensive] Control Application

More Effective

The VS Series is base on high performance 32-bit 96 MHz processor, the overall efficiency is 10 times more then the VB or VH series PLC.

The size of project memory is enlarged from 4~16K to 16~64K words, also the number of data registers is greatly increased.

The communication could expand up to 6 ports (USB and CP1~CP5 multi-functional ports), fully support high-level control system.

The 4 pulse out points have various positioning functions. The 8 high-speed inputs provide plenty functions such as external interrupt, hardware / software high-speed counter, pulse capture, period measurement, handwheel...

More Fast

The new processor only takes 0.15μs per basic instruction step, that performance is 2.5 times faster than before. Both the pulse inputs and outputs can reach 1 MHz, more powerful than similar competitor.

By superbly fast USB port to read or write the user project just spends in an instant, 16K Words less than 3 Sec. This progress far exceeds the past.

More Diverse

The VS Series has the VS1 (General), VS2 (Advanced), VSM (Motion Control) and VS3 (High Performance) Main Units. The applicable coverage is from simple to complicated control.

By various Main Units, Modules, Cards, Memory Cards and the modular design to produce a complete and flexible combination.

Remarkable add-on card structure with the DIO, Communication and Special Cards to provide a superb cost-effective, space saving and flexible expansion.

Simple to construct and maintain, this VS Series is the best choice of programmable logic controller.

More Competitive Advantage

The VIGOR R&D team has accumulated decades of experience for "More diverse combination" and "The most suitable product" design concepts. Carefully selected high quality CPU to develop the excellent and stable VS Series also with highly competitive price.

The VS Series is close to the automation market and demand by flexible combination. Can raise product level with expandability and more competitive.

Item \ Series	VS1 General	VS2 Advanced	VSM Motion Control	VS3 High Performance
Process Time of Basic Instruction	0.17μs/Step	0.17μs/Step	0.17μs/Step	0.15μs/Step
Memory Capacity of Project	16K Words	32K Words	32K Words	64K Words
Max. Input/Output Points	128 pt. + 24 (at Exp. Cards)	256 pt. + 24 (at Exp. Cards)	256 pt. + 24 (at Exp. Cards)	512 pt. + 24 (at Exp. Cards)
Programming Port	Built-in 12Mbps high-speed Mini USB port			
Unit Built-In Comm. Port	CP1 (RS-485) provides various communication modes: Computer Link, MODBUS (Master / Slave), CPU Link, Non-protocol...			
Expandable Comm. Port	CP2~CP3 (abilities = CP1)			CP2~CP5 (abilities = CP1)
Multi-Func. High Speed In	8 points 10 kHz	8 points 50 kHz	4 points 200 kHz☆ & 4 points 50 kHz	
Pulse Output	4 points (axes) 50 kHz※	4 points (axes) 50 kHz※	4 points (axes) 200 kHz※	
Number of Special Modules	—	8	8	16
Number of Special Cards	1	3	3	3
Function of Expansion Card	EC1 ~ EC3 for the DIO, communication (RS-232, RS-485) or special card (e.g. Analog, Temperature, Inverter Speed Control)			
Function of Memory Card	Maintenance-free user project & large data memory card provides the best subject transplanting method for system maintain			

☆For the VSM-28ML-D Line Driver model, its two Hardware High-Speed Counters can count 1 MHz pulses respectively.

※ Those 4 outputs are available generate 1 MHz pulses individually at the VSM-28-ML-D Line Driver model; 200 kHz at the VSM/VS3's NPN; 50 kHz at the VS1/VS2's NPN or 5 kHz at the PNP Main Unit. Not available in the relay output unit.

The VS Family has the VS1 General, VS2 Advanced, VSM Motion Control and VS3 High Performance series controllers, the complete product line can satisfy various applications from basic to high-end and the combination of the best balance between cost and performance. Also, based on the concept of "the most suitable product" to enhance the competitive power and achieve the value beyond price.

The VS1 General series is suitable for various easy auto-control systems to satisfy with simple sequence control functions, such as cargo lift, parking equipment, conveyor, shoe machinery, brick machinery, woodwork machinery, etc.

The VS2 Advanced series is suitable for general purpose auto-control systems to satisfy with analog or temperature demanded controls, such as passenger lift, rubber vulcanizer, plastic injection molding machine, metal stamping machine, packing machinery, etc.

The VSM Motion Control series is a good match for various industrial machinery needing precise positioning functions by servo/stepper motors, which including labeling machine, sleeving machine, dispenser, film laminating machine, pipe bending machine, cutting machine, bar feeder, etc.

The VS3 High Performance series is the solution for various control systems of complicated sequence or large scale, that including printing machinery, automatic production line, semiconductor peripheral device, automated storage/retrieval system, electroplating procedure control, etc.

1-3 Specification Table of All the VS Series Main Units

Item			VS1 Series	VS2 Series	VSM Series	VS3 Series
Operation Control Method			Cyclic Operation by Stored Program			
Programming Language			Ladder Diagram + Sequential Function Chart (SFC) or Ladder Diagram + Step Ladder (STL)			
I/O Control Method			Batch Processing			
Process Time	Basic Instruction		0.17 μs			0.15 μs
	Application Instruction		A few μs ~ Hundreds of μs			
Number of Instructions	Basic Instruction		29			
	SFC Instruction		2			
	STL Instruction		2			
	Application Instruction		169	171	171	209
Project Memory Capacity (Flash ROM)			The project at the memory is including the parameter area, user program, pointers, tables and comments.			
			16k Words	32k Words	32k Words	64k Words
Max. Input/Output Points			128 points + 24 at Expansion Card	256 points + 24 at Expansion Card	256 points + 24 at Expansion Card	512 points + 24 at Expansion Card
Digital Input / Output	External Input (X)		64 points: X0 ~ X77	128 points: X0 ~ X177	128 points: X0 ~ X177	256 points: X0 ~ X377
	External Output (Y)		64 points: Y0 ~ Y77	128 points: Y0 ~ Y177	128 points: Y0 ~ Y177	256 points: Y0 ~ Y377
Internal Relay	Auxiliary Relay (M)	General	6192 points: M0 ~ M1999 and M4000 ~ M8191			
		Latched	2000 points: M2000 ~ M3999			
		Special	512 points: M9000 ~ M9511			
	Step Relay (S)	Initial	10 points: S0 ~ S9			
		General	3086 points: S10 ~ S499 and S1500 ~ S4095			
		Latched	900 points: S500 ~ S899 and S1000 ~ S1499			
		Annunciator	100 points: S900 ~ S999 (Latched)			
Timer (T)	100ms		200 points: T0 ~ T199 (Timer range: 0.1 ~ 3,276.7 sec.)			
	10ms		46 points: T200 ~ T245 (Timer range: 0.01 ~ 327.67 sec.)			
	1ms (Retentive)		4 points: T246 ~ T249 (Timer range: 0.001 ~ 32.767 sec.)			
	100ms (Retentive)		6 points: T250 ~ T255 (Timer range: 0.1 ~ 3,276.7 sec.)			
	1ms		256 points: T256 ~ T511 (Timer range: 0.001 ~ 32.767 sec.)			
Counter (C)	16-bit Up	General	100 points: C0 ~ C99 (Range: 0 ~ 32,767)			
		Latched	100 points: C100 ~ C199 (Range: 0 ~ 32,767)			
	32-bit Up / Down	General	20 points: C200 ~ C219 (Range: –2,147,483,648 ~ 2,147,483,647)			
		Latched	15 points: C220 ~ C234 (Range: –2,147,483,648 ~ 2,147,483,647)			
Software High Speed Counter (C)	32-bit Up / Down, Latched	1-Phase	11 points: C235 ~ C245 (Range: –2,147,483,648 ~ 2,147,483,647)			
		2-Phase	5 points: C246 ~ C250 (Range: –2,147,483,648 ~ 2,147,483,647)			
		A / B Phase	5 points: C251 ~ C255 (Range: –2,147,483,648 ~ 2,147,483,647)			
Hardware High Speed Counter			2 points: HHSC1 ~ HHSC2 (Range: –2,147,483,648 ~ 2,147,483,647)			
Data Register	General (D)		7000 points: D0 ~ D6999			
	Latched (D)		2000 points: D7000 ~ D8999			
	Special (SD)		512 points: D9000 ~ D9511			
	Index Register (V / Z)		16 points: V0 ~ V7 and Z0 ~ Z7			
	Extension Register (R)		10000 points: R0 ~ R9999			24000 points: R0~23999
Pointer	Mark Pointer		1024 points: Each pointer can be named by P0 ~ P1023 or 16 characters			
	Branch Pointer (P)		1024 points: P0 ~ P1023			
	Table Nickname		32 points: Each table can be named by Q0 ~ Q31 or 16 characters			
	Table Code (Q)		32 points: Q0 ~ Q31			
	Interrupt Pointer (I)		21 points: 8 pt. for external interrupt, 3 pt. for timer interrupt, 10 pt. for high speed counter interrupt			
	Nest Pointer (N)		8 points: N0 ~ N7			

Item			VS1 Series	VS2 Series	VSM Series	VS3 Series
Range of Constant	Decimal (K)	16-bit	K–32,768 ~ K32,767			
		32-bit	K–2,147,483,648 ~ K2,147,483,647			
	Hexadecimal (H)	16-bit	H0 ~ HFFFF			
		32-bit	H0 ~ HFFFFFFFF			
	Real No. (E)	32-bit	E–3.402 + 38 ~ E3.402 + 38, decimal or exponent notation			
Comm. Function	Main Unit Built-in Comm. Port	Programming	12Mbps high-speed Mini USB communication port			
		Multi-Func.	CP1 (RS-485) is available for the Computer Link, MODBUS, CPU Link, Non-Protocol, etc.			
	Expanded Multi-Func. Port		CP2~CP3 (at the EC1)			CP2 ~ CP3 (at the EC1)
			—	—	—	CP4 ~ CP5 (at the EC3)
Multi-Function High Speed Input	Input Response Frequency		10kHz × 8 points	50kHz × 8 points	200kHz × 4 points ☆ 50kHz × 4 points	200kHz × 4 points 50kHz × 4 points
	Input Response Time Adj.		8 points: X0 ~ X7 (0~60ms)			
	External Interrupt Input		8 points: X0 ~ X7 (with delay function)			
	Pulse Capture Input		8 points: X0 ~ X7			
	Pulse Measurement Input		4 points: X0, X1, X3, X4 (with width period measurement function)			
	Frequency Meter Input		8 points: X0 ~ X7			
	Software High Speed Counter		Support 1, 2 or AB phase counting mode, 1-phase 8 points or 2/AB phase 4 sets max.			
	Hardware High Speed Counter		2 sets: HHSC1 and HHSC2. Support U, U/D+DIR, U+D, AB×1, AB×2 or AB×4 operating mode			
	Electronic Handwheel		Cooperate with high speed pulse output to control positioning			
High Speed Pulse Output			4 points 50kHz (4-axis positioning control)	4 points 50kHz (4-axis positioning control)	4 points 200kHz (4-axis positioning control) ☆	
Real Time Clock (Optional)			By installing VS-MCR Multi-Function Memory Card to indicate year, month, date, hour, min., sec. & week			
Expansion Memory (Optional)			By installing a VS-MCR/VS-MC card to expand no-battery required 16Mb latched memory for user project and 655,360 Words data bank			
Special Module	Number of Special Module Available		—	8	8	16
	Type of Special Module		—	Analog I/O Module, Temperature Input Module, etc.		
Expansion Card	Expansion Card Socket at Unit		10/14M Main Unit (EC1), 20/24M Main Unit (EC1~EC2), 28/32M Main Unit (EC1~EC3)			
	Type of Expansion Card		DI/DO, communication or special function card (AI, AO, temperature input, inverter speed control, etc.)			
	Number of Special Card Available		1 (VS-3AV-EC won't occupy)	3	3	3

☆ At the VSM-28ML model, the 4 line driver input points for the HHSC1 & HHSC2 can individually count 1 MHz pulses; also, the 4 line driver output points can individually generate 1MHz pulses.

1-4 Overview of VS Series PLC Models

Item	Model Name	Main Specification
VS1 Series Main Unit	VS1-10M★-D	VS1 Main Unit: 6 DI (DC 24V, X0~X5 10 kHz); 4 DO ★; 16K words project memory; 1 Expansion Card socket; I/O by screw-clamp terminal
	VS1-14M★-D	VS1 Main Unit: 8 DI (DC 24V, X0~X7 10 kHz); 6 DO ★; 16K words project memory; 1 Expansion Card socket; I/O by screw-clamp terminal
	VS1-20M★-D	VS1 Main Unit: 12 DI (DC 24V, X0~X7 10 kHz); 8 DO ★; 16K words project memory; 2 Expansion Card sockets; I/O by screw-clamp terminal
	VS1-24M★-D	VS1 Main Unit: 14 DI (DC 24V, X0~X7 10 kHz); 10 DO ★; 16K words project memory; 2 Expansion Card sockets; I/O by screw-clamp terminal
	VS1-28M★-D	VS1 Main Unit: 16 DI (DC 24V, X0~X7 10 kHz); 12 DO ★; 16K words project memory; 3 Expansion Card sockets; DIO Expansion Module available; I/O by screw-clamp terminal
	VS1-32M★-D	VS1 Main Unit: 20 DI (DC 24V, X0~X7 10 kHz); 12 DO ★; 16K words project memory; 3 Expansion Card sockets; DIO Expansion Module available; I/O by screw-clamp terminal
	VS1-32MT-DI	VS1 Main Unit: 16 DI (DC 24V, X0~X7 10 kHz); 16 DO (100mA NPN transistor, Y0~Y3 50 kHz); 16K words project memory; 3 Expansion Card sockets; DIO Expansion Module available; I/O by IDC connector.
VS2 Series Main Unit	VS2-24M★-D	VS2 Main Unit: 12 DI (DC 24V, X0~X7 50 kHz); 12 DO ★; 32K words project memory; 2 Expansion Card sockets; DIO Expansion & 8 Special Modules available; I/O by screw-clamp terminal
	VS2-32M★-D	VS2 Main Unit: 16 DI (DC 24V, X0~X7 50 kHz); 16 DO ★; 32K words project memory; 3 Expansion Card sockets; DIO Expansion & 8 Special Modules available; I/O by screw-clamp terminal
	VS2-32MT-DI	VS2 Main Unit: 16 DI (DC 24V, X0~X7 50 kHz); 16 DO (100mA NPN transistor, Y0~Y3 50 kHz); 32K words project memory; 3 EC sockets; DIO Expansion & 8 Special Modules available; I/O by IDC connector
VSM Series Main Unit	VSM-14MT-D	VSM Main Unit: 8 DI (DC 24V, 4×200 kHz + 4×50 kHz); 6 DO (500mA NPN transistor, Y0~Y3 200 kHz); 32K words project memory; 1 Expansion Card socket; I/O by screw-clamp terminal
	VSM-24MT-D	VSM Main Unit: 12 DI (DC 24V, 4×200 kHz + 4×50 kHz); 12 DO (500mA NPN transistor, Y0~Y3 200 kHz); 32K words project memory; 2 Expansion Card sockets; DIO Expansion & 8 Special Modules available; I/O by screw-clamp terminal
	VSM-32MT-D	VSM Main Unit: 16 DI (DC 24V, 4×200 kHz + 4×50 kHz); 16 DO (500mA NPN transistor, Y0~Y3 200 kHz); 32K words project memory; 3 Expansion Card sockets; DIO Expansion & 8 Special Modules available; I/O by screw-clamp terminal
	VSM-28ML-D	VSM Main Unit: 4 Line Driver DI (for 2 hardware counters up to 1 MHz) + 12 DI (DC 24V, 4×50 kHz & 8 normal); 8 Line Driver DO (4 × 1 MHz & 4 normal) + 4 DO (500mA NPN transistor); 32K words project memory; 3 Expansion Card sockets; DIO Expansion & 8 Special Modules available; I/O by screw-clamp terminal
	VSM-32MT-DI	VSM Main Unit: 16 DI (DC 24V, 4×200 kHz + 4×50 kHz); 16 DO (100mA NPN transistor, Y0~Y3 200 kHz); 32K words project memory; 3 Expansion Card sockets; DIO Expansion & 8 Special Modules available; I/O by IDC connector.
VS3 Series Main Unit	VS3-32M★-D	VS3 Main Unit: 16 DI (DC 24V, 4×200 kHz + 4×50 kHz); 16 DO ★(Y0~Y3 200 kHz at NPN transistor model); 64K words project memory; 3 Expansion Card sockets; DIO Expansion & 16 Special Modules available; I/O by screw-clamp terminal
	VS3-32MT-DI	VS3 Main Unit: 16 DI (DC 24V, 4×200 kHz + 4×50 kHz); 16 DO (100mA NPN transistor, Y0~Y3 200 kHz); 64K words project memory; 3 EC sockets; DIO Expansion & 16 Special Modules available; I/O by IDC connector
DIO Expansion Module	VS-8X	DI Expansion Module: 8 DI (DC 24V); input by screw-clamp terminal
	VS-16X	DI Expansion Module: 16 DI (DC 24V); input by screw-clamp terminal
	VS-8Y★	DO Expansion Module: 8 DO ★; output by screw-clamp terminal
	VS-16Y★	DO Expansion Module: 16 DO ★; output by screw-clamp terminal
	VS-8XY★	DIO Expansion Module: 4 DI (DC 24V); 4 DO ★; I/O by screw-clamp terminal
	VS-16XY★	DIO Expansion Module: 8 DI (DC 24V); 8 DO ★; I/O by screw-clamp terminal
	VS-28XYR	DIO Expansion Module: 16 DI (DC 24V); 12 DO (2A Relay); I/O by screw-clamp terminal
	VS-32XY★	DIO Expansion Module: 16 DI (DC 24V); 16 DO ★; I/O by screw-clamp terminal
	VS-16X-I	DI Expansion Module: 16 DI (DC 24V); input by IDC connector
	VS-16YT-I	DO Expansion Module: 16 DO (100mA NPN transistor); output by IDC connector
	VS-16XYT-I	DIO Expansion Module: 8 DI (DC 24V); 8 DO (100mA NPN transistor); I/O by IDC connector
	VS-32XYT-I	DIO Expansion Module: 16 DI (DC 24V); 16 DO (100mA NPN transistor); I/O by IDC connector
Power Module	VS-PSD	Power Repeater Module: DC 24V power input to transfer to DC 5V 500mA + DC 12V 800mA, those inner power outputs provide for the Modules behind
Special Function Module	VS-4AD	Analog Input Module: 4 channel (16-bit) inputs, each channel could input either –10~+10V, 4~20mA or –20~+20mA; isolated
	VS-2DA	Analog Output Module: 2 channel (16-bit) outputs, each channel could output either –10~+10V, 4~20mA or –20~+20mA; isolated
	VS-3A	Analog I/O Module: 2 channel (16-bit) inputs + 1 channel (16-bit) output, each channel could input/output either –10~+10V, 4~20mA or –20~+20mA; isolated
	VS-6A	Analog I/O Module: 4 channel (16-bit) inputs + 2 channel (16-bit) outputs, each channel could input/output either –10~+10V, 4~20mA or –20~+20mA; isolated
	VS-4TC	Thermocouple Temperature Input Module: 4 channel thermocouple (K, J, R, S, T, E, B or N type) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-8TC	Thermocouple Temperature Input Module: 8 channel thermocouple (K, J, R, S, T, E, B or N type) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-2PT	PT-100 Temperature Input Module: 2 channel (3-wire PT-100) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-4PT	PT-100 Temperature Input Module: 4 channel (3-wire PT-100) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-2PG	Pulse Generator Module: 2 sets of 200 kHz high speed pulse outputs for 2 axes position control
	VS-4PG	Pulse Generator Module: 4 sets of 200 kHz high speed pulse outputs for 4 axes position control

Item	Model Name	Main Specification
DIO Expansion Card	VS-4X-EC	DI Expansion Card: 4 DI (DC 24V); output by screw-clamp terminal
	VS-8X-EC	DI Expansion Card: 8 DI (DC 24V); input by screw-clamp terminal
	VS-4Y★-EC	DO Expansion Card: 4 DO ★; output by screw-clamp terminal
	VS-8YT-EC	DO Expansion Card: 8 DO (DC 24V, 300mA NPN transistor); output by screw-clamp terminal
	VS-4XY★-EC	DIO Expansion Card: 2 DI (DC 24V); 2 DO ★; I/O by screw-clamp terminal
	VS-8XY★-EC	DIO Expansion Card: 4 DI (DC 24V); 4 DO ★; I/O by screw-clamp terminal
	VS-8XI-EC	DI Expansion Card: 8 DI (DC 24V); input by IDC connector
	VS-8YTI-EC	DO Expansion Card: 8 DO (DC 24V, 100mA NPN transistor); output by IDC connector
	VS-E8X-EC	DI Expansion Card for VS-32XY★ module: 8 DI DC 24V, Sink/Source selectable; input by screw-clamp terminal
	VS-E8YT-EC	DO Expansion Card for VS-32XY★ module: 8 DO (DC 24V, 0.3A NPN transistor); output by screw-clamp terminal
Comm. Expansion Card	VS-485-EC	RS-485 Communication Expansion Card: One non-isolated RS-485 port with TX / RX indicators; dist. 50m Max.
	VS-485A-EC	RS-485 Communication Expansion Card: One isolated RS-485 port with TX / RX indicators; dist. 1000m Max.
	VS-D485-EC	RS-485 Communication Expansion Card: Dual non-isolated RS-485 ports with TX / RX indicators; dist. 50m Max.
	VS-D485A-EC	RS-485 Communication Expansion Card: Dual isolated RS-485 ports with TX / RX indicators; dist. 1000m Max.
	VS-D232-EC	RS-232C Communication Expansion Card: Dual non-isolated RS-232 ports with TX / RX indicators; dist. 15m Max.; wiring by the RX / TX / SG terminals
	VS-D52A-EC	RS-485 ＋ RS-232C Communication Expansion Card: One isolated RS-485 port (1000m) & one non-isolated RS-232C port (15m), both with TX / RX indicators and wiring by terminals
	VS-ENET-EC	Ethernet ＋ RS-485 Communication Expansion Card: One Ethernet port (with additional non-isolated RS-485, dist. 50m) & one non-isolated RS-485 port (dist. 50m), both with TX / RX indicators
Special Function Card	VS-3AV-EC	Brief Voltage I/O Card: 2 channel (0~10V, 12-bit) inputs; 1 channel (0~10V, 10-bit) output; with a calibrated DC 10V output; non-isolated
	VS-4AD-EC	Analog Input Card: 4 channel (12-bit) inputs, each channel could output either 0~10V, 4~20mA or 0~20mA; non-isolated
	VS-2DA-EC	Analog Output Card: 2 channel (12-bit) outputs, each channel could input either 0~10V, 4~20mA or 0~20mA; non-isolated
	VS-4A-EC	Analog I/O Card: 2 channel (12-bit) inputs ＋ 2 channel (12-bit) outputs, each channel could input/output either 0~10V, 4~20mA or 0~20mA; non-isolated
	VS-3ISC-EC	Inverter Speed Control Card: 3 channel (0.1% resolution) voltage outputs; totally isolated for each channel
	VS-2TC-EC	Thermocouple Temperature Input Card: 2 channel (K, J, R, S, T, E, B or N type thermocouple, 0.2~0.3℃ resolution) inputs; non-isolated
	VS-4TC-EC	Thermocouple Temperature Input Card: 4 channel (K, J, R, S, T, E, B or N type thermocouple, 0.2~0.3℃ resolution) inputs; non-isolated
	VS-1PT-EC	PT-100 Temperature Input Card: 1 channel (3-wire PT-100, 0.1℃ resolution) input; non-isolated
	VS-2PT-EC	PT-100 Temperature Input Card: 2 channel (3-wire PT-100, 0.1℃ resolution) inputs; non-isolated
Memory Card	VS-MC	Memory Card: No battery required 16Mb Flash ROM for user's project and data-bank (655,360 words) storage
	VS-MCR	Multi-Function Memory Card: 16Mb Flash ROM for user's project and data-bank (655,360 words) storage; with the Real Time Clock function
Connection Cable	VSPC-200A	USB Communication Cable: Between the PLC's Mini USB Programming Port and a computer's A-type USB; length: 200 cm
	VSEC-□□□	Extension Cable: For the Expansion Slot of the VS series; length□□□:50/100 cm
IDC Connector Related Accessory	VB-T8R	8 Relays Output Module: 16A 1c contact relays; with varistors and relay sockets
	VB-T8RS	8 Relays Output Module: 5A 1a contact relays; with 5mm removable screw-clamp terminals
	VB-T8M	8 MOSFETs Output Module: 2A current source MOSFETs; with flyback diodes
	VB-T16M	16 MOSFETs Output Module: 2A current source MOSFETs; with flyback diodes
	VB-T16TB	16 Points Adapted Board: Transfer between the IDC connectors and screw-clamp terminals
	VBIDC-□□□	IDC Ribbon Cable: Assembled with two 10-pin female connectors; length□□□:50/100/150/200/250/300 cm
	VBIW-□□□	IDC Dispersed Wires: An IDC female connector with 10 rainbow 22 AWG wire; length□□□:50/100/200/300 cm
	VBIDC-FC100	10-pin Ribbon Cable: Flat, Grey, 28 AWG; length: 100 foot
	VBIDC-FC250	10-pin Ribbon Cable: Flat, Grey, 28 AWG; length: 250 foot
	VBIDC-HD20	10-pin IDC Connector: Female connector with strain relief, Grey, 20 pcs.
	VBIDC-HD100	10-pin IDC Connector: Female connector with strain relief, Grey, 100 pcs.
	VB-HT214	A crimping tool of IDC ribbon cable

★ Selectable output:

R: 2A Relay;
T: 0.5A NPN transistor (EC cards are 0.3A); at Y0~3 could generate purse (VS1/2: 50kHz; VSM/3: 200kHz);
P: 0.5A PNP transistor, at Y0~3 could generate 1kHz purse
All Main Unit, Special Module and IDC's module are required DC 24V -15% / +20% power input

MEMO

2. Component Descriptions

2-1 Table of Components

Item	Description		
External Input (X)	VS1 Series	X0~X77, 64 Pt.(named by the octal system)	
	VS2 Series	X0~X177, 128 Pt.(named by the octal system)	
	VSM Series	X0~X177, 128 Pt.(named by the octal system)	
	VS3 Series	X0~X377, 256 Pt.(named by the octal system)	
External Output (Y)	VS1 Series	Y0~Y77, 64 Pt.(named by the octal system)	
	VS2 Series	Y0~Y177, 128 Pt.(named by the octal system)	
	VSM Series	Y0~Y177, 128 Pt.(named by the octal system)	
	VS3 Series	Y0~Y377, 256 Pt.(named by the octal system)	

Auxiliary Relay (M)	General Use	M0~M1999 , M4000~M8191, Tot.= 6192 Pt.	
	Latched	M2000~M3999 , 2000 Pt.	
	Special Use	M9000~M9511 , 512 Pt.	
Step Relay (S)	General Use	S0~S9 , 10 Pt.	
	Latched	S10~S499, S1500~S4095 , Tot.= 3086 Pt.	
	Special Use	S500~S899, S1000~S1499 , Tot.= 900 Pt.	
	Annunciator	S900~S999, 100 Pt. latched	
Timer (T)	100ms	T0~T199, 200 Pt.Range: 0.1~3,276.7 Sec.	
	10ms	T200~T245, 46 Pt.Range: 0.01~327.67 Sec.	
	1ms Retentive	T246~T249, 4 Pt.Range: 0.001~32.767 Sec.	
	100ms Retentive	T250~T255, 6 Pt. Range: 0.1~3,276.7 Sec.	
	1ms	T256~T511, 256 Pt. Range: 0.001~32.767 Sec.	
Counter (C)	16-bit Up	C0~C99, 100 Pt. Range: 0~32,767	
		C100~C199, 100 Pt. (Range: 0~32,767; Latched)	
	32-bit Up / Down	C200~C219, 20 Pt. (Range: –2,147,483,648~2,147,483,647)	
		C220~C234, 15 Pt. (Range: –2,147,483,648~2,147,483,647; Latched)	
Software High Speed Counter (C)	32-bit Up / Down, Latched	C235~C245, 11 Pt. 1-phase counter	For the X0~X7, each can be used once
		C246~C250, 5 Pt. 2-phase counter	
		C251~C255, 5 Pt., AB-phase counter	
Register (D)	General Use	D0~D6999, 7000 Pt.	
	Latched	D7000~D8999, 2000 Pt.	
	Special Use	D9000~D9511, 512 Pt.	
Index Register (V & Z)		V0~V7, Z0~Z7, 16 Pt.	
Extension Register (R)	VS1, VS2, VSM Series	R0~R9999, 10000 Pt.	
	VS3 Series	R0~23999, 24000 Pt.	
Mark Pointer		1024 points: Each pointer can be named by P0~P1023 or 16 characters	1024 pointers in total
Branch Pointer (P)		1024 points: P0~P1023	
Table Nickname		32 points: Each table can be named by Q0~Q31 or 16 characters	32 tables in total
Table Code (Q)		32 points: Q0~Q31	
Interrupt Pointer (I)	External Interrupt	8 points: IX0P/F~IX7P/F	
	Timer Interrupt	3 points: ITA□□~ITC□□	
	Software High Speed Counter	8 points: IHC0~IHC7	
	Hardware High Speed Counter	2 points: IHHC1~IHHC2	
Nested Pointer N		N0~N7, 8 Pt.	

Decimal (K)	16-bit	K–32,768~K32,767
	32-bit	K–2,147,483,648~K2,147,483,647
Hexadecimal (H)	16-bit	H0~HFFFF
	32-bit	H0~HFFFFFFFF
Real Number (E)	32-bit	E–3.402 +38~E3.402 +38 decimal or exponent notation

2-2 External Input (X) and External Output (Y)

2-2-1 External Input (X)

VS series PLCs read the ON/OFF status of various external switches and sensing elements as operating conditions through the input points. To prevent problems such as noise interference and switch bouncing, there is a filter of about 10ms equipped at each input point. Since the external input X0~X7 of a Main unit are designated as multi-function input points to perform various high speed functions, therefore, the filter time of these 8 input points is adjustable.

Functions of these 8 output points are listed below.

Function	External Input Point								Ref. Page
	X0	X1	X2	X3	X4	X5	X6	X7	
Common Input	Use D9020 to adjust the filter time as 0~60ms								183
Frequency Meter	Use the SPD instruction to perform the speed detection function								190
Software High Speed Counter	Use C235~C255 1-phase/2-phase/AB phase counters, besides to generate IHC0~IHC7 interrupt								25
External Interrupt	IX0P/F	IX1P/F	IX2P/F	IX3P/F	IX4P/F	IX5P/F	IX6P/F	IX7P/F	54
Pulse Capture	M9170	M9171	M9172	M9173	M9174	M9175	M9176	M9177	55
Hardware High Speed Counter	HHSC1, to make the IHHC1 interrupt		—	HHSC2, to make the IHHC2 interrupt		—	—	—	58
Pulse Measurement	Width / Period measurement		—	Width / Period measurement		—	—	—	56
Positioning Control	Can be the input points of the positioning control's DOG, PG0, INT signals or for the handwheel.								422

- For the descriptions of each item, please refer to the page listed at the “Ref. Page ” above.
- The contact of a common input point in the program is available to work with an other special function.
- When one of the X0~X7 performs a mentioned special function above, this input point is not reusable with another special function. However, the External Interrupt could cooperate with Pulse Measurement function. (For details, please see the specific function description.)
- When a mentioned special function above performs, the filter time of this input point will be automatically adjusted to 0ms (deviates form D9020). To prevent noise interference, input points X0~X7 are also equipped with RC filters. Thus, the filter time of 0 ms is not true 0 ms. In addition, the response time of input points X0~X7 is varied depending on the series of VS1, VS2, VSM or VS3.

  When filter time is adjusted as 0ms, the minimum pulse width required to transmit input signals to respective input points are listed as follows.

Series	External Input Point							
	X0	X1	X2	X3	X4	X5	X6	X7
VS1	50μs	50μs	50μs	50μs	50μs	50μs	50μs	50μs
VS2	10μs	10μs	10μs	10μs	10μs	10μs	10μs	10μs
VSM	2.5μs	2.5μs	10μs	2.5μs	2.5μs	10μs	10μs	10μs
VSM-28ML	0.5μs	0.5μs	10μs	0.5μs	0.5μs	10μs	10μs	10μs
VS3	2.5μs	2.5μs	10μs	2.5μs	2.5μs	10μs	10μs	10μs

- When a multi-function input point is performing a special function, the response of the point requires fast and relatively sensitive (the fast reaction, the more sensitive), contrarily susceptible to noise interference.Therefore, pay special attention to its external wiring, try to avoid interfering sources, and use isolation lines.

2-2-2 External Output (Y)

The contacts of external output in the VS series PLC are for the purpose of to drive external loads. By transmitting the operation results through its external output points, the PLC drives various loads, such as motors, electromagnetic valves, electromagnetic conductors, etc. and virtually perform control motions.
For various loading needs, the VS series PLCs have different output types, such as relay, NPN transistor and PNP transistor. Every relay has a coil and contact that uses magnetically coupled to separate the PLC internal circuit from the external, but a transistor output uses a photocoupler to prevent noise interference.

Relay outputs have approximately 10ms of output delay, while transistor outputs have approx. 1ms. These delays result that certain applications cannot be fully met. Therefore, the 4 output points of Y0~Y3 are designated to be multi-function output points.

Function	External output point				Ref. Page No.
	Y0	Y1	Y2	Y3	
Common Output	The output type is selectable, that includes the relay, NPN transistor or PNP transistor.				—
Paulse output	Use the PLSY or PLSR instruction to generate a pulse string to the driver of stepper or servo motor				191
PWM Output	Use the PWM instruction to generate PWM signal				193
Positioning Control	Use the Positioning Control instruction to generate positioning pulses directly to the driver of the step motor or servo motor, by the method to execute the accurate positioning control				433

- Except for the common output function, the other functions mentioned above can only available be used at a transistor output main unit.
- For the descriptions of each item, please refer to the page listed at the "Ref. Page No." above.
- At a time, each Y0~Y3 output point allows to execute one instruction of special function only. Also, cannot intermix different functional groups mentioned above.
- The frequency of the Y0~Y3 output at the VS1, VS2, VSM or VS3 series is different, the list below is the capability:

Series	External output point			
	Y0	Y1	Y2	Y3
VS1	50kHz	50kHz	50kHz	50kHz
VS2	50kHz	50kHz	50kHz	50kHz
VSM	200kHz	200kHz	200kHz	200kHz
VSM-28ML	1MHz	1MHz	1MHz	1MHz
VS3	200kHz	200kHz	200kHz	200kHz

2-2-3 External Input/Output Assigned Numbers

- The identification numbers of the External Input / Output points are assigned by the octal numeral system.
- The table below lists the assigned numbers of Input (X) and Output (Y) in the VS1 Main Unit:

Models	VS1-10M	VS1-14M	VS1-20M	VS1-24M	VS1-28M	VS1-32M	VS1-32MT-DI
Input No.	X0~X5 6 Pt.	X0~X7 8 Pt.	X0~X13 12 Pt.	X0~X15 14 Pt.	X0~X17 16 Pt.	X0~X23 20 Pt.	X0~X17 16 Pt.
Output No.	Y0~Y3 4 Pt.	Y0~Y5 6 Pt.	Y0~Y7 8 Pt.	Y0~Y11 10 Pt.	Y0~Y13 12 Pt.	Y0~Y13 12 Pt.	Y0~Y17 16 Pt.

- The table below lists the assigned numbers of Input (X) and Output (Y) in the VS2 Main Unit:

Models	VS2-24M	VS2-32M	VS2-32MT-DI
Input No.	X0~X13 12 Pt.	X0~X17 16 Pt.	X0~X17 16 Pt.
Output No.	Y0~Y13 12 Pt.	Y0~Y17 16 Pt.	Y0~Y17 16 Pt.

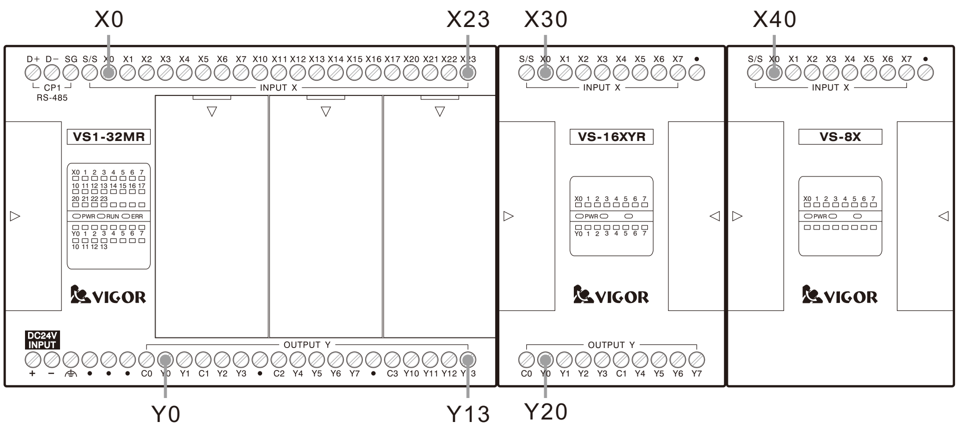
- The table below lists the assigned numbers of Input (X) and Output (Y) in the VSM Main Unit:

Models	VSM-14MT-D	VSM-24MT-D	VSM-32MT-D	VSM-32MT-DI	VSM-28ML-D
Input No.	X0~X7 8 Pt.	X0~X13 12 Pt.	X0~X17 16 Pt.	X0~X17 16 Pt.	X0~X17 16 Pt.
Output No.	Y0~Y5 6 Pt.	Y0~Y13 12 Pt.	Y0~Y17 16 Pt.	Y0~Y17 16 Pt.	Y0~Y13 12 Pt.

- The table below lists the assigned numbers of Input (X) and Output (Y) in the VS3 Main Unit:

Models	VS3-32M	VS3-32MT-DI
Input No.	X0~X17 16 Pt.	X0~X17 16 Pt.
Output No.	Y0~Y17 16 Pt.	Y0~Y17 16 Pt.

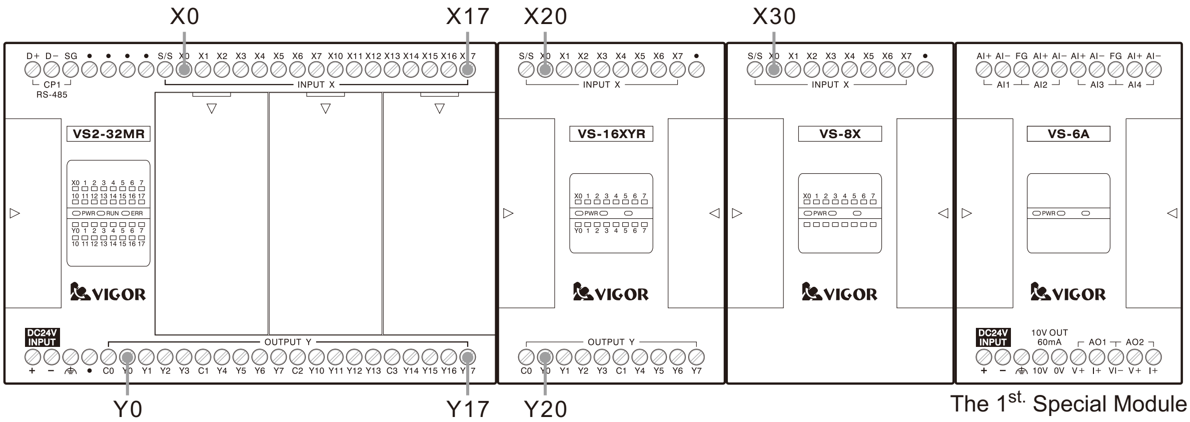
- Notes for the VS1 series PLC module expansion:
  - The Main Unit of VS1-10M / VS1-14M / VS1-20M / VS1-24M is not equipped with the module expansion slot.
  - The Main Unit of VS1-28M / VS1-32M is equipped with the module expansion slot, which can reach 64 inputs and 64 outputs (X0~X77, Y0~Y77), 128 points in total.
  - The module expansion slot of the VS1 series PLC can be equipped with DIO modules, but not allow special module.
  - The Main Unit of VS1-28M/VS1-32MT-DI occupies the assigned numbers of X0~X17 and Y0~Y17. Therefore, the first expansion module is started from X20 and Y20.
  - The Main Unit of VS1-32M occupies the assigned numbers of X0~X27 and Y0~Y17. Therefore, the first expansion module is started from X30 and Y20.



  - The VS-8XY module occupies 8 input points and 8 output points.
  - The VS-28XYR expansion module will occupy 16 input and 16 output points. Besides, it is unable to expand any module on its right side.

● Notes for VS2, VSM and VS3 series PLCs module expansion:

- The Main Unit of VSM-14M is not equipped with the module expansion slot, cannot be connected with any DIO expansion module or special module.
- The Main Unit of VS2, VSM or VS3 equips a module expansion slot, could connect with DIO expansion modules and special modules. (excluded the VSM-14M)
- The Main Unit of VS2 or VSM can use expansion modules to handle up to 128 inputs (X0~X177) and 128 outputs (Y0~Y177), total 256 I/O points. And also available 8 special modules.
- The Main Unit of VS3 can use expansion modules to handle up to 256 inputs (X0~X377) and 256 outputs (Y0~Y377), total 512 I/O points. And also available 16 special modules.
- All the Special and DIO Expansion Modules are serial connected on the right side of the Main Unit, and the connection sequence is without reserved. The closest Special Module is designated as the 1st. Special Module. After that, the followed Special Module is the 2nd., and so on. But, the DIO Expansion Module will not interfere with the ranking of Special Modules.
- The Main Unit of VS2, VSM or VS3 will occupy the assigned numbers X0~X17 and Y0~Y17, thus the beginning I/O address of the first expansion unit/module are the X20 and Y20.



- The VS-8XY module occupies 8 input points and 8 output points.
- The VS-28XYR expansion module will occupy 16 input and 16 output points. Besides, it is unable to expand any module on its right side.

● The VS Main Unit has the circuit for inner power supplying but the expanded modules and cards do not have, all power is from the Main Unit Please be aware of power consumption. If the inner power is insufficient, add the VS-PSD power repeater module is required.
For the power consumption calculations of individual expansion modules and expansion cards, please refer to “1-8 Specifications of Power Supply” in the Product Manual.

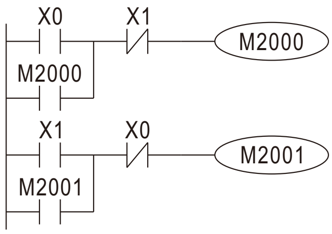
2-3 Auxiliary Relay (M)

The PLC includes considerable internal Auxiliary Relays (M), the function of those are to store up plenty ON/OFF status, which provided data for the processing demand. The operating method of an Auxiliary Relay is the same way to operate the External Output Y, but the contact of Auxiliary Relay can not directly drive an external load.
The identifier number of an Auxiliary Relay M uses the decimal method, and the functions can be divided into three different types:

(1) General Auxiliary Relay
When the input power is cut off during the PLC operation, the contents of this kind relay will be cleared. After the power resumes, the contents of this kind relay are all OFF.

(2) Latched Auxiliary Relay
When the input power is cut off during the PLC operation, the contents of this kind relay will be latched. After the power resumes, the contents still remain. That means, if the content of a latched relay is ON before power cut, its state will keep ON at the moment when the power is back.

- Below show some examples:
  Suppose there is a platform that, when the motor rotates forward and backward, it drives the platform to move left and right. In addition, the left limit switch (X0) and the right limit switch (X1) are connected to the left and right limitations of the moving range. If the operation of the motor should finish the uncompleted movement after the power is returned, that must use a latched auxiliary relay.



Self-holding circuit, use M2000 to retain the platform moving to the right.

Self-holding circuit, use M2000 to retain the platform moving to the left.

Also, now there are many mechines using human-machine interface (HMI) as operation interface. In such cases, many setups to PLCs by HMI may need the latched function.

(3) Special Auxiliary Relay
Each special relay has its own specific function. Some special relay has only the readable contact of its status, but without the writable coil to be driven by the program. Do not use special relays which are not defined. For the details of special relays, please refer to the section 2-14 “Special Relay and Special Register”.

- The auxiliary relays in the VS1, VS2, VSM and VS3 series share exactly the same ranges.:

General Auxiliary Relay	Latched Auxiliary Relay	Special Auxiliary Relay
M0~M1999, M4000~M8191 Total = 6192Pt.	M2000~M3999, Total = 2000 Pt.	M9000~M9511, Total = 512Pt.

2-4 Step Relay (S)

A Step Relay is the basic component in the Sequential Function Chart (SFC) or Step Ladder (STL). The identifier number of a Step Relay S uses the decimal method, and the functions can be divided into four different types:

(1) Initial Step Relay
It is used in the initial status of the Sequential Function Chart .

(2) General Step Relay
It is for general use in the Sequential Function Chart. When power cut off occurs during PLC operation, the status will be reverted to void (OFF).

(3) Latched Step Relay
When power cut off occurs during PLC operation, the status will be retained. When the power resumes, PLC can continue its work from the Latched Step Relay.

(4) Annunciator Step Relay
The annunciator with latched function. It is mainly driven by the ANS (FNC46) instruction as a contact of the annunciator, used to record the relevant warning messages to facilitate troubleshooting.

- The Step Relays in the VS1, VS2, VSM and VS3 series share exactly the same ranges.

Initial Step Relay	General Step Relay	Latched Step Relay	Annunciator Step Relay
S0~S9, 10 Pt.	S10~S499, S1500~S4095 Total = 3086 Pt.	S500~S899, S1000~S1499 Total = 900 Pt.	S900~S999, 100 Pt.

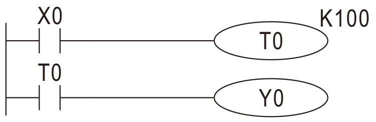
2-5 Timer (T)

- At the VS series PLC, all the timers are handling the time by increase counting.
  When the timer's Present Value (PV) = Set Value (SV, the value is designated to this timer), the timer's contact turns ON.
- The real set time of a timer = Designated Set Value × Unit of this timer
- The Set Value of a timer can be set directly by using a constant number K or indirectly by using the content value in a Data Register D or R. Also, can be modified by using the Index Register V/Z.

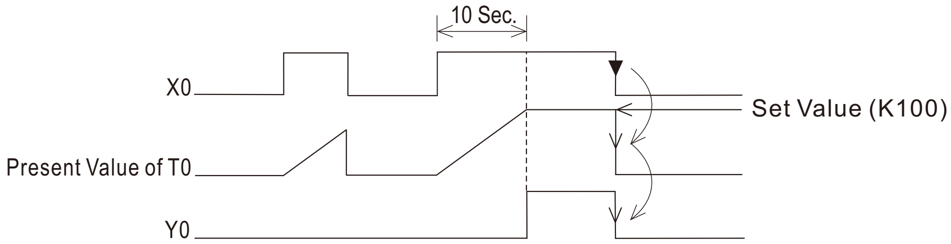
- The definition of timers at the VS1, VS2, VSM and VS3 serie are exactly the same:

General (non-retentive) Timer			**Retentive Timer**	
100ms timer 0.1~3276.7 Sec.	10ms timer 0.01~327.67 Sec.	1ms timer 0.001~32.767 Sec.	1ms timer 0.001~32.767 Sec.	100ms timer 0.1~3276.7 Sec.
T0~T199 200 Pt.	T200~T245 46 Pt.	T256~T511 256 Pt.	T246~T249 4 Pt.	T250~T255 6 Pt.

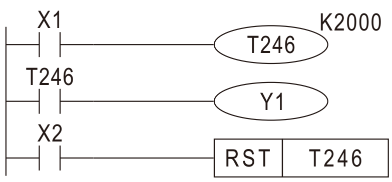
2-5-1 General Timers



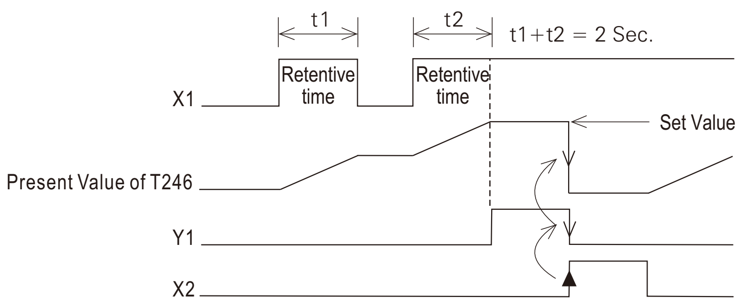
- When X0 = ON, the Present Value of timer T0 starts to up count clock pulse by 100ms, and when its value reaches the Set Value, K100 (10 seconds), its contact will be activated.
- When the input contact X0 = OFF or power cut, the Present Value of the timer will return to “0” and the contact will turn into OFF.



2-5-2 Retentive Timers



- When X1 = ON, the Present Value of timer T246 starts to up count clock pulse by 1ms, and when its value reaches the Set Value K2000 (2 seconds), its contact will be activated.
- During the timer is counting, if the X1 turned OFF or power cut, the T246 will pause counting and retain the Present Value. After the X1 turned ON again, it will resume counting to increase its Present Value continually, until its Present Value = Set Value K2000 (2 seconds) and its contact will become ON.
- When X2 = ON, the Present Value of timer T246 returns to “0” and its contact will be OFF.



2-5-3 Using a Timer in a Subroutine

When the PLC scans to the coil of an general non-retentive timer (not includs T192~T199), it accumulates or clears the Present Value of this timer then controls the timer's contact.
When the PLC scans to the END instruction, it accumulates or clears the Present Value of all the T192~T199 timers then controls the contacts of T192~T199.

At the 1 ms system interrupt, it accumulates the Present Value of activated T246~T249 timers. Then when the PLC scans to the timer's coil, it controls the contact of this timer.

To conclude, reminders for using timers in subroutine are shown below:

(1) A subroutine which will be executed once during each scan time can adopt all types of timers.

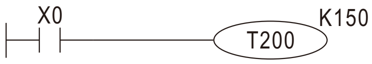
(2) A subroutine which will be executed several times during each scan time can adopt T192~T199 and T246~T249 timers.

(3) The subroutine which may not be executed at every scan time can adopt T192~T199 timers.

(4) An interrupt subroutine (which is only executed in interruption) must adopt T192~T199 timers.

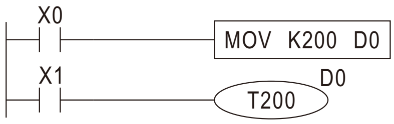
2-5-4 Methods to Appoint the Set Value of a Timer

● Direct setting by using a constant number K



- T200 is a timer using a 10 ms as the time unit.
- When the setting constant K is 150, the Set Value of T200 is 1.5 Sec. (150×10 ms = 1500 ms = 1.5 s).

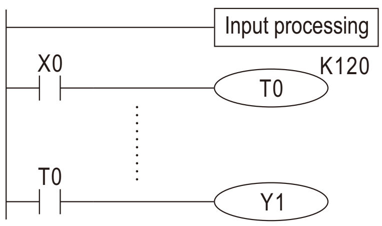
● Indirect setting by using a Data Register D



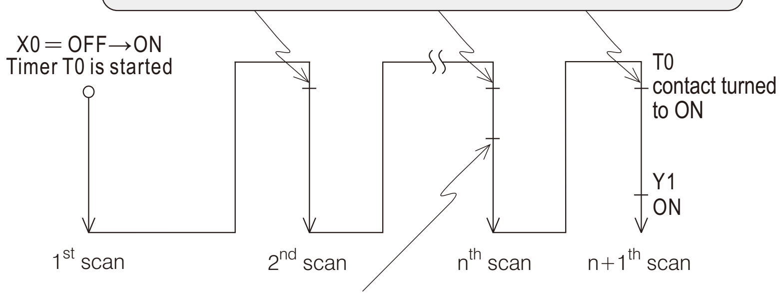
- T200 is a timer using a 10 ms as the time unit.
- If D0 = 200 , T200 = 2 Sec. timer.
- If D0 = 1000 , T200 =10 Sec. timer.
- The Set Value of T200 can be modified by to change the content value of D0.

2-5-5 Detailed Description about the Output Action and Accuracy of a Timer

Below shows the action procedures of a general non-retentive timer:



During the timer T0 is started and the CPU repeatedly to execute its “OUT” coil. That may increase the Present Value of the timer by a few numbers (not only 1) if the period of scan time is more than the timer's resolution unit.



At the moment, the accumulated time is reaching to 12 Sec.
Since the scan of CPU already passed the coil of T0, so the contact of T0 would not be activated.

Below describes the accuracy of a timer, from the coil is triggered to the contact is turned ON:

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

α : 0.001 sec. for the 1 ms timer; 0.01 sec. for the 10 ms timer; 0.1 sec. for the 100 ms timer
T : Set time of timer (sec.)
Ts : Scan Time (sec.)

- If in the program, a timer contact is written in front of its coil, the maximum timing error would be extra 2 Ts. If the set time of timer is “0”, the output contact will action in the next scan.
- For the T246~T249, which is using the system's 1ms interrupt to increase the Present Value if the coil of the timer has been triggered.

2-6 Counter (C)

- When one pulse input signal for a counter is turned from OFF to ON, by the up/down direction control and different type of the counter, the counter's Present Value (PV) will increase (+1, counting up) or decrease (−1, counting down) at the turning edge. If its Present Value (PV) is equal to Set Value (SV), this counter's contact will turn ON.
- The Set Value of a counter can be set directly by using a constant number K or indirectly by using the content value in a Data Register D or R. Also, can be modified by using the Index Register V/Z.
- The characteristics of the 16-bit and 32-bit counters are shown in the following table:

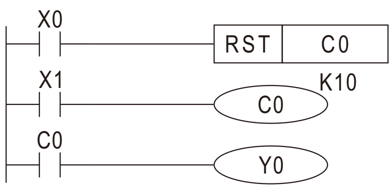
Item	16-bit Counter	32-bit Counter
Counting Direction	Up only	Up or Down (selectable or control by inputs)
Available Set Value	1~32,767 (equivalent to 1, if the Set Value exceeds the range)	−2,147,483,648~ +2,147,483,647
Set Value Appoint	Constant K or a data register	Constant K or data registers (a 32-bit data will occupy 2 continuous data registers)
Changes of Present Value	Increase; retain if reaching the Set Value	Increase / Decrease (not involved to the Set Value)
Status of Contact	Turns ON & retains if the Present Value reaches the Set Value	Up count: turns ON if reaches the Set Value Down count: turns OFF if passes through the Set Value
Reset Action	When the RST instruction is executed, the Present Value becomes “0” and the contact will turn to be OFF.	
Present Value Register	16 bits	32 bits

- The definition of counters at the VS1, VS2, VSM and VS3 serie are exactly the same:

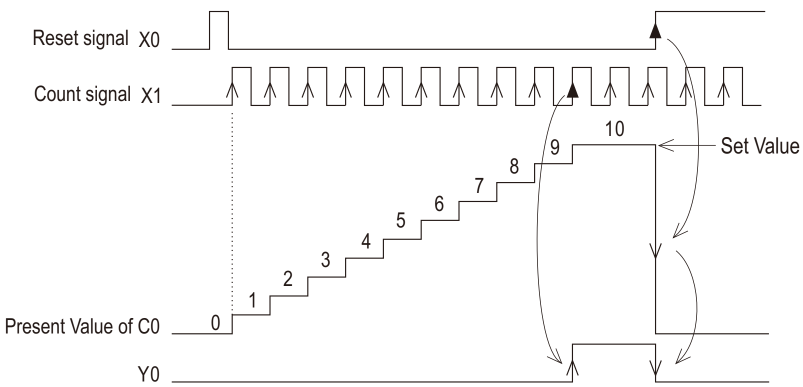
16-bit Counter		32-bit Counter	
General	Latched	General	Latched
C0~C99 100 Pt.	C100~C199 100 Pt.	C200~C219 20 Pt.	C220~C234 15 Pt.

2-6-1 16-bit Counter

- The Present Value of a general counter will be reset when a PLC encounters power cut off. However, the latched counter will retain its Present Value before the power is cut off and starts from there when it is powered.

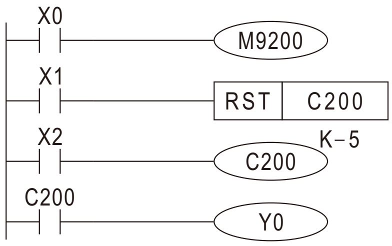


- The contact signal X1 is to drive the C0 counter. When the signal X1 turns from OFF to ON once, the Present Value of C0 will increase by 1. Then, on the tenth input turning ON, C0's output contact is activated, turns ON. After that, the Present Value remains unchanged even the X1 changes.
- When reset signal X0 is ON, the RST instruction operates. The Present Value of C0 is reset to “0” and its contact will become OFF.

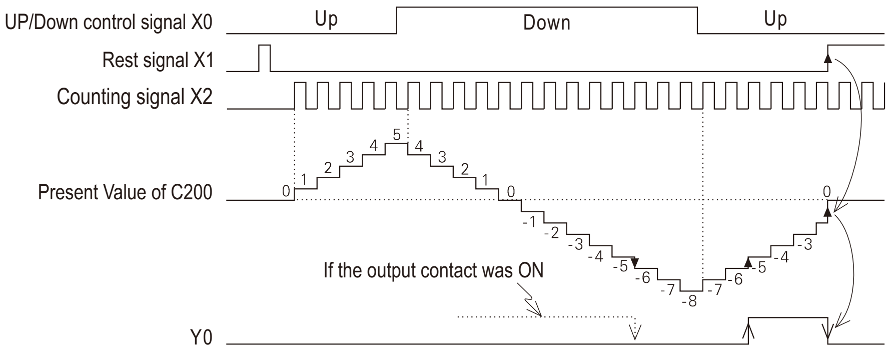


- The counter's Set Value can be set by using a constant K directly or a content value in the Data Register D, R indirectly. Also, this can be modified by the Index Register V/Z.
- When the MOV instruction transfers a new Present Value to the counter and which is greater than its Set Value, the counter's contact will turn to be ON promptly at the next input signal is ON, and meanwhile the Present Value will become the same as the Set Value.

2-6-2 32-bit Counter



- The X0 drives the M9200 to define the up/down count direction of the C200. It OFF is defined as an up count; while ON for down count.
- When the counting signal X2 switches from OFF to ON, C200 accumulates counting, and the Present Value in the register alters.
- When the Present Value of a counter changes from −6 to −5, its output contact switches from OFF to ON.
  On the other hand, when the Present Value of a counter changes from −5 to −6, its output contact switches from ON to OFF.
- When the rest signal X1 is ON, the RST instruction is executed.
  The Present Value of this counter will reset to “0” and its contact will be OFF.

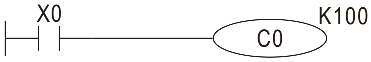


- Since the range of a 32-bit Present Value is between −2,147,483,648 to +2,147,483,647. If a counter up counts beyond +2,147,483,647, the Present Value will automatically change to −2,147,483,648. Similarly, the down counting below −2,147,483,648 will have the result +2,147,483,647. This is a typical “Ring Counter” technique.
- When the power is turned off, the latched counter will retain its Present Value and status of the contact.
- The 32-bit counter can be used as a 32-bit data register.
- If use the DMOV instruction to transfer a number to the counter's Present Value register and which is greater than its Set Value, then at the next input ON signal will let its Present Value accumulated but will not change the status of contact.
- The up/down direction control for a 32-bit counter C200~C234 is defined by the special relay M9200~M9234. The C200's direction is defined by M9200. If the M9200 is OFF, the C200 executes up counting; ON counts down. While the M9201 is to control the direction of the C201, and so on.
- The counter's Set Value can be set by using a constant number K directly or a content value in a pair of continuous Data Registers D, R indirectly. The value can be either a 32-bit positive or negative number.
- In the part of 32-bit counter's output coil, its component ID number is not available to modify by the Index Register V/Z. However, the Set Value can be modified by the V/Z.

2-6-3 Methods to Appoint the Set Value of a Counter

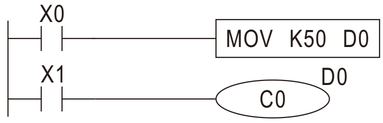
16-bit Counter

- Direct setting by a constant number K



• C0 becomes an up counter with 100 counts.

- Indirect setting by using a Data Register D

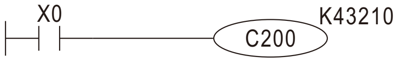


• When D0 = 50, C0 becomes a counter with 50 counts.

• When D0 = 200, C0 becomes a counter with 200 counts. The set number of counts can be modified by changing the value of D0.

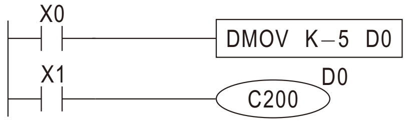
32-bit Counter

- Direct setting by a constant number K



• C200 becomes an up/down counter with the Set Value of K43210.

- Indirect setting by using a double word Data Register D



• When the content's Set Value of a 32-bit register that composed by D1 and D0 is K−5, the C200 becomes an up/down counter with the Set Value of −5. (the D1 is for the upper 16-bit, D0 is for the lower 16-bit).

• The Set Value of the C200 can be modified by to change the content value of the combined D1 and D0.

2-7 Software High Speed Counter

Each one of the input points X0~X7 in the VS series PLC can be used for high speed function, such as the high speed counter, external interrupt or frequency meter. If a X0~X7 is not designed for high speed function, it can still be used as a general input point.

For the purpose of the Software High Speed Counter, it uses the interrupt to receive and count every high speed input pulse, hence with a "Software" in the front of its name. The counter is a 32-bit up/down counter with latched function, and can be classified into three types. Their characteristics are shown in the table below:

Counter's ID No.	Counter Type	Count Direction	Range of Set Value
C235~C245	1-phase high speed counter	Determined by M9235~M9245. OFF is defined as up count while ON as down count.	−2,147,483,648 ~ +2,147,483,647
C246~C250	2- phase high speed counter	The up and down count signals have their own input points. The direction can be identified from M9246~M9250. OFF means up counting; ON means down counting.	
C251~C255	A/B phase high speed counter	Determined by the sequence of A/B phase input signals. Up counting if A-phase signal is ON then B-phase signal turns from OFF to ON. Down counting if A-phase signal is ON then B-phase signal turns from ON to OFF. The direction can be identified from M9251~M9255. OFF means up counting; ON means down counting.	

When input points X0~X7 are applied to perform the Software High Speed Counters (SHSC), there are two operation modes available. The setting page is at the "Project Parameter Setup" within the programming software.

SHSC Mode 1

Input	1-phase High Speed Counter											2-phase High Speed Counter					A/B phase High Speed Counter				
	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	
X1		U/D					R			R		D	D		D		B	B		B	
X2			U/D					U/D			U/D		R		R			R		R	
X3				U/D				R			R			U		U			A		A
X4					U/D				U/D					D		D			B		B
X5						U/D			R					R		R			R		R
X6										S					S					S	
X7											S					S					S

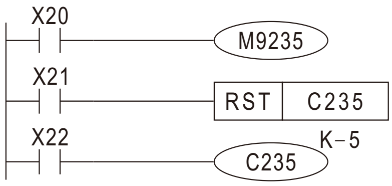
SHSC Mode 2

Input	1-phase High Speed Counter											2-phase High Speed Counter					A/B phase High Speed Counter				
	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				U	A				A
X1		U/D							R			D				D	B				B
X2			U/D							U/D				U		R			A		R
X3				U/D						R			U					A			
X4					U/D						U/D		D					B			
X5						U/D					R			D					B		
X6							U/D								U					A	
X7								U/D							D					B	

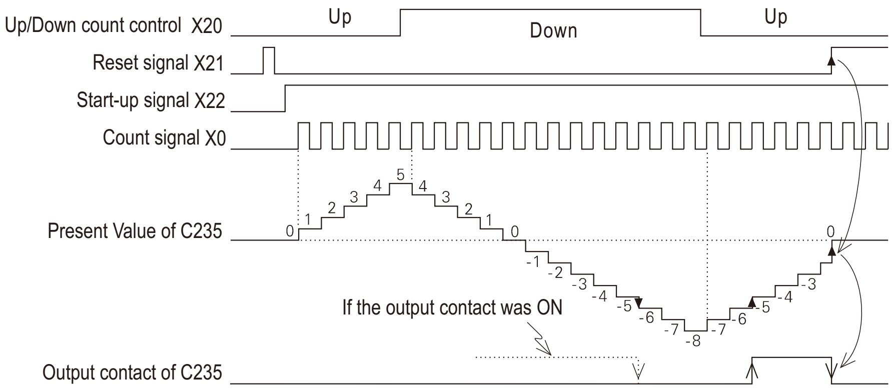
U: Up count input; D: Down count input; A: A-phrase input; B: B-phrase input;
U/D: Up/Down count input; R: Built-in Reset input; S: Built-in Start-up input

- Since a PLC only has 8 high speed inputs X0~X7, once an input point is occupied then it cannot be used for other high speed functions. Therefore, users should plan carefully in order to make good use of these input points.
- This section only describes the software high speed counters; for planning the PLC program, please refer to other sections which are regarded more functions of high speed input points.
- In the sections 2-7-1~2-7-3, the descriptions of various counters are based on SHSC Mode 1.

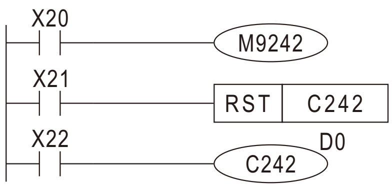
2-7-1 1-Phase High Speed Counter



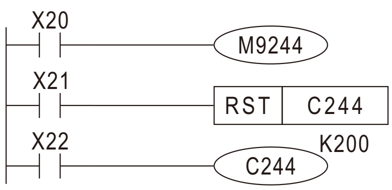
- The X20 drives the M9235 to define the up/down count direction of the C235.
- When the X22 is ON, the C235 will receive that count signal from the X0.
- When the signal X21 is ON, the RST instruction is executed. C235's Present Value is reset to “0” and its output contact becomes OFF.
- The C235~C240 (and the C241~C242 at the SHSC Mode 2) are 1-phase high speed counters which should use the user program to control the activation and reset processes.



- When the start-up signal X22 is ON and there is pulse input from the X0 point, the Present Value of C235 will up/down change.
- When the Present Value of the C235 increases from −6 to −5, its output contact switches from OFF to ON. On the other hand, the Present Value of a counter decreases from −5 to −6, its output contact switches from ON to OFF.
- If the counter up counts accumulated beyond +2,147,483,647, the Present Value will automatically change to −2,147,483,648. Similarly, the down counting below −2,147,483,648 will have the result +2,147,483,647.
- When the rest signal X1 is ON, the RST instruction is executed. Meanwhile, the C235's Present Value is reset to “0” and its output contact becomes OFF.
- The direction of up/down count for an 1-phase counter C235~C245 is determined by the ON/OFF status of related M9235~M9245. If that special relay is OFF, the counter executes up counting; while ON is for down counting.

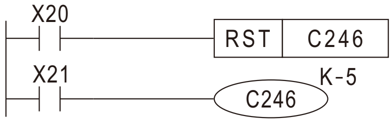


- The X20 drives the M9242 to define the up/down count direction of the C242.
- When X22 is ON, C242 is activated to receive the counting signal from X2.
- When X21 is ON, the RST instruction is executed. The C242's Present Value resets to “0” and its output contact becomes OFF. If the application of C242 is not demanded to reset by the program, could ignore this line.
- When X3 (the designated built-in reset signal of C242) is ON, the Present Value of C242 will be reset to “0” and its contact will become OFF.
- The Set Value of C242 is provided by the content value of D1 and D0.
- The C241~C243 at the Mode 1 or C243~C245 at the Mode 2 are 1-phase high speed counters which should use the program to activate then by the program or built-in reset input to clear.

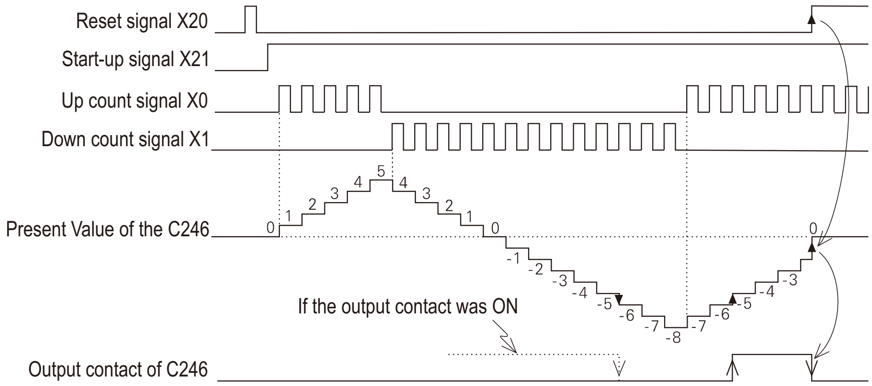


- The X20 drives the M9244 to define the up/down count direction of the C244.
- When X22 is ON and X6 is ON (the built-in start-up signal of C244), the C244 is activated to receive the counting signal from X0.
- When X21 is ON, the RST instruction is executed. The C244's Present Value resets to “0” and its output contact becomes OFF. If the application of C244 is not demanded to reset by the program, could ignore this line.
- When X1 (the designated built-in reset signal of C244) is ON, the Present Value of C244 will be reset to “0” and its contact will become OFF.
- The C244~C245 at the Mode 1 are 1-phase high speed counters which should use the program and start-up input to activate then by the program or built-in reset input to clear.

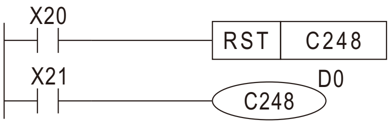
2-7-2 2-Phase High Speed Counter



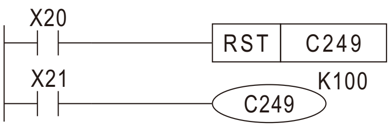
- When X21 is ON, the C246 receives counting signal from either X0 or X1 input point.
- When X20 is ON, the RST instruction is executed. C246's Present Value is reset to 0, and its output contact becomes OFF.
- The C246 2-phase high speed counter which should use the user program to control the activation and reset processes.



- When the start-up signal X21 is ON and there is a pulse entering from either X0 or X1, the Present Value of C246 will up/down change.
  When X0 changes from OFF to ON, the Present Value of C246 increases by 1.
  When X1 changes from OFF to ON, the Present Value of C246 decreases by 1.
- When the Present Value of the C235 increases from −6 to −5, its output contact switches from OFF to ON. On the other hand, when the Present Value of a counter decreases from −5 to −6, its output contact switches from ON to OFF.
- If the counter up counts accumulated beyond +2,147,483,647, the Present Value will automatically change to −2,147,483,648. Similarly, the down counting below −2,147,483,648 will have the result +2,147,483,647.
- The up/down counting direction of a 2-Phase High Speed Counter C246~C250 can be watched by to monitor the M9246~M9250. The related special relay is OFF that means up counting; ON means down counting.



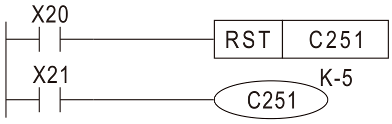
- When X21 is ON, C248 is activated to receive the counting signal from X3 or X4.
  When X3 changes from OFF to ON, the Present Value of C248 increases by 1.
  When X4 changes from OFF to ON, the Present Value of C248 decreases by 1.
- When X20 is ON, the RST instruction is executed. The C248's Present Value is reset to “0” and its output contact becomes OFF. If the application of C248 is not demanded to reset by the program, could ignore this line.
- When X5 (the designated built-in reset signal of C248) is ON, the Present Value of C248 will be reset to “0” and its contact will become OFF.
- The Set Value of C248 is provided by the content value of D1 and D0.
- The C247~C248 at the Mode 1 are 2-phase high speed counters which should use the program to activate then by the program or built-in reset input to clear.



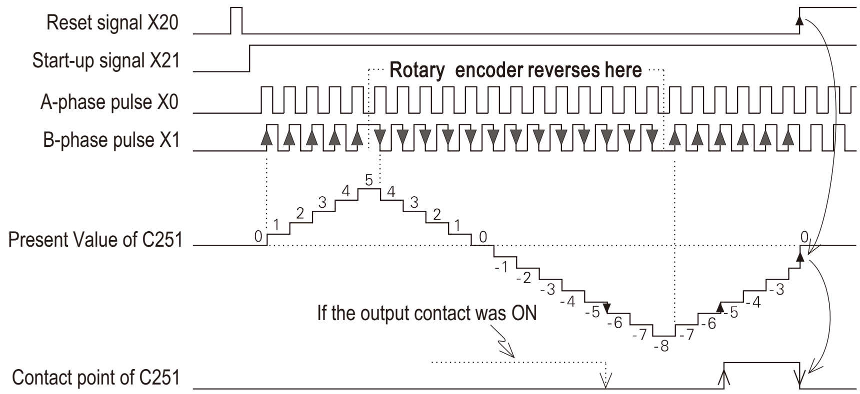
- When the X21 is ON and the X6 (designated start-up input of C249) is ON, the C249 is activated to receive the counting signal from X0 or X1.
  When X0 changes from OFF to ON, the Present Value of C249 increases by 1.
  When X1 changes from OFF to ON, the Present Value of C249 decreases by 1.
- When X20 is ON, the RST instruction is executed. The C249's Present Value is reset to “0” and its output contact becomes OFF. If the application of C249 is not demanded to reset by the program, could ignore this line.
- When X2 (the designated built-in reset signal of C249) is ON, the Present Value of C249 will be reset to “0” and its contact will become OFF.
- The C249~C250 at the Mode 1 are 2-phase high speed counters which should use the program and start-up input to activate then by the program or built-in reset input to clear.

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

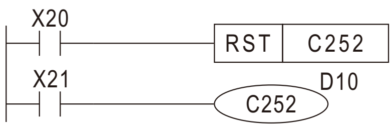
The A/B phase high speed counter is exclusively for receiving A/B phase pulses from a rotary or linear encoder.



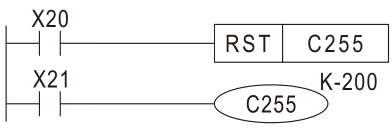
- When X21 is ON, C251 is activated to receive pulse signals from input point X0 (A-phase) and X1 (B-phase). Then bases on the sequence of input signals to perform up/down count.
- when X20 is ON, the RST instruction is executed. C251's Present Value is reset to 0, and its output contact becomes OFF.
- The C251 A/B phase high speed counter should use the user program to control the activation and reset processes.



- When the start-up signal X21 is ON and there are orderly pulses entering from X0 and X1, the Present Value of C251 will up/down change.
  When X0 is ON (A-phase pulse is ON) and X1 turns from OFF to ON (B-phase pulse turns from OFF to ON), the Present Value of C251 increases by 1.
  When X0 is ON (A-phase pulse is ON) and X1 turns from ON to OFF (B-phase pulse turns from ON to OFF), the Present Value of C251 decreases by 1.
- The up/down counting direction of the A/B Phase High Speed Counter C251~C255 can be watched by to monitor the M9251~M9255. The related special relay is OFF that means up counting; ON means down counting.
- The rotary encoder which is connected to a motor shaft that produces corresponding A/B phase pulse according to the motor's forward/backward rotation. This A/B phase pulses are transmitted to C251's A/B phase inputs. Therefore, the Present Value of C251 increases/decreases according to the motor's moving direction.



- When the X21 is ON and A/B phase pulses entering from X0 and X1 for the C251.
  When X0 is ON and X1 turns from OFF to ON, the Present Value of C251 increases by 1.
  When X0 is ON and X1 turns from ON to OFF, the Present Value of C251 decreases by 1.
- When the X20 is ON, the RST instruction is executed. The C252's Present Value is reset to “0” and its output contact becomes OFF. If the application of the C252 is not demanded to reset by the program, could ignore this line.
- When the X2 (designated built-in reset input of C252) is ON, the Present Value of C252 will be reset to “0” and its contact will become OFF.
  The Set Value of C252 is provided by the content value of D11 and D10.
- The C252~C253 at the Mode 1 are A/B phase high speed counters which should use the program to activate then by the program or built-in reset input to clear.



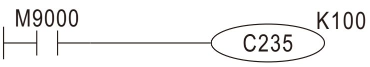
- When the X21 is ON and the X7 (designated start-up input of C255) is ON, the C255 is activated to receive the counting signals from X3 and X4.
  When X3 is ON and X4 changes from OFF to ON, the Present Value of C255 increases by 1.
  When X3 is ON and X4 changes from ON to OFF, the Present Value of C255 decreases by 1.
- When the X20 is ON, the RST instruction is executed. The C255's Present Value is reset to “0” and its output contact becomes OFF. If the application of C255 is not demanded to reset by the program, could ignore this line.
- When the X2 (designated built-in reset input of C255) is ON, the Present Value of C255 will be reset to “0” and its contact will become OFF.
- The C254~C255 at the Mode 1 are A/B phase high speed counters which should use the program and start-up input to activate then by the program or built-in reset input to clear.

2-7-4 Precautions for Using the Software High Speed Counter

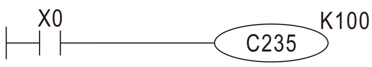
- The VS series PLC is equipped with the Software High-Speed Counters (SHSC) and the Hardware High-Speed Counters (HHSC). Since the Software High-Speed Counters operate in a way of interrupt and thus occupy considerable CPU's capacity and influence its efficiency. In contrast, the Hardware High Speed Counters are equipped with their own hardware circuits in charge of counting, which hardly occupy the efficacy of CPU. At the application of high speed counter, the Hardware High-Speed Counters are recommended.
- The Software High-Speed Counter comparison instructions (HSCS, HSCR, HSZ and HSCT) were to compare and produce results immediately when the related inputs change, then the results could immediately transmit to the specified outputs to drive the load.
  Thus, using the Software High-Speed Counter comparison instruction can get the fastest comparison result. However, because each related input changing is to be compared, that consumes the CPU time and reduces the overall performance. Therefore, after a Software High-Speed Counter is started in the program, according to application requirements user should appropriately make a choice between the use of this counter's output or counter's comparison instructions.
- The use of software high-speed counter comparison instructions in the program has the following limitations:
  (1) Use not more than eight instructions simultaneously among HSCS, HSCR, HSZ and HSCT instructions.
  (2) The HSCT instruction can only be used once in the program.
- Note that the signal source of high-speed counters should not have clutter or even switch bounce signals. They will cause the high-speed counter to count incorrectly.
- When the Software High-Speed Counter is used in the program, the filter time of correlative inputs will automatically adjust to the high speed mode.
- The output contact status of the Software High-Speed Counter is determined by the count operation of the counter input. The state of output contact cannot be affected by to insert a new Present Value which is equal to the Set Value.

To Active the Software High-Speed Counter

- The conditional contact with the program for driving a Software High-Speed Counter is not the same as to use for an input point of a general contour.
  Therefore, please do not use the pulse input point corresponding to the Software High-Speed Counter as the conditional for driving itself in the program. Doing so will make the count error.
- Ideally the special relay M9000 should be used for activation.

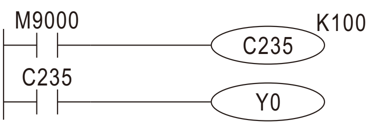


Correct



Incorrect

The Output of Software High-Speed Counter



The Software High-Speed Counter operate by the interrupt, the counting is not dependent to the scan time. Therefore, when the external inputs make the counter's Present Value (PV) = Set Value (SV), the output contact of the counter will change immediately.

However, as shown in the figure above, the contact state of the C235 will be transferred to Y0 when the program is scanned there, and the state of Y0 will be actually sent to the output when the END instruction is executed. It is still relevant to the scan time and is not an immediate output that drives the load.

If an immediate output is desired, the specific Software High-Speed Counter comparison instructions FNC 53 (DHSCS), FNC 54 (DHSCR), FNC 55 (DHSZ) and FNC 280 (DHSCT) can be used.

Finally, using relay outputs will still cause an output delay of about 10ms. If transistor outputs are used, there will be an output delay of less than 1 ms. (depending on the response speed of the output point). This needs to be taken into consideration if necessary.

The Counting Frequency of SHSC

There are two important factors in determining the frequency of SHSC counting:

(1) Limit of hardware circuit: the reaction speed of the photo couplers at high-speed input points X0~X7 and the filter time of input filter circuits, those are limited the maximum pulse frequency can be fed into the PLC. According to the controller series, the pulse frequency is listed in the following table.

Series	X0	X1	X2	X3	X4	X5	X6	X7
VS1	10kHz	10kHz	10kHz	10kHz	10kHz	10kHz	10kHz	10kHz
VS2	50kHz	50kHz	50kHz	50kHz	50kHz	50kHz	50kHz	50kHz
VSM	200kHz	200kHz	50kHz	200kHz	200kHz	50kHz	50kHz	50kHz
VSM-28ML	1MHz	1MHz	50kHz	1MHz	1MHz	50kHz	50kHz	50kHz
VS3	200kHz	200kHz	50kHz	200kHz	200kHz	50kHz	50kHz	50kHz

When these input points are combined for an A/B phase counter, the counting frequency is reduced to half.

(2) Limit of total interruption frequency that PLC system can accept:

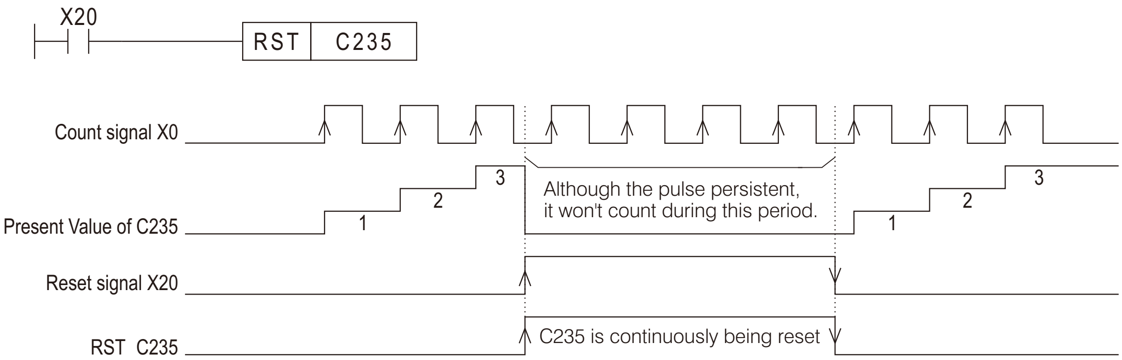
The VS series PLC can accept about 200kHz of total interruption frequency.

(The sum of 1-phase counting frequencies) + (The sum of 2-phase counting frequencies)
+ (The sum of A/B-phase counting frequencies) × 2
= The total of the interrupt frequencies that occur by the Software High-Speed Counter

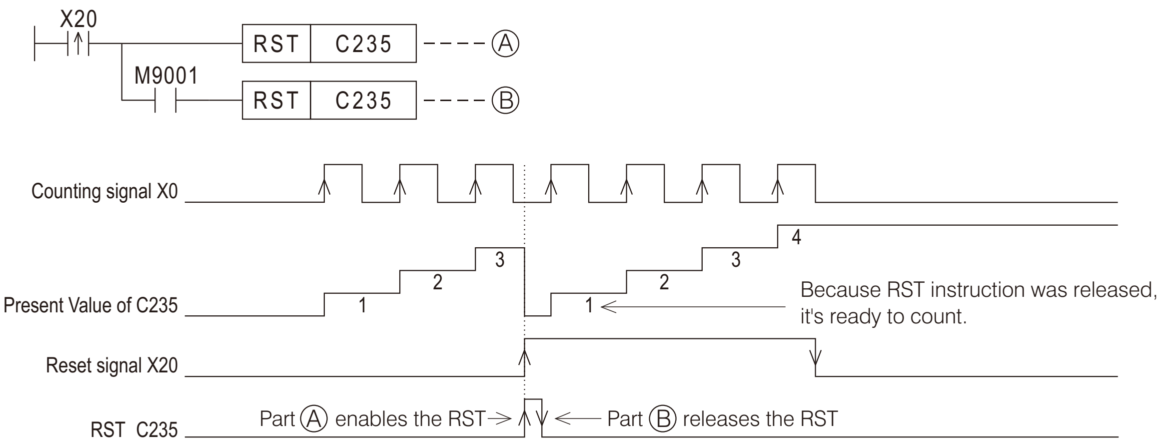
A lot of applications need to use interrupt at the PLC, those including the software HSC, external interrupt, frequency meter, pulse measurement, pulse output instruction, timer interrupt and so on. When considering the total interrupt frequency, these factors should be taken into an overall assessment.

The Notice About to Reset the Software High-Speed Counter

- Usually to reset the Software High-Speed Counter, we use the following program. The counter stops counting when the RST instruction is activated in the example.



- To avoid above situations, please use the following program.



2-8 Data Register (D) and Expansion Register (R)

The characteristics of the Data Register (D) and the Extension Register (R) are identical, so hereafter referred to as a Data Register. The data register is used to store a numerical data. The data length is 16 bits and the Most Significant Bit (MSB) represents a positive/negative sign. A register can store a content value between −32,768 and +32,767. Also, that is possible to combine two 16-bit registers into a 32-bit one; the 16 bits with a smaller number is defined as the lower 16 bits and the larger number is the upper 16 bits. A 32-bit data register can store a content value between −2,147,483,648 and +2,147,483,647, the Most Significant Bit (MSB) represents a positive/negative sign.

The data register can be distinguished by its use as follows:

(1) General Register

- When the PLC turns from RUN to STOP or its power is cut out the data in the register is reset to zero. If the M9033 is ON then the PLC turns from RUN to STOP, the data will retain; however, when the power is cut out, the data will still be cleared and become "0".

(2) Latched Register

- When the PLC's power is disconnected, the data in the Latched Register will be maintained as same as the value stored before the event occurred.
- To clear the contents in the Latched Register, could use the RST or ZRST instruction.
- The main purposes of the Latched Register are to store setup data, record data and the memory of mold parameter.
- When the number of Latched Register is not enough or the stored data has the demand to be transplant, should use the expanded memory card and put the data into the card.

VS series Memory Cards

The VS series PLC provides the VS-MC and VS-MCR memory cards. After the installation of a memory card, 655,360 words of latched data storage space are available. Data can be transferred between the data register and the memory in the card via the data bank write instruction DBWR (FNC 91) and the data bank read instruction DBRD (FNC 90).

Since the memory card uses Flash ROM to store data, the number of writes is limited to 100,000 times. The improper use may shorten the lifespan of the Flash ROM. Therefore, use the DBWRP to substitute the DBWR instruction in a program to write data, that could avoid unnecessary operations and extend the lifespan of the Flash ROM.

(3) Special Register

- Each Special Register has its own specific function, the main usage is as the storage of system status, error code or status monitoring. For details, please refer to the section 2-14 "Special Relay and Special Register".

(4) Extension Register

- When the PLC turns from RUN to STOP or its power is cut out the data in the register is reset to zero.
  If the M9033 is ON then the PLC turns from RUN to STOP, the data will retain; however, when the power is cut out, the data will still be cleared and become "0".

● The function and discrimination of registers D in the VS1, VS2, VSM and VS3 series are exactly the same:

Series	General Register	Latched Register	Special Register	Extension Register
VS1	D0~D6999 Total = 7000Pt.	D7000~D8999 Total = 2000 Pt.	D9000~D9511 Total = 512 Pt.	R0~R9999, Total = 10000 Pt.
VS2				
VSM				
VS3				R0~R23999, Total = 24000 Pt.

2-9 Index Register (V and Z)

Index Register V, Z is a very special register in the VS series PLC. Its purpose is to use the index to modify the operand in an instruction, to serve the purpose of specifying the operand indirectly and exchangeable, thereby improving the flexibility and efficiency of program editing.

The index register provides the ability to specify the operand with flexibility when the instruction is being executed. That providing great help in the preparation of complex control program, and can often simplify the process. Here are some possible applications for reference:

(1) At the Subroutine. There is generally a need for a subprogram to perform the same function repeatedly for different operands.

(2) The instruction in the program has limitation about the used number of times.

(3) When the data in bulk needs process, the source or target data can be specified by the index register.

- The index register is a 16-bit register, numbered from the V0 to the V7 and from the Z0 to the Z7, 16 points in total. Each index register can be reused for different instructions, that just need to give a correct content value to the index register before the instruction is executed.

- V, Z registers can be paired up to form a 32-bit register. In the 32-bit application instruction, V, Z registers should be paired up as (V0, Z0) (V1, Z1)..... (V7, Z7). However, when the paired index register is used to modify a 32-bit operand, that syntax is to put the Z register behind the operand (the V register is hidden).

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

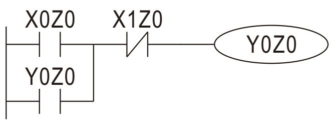
In a 32-bit instruction, only the Z0 is needed to be specified.

2-9-1 Using Index Register in Basic Instruction

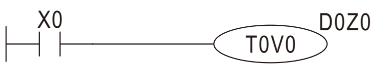
- The Index Registers can be used to modify the operand in a basic instruction, the modifiable components are shown below:
  The Set Value of a T: when it uses K, D or R at the OUT coil.
  The Set Value of a C: when it uses K, D or R at the OUT coil.

- Here provide some examples that Index registers V, Z modify operands:
  When Z0 = 10, X0Z0 = X12 (The X is named by octal number system)
  Y5Z0 = Y17 (The Y is named by octal number system)
  M10Z0 = M20
  S2Z0 = S12
  K100Z0 = K110
  D0Z0 = D10

- Examples of using index register in basic instructions:



As shown in the left diagram it is a self-holding circuit.
When Z0 = 0, that triggers by X0, releases by X1 and has an output at Y0.
When Z0 = 5, that triggers by X5, releases by X6 and has an output at Y5.



When Z0 = V0 = 0, X0 drives the T0 timer and D0 is the Set Value.
When Z0 = 1 and V0 = 2, X0 drives the T2 timer and D1 is the Set Value.

2-9-2 Using Index Register in Application Instruction

- The Index Registers can be used to modify the operand in an application instruction, the modifiable components are shown below:

  Bit component: X, Y, M, S

  Pointer: P, Q (P as the label name of the jump or subroutine cannot be modified)

  Word component: The Present Value of a T or C
  Index register D, R
  The K*n*X, K*n*Y,, K*n*M, K*n*S which is composed by X, Y, M, S (K*n* itself can not be modified)
  The UnG at the part of G (Un itself can not be modified)

  Constant: K, H

- When using the index register in a 32-bit application instruction, be sure to use paired V, Z registers. At this point, pay special attention to whether there is residual value in the upper register V. To be safe, use the DMOV instruction when placing value into the V, Z paired registers.

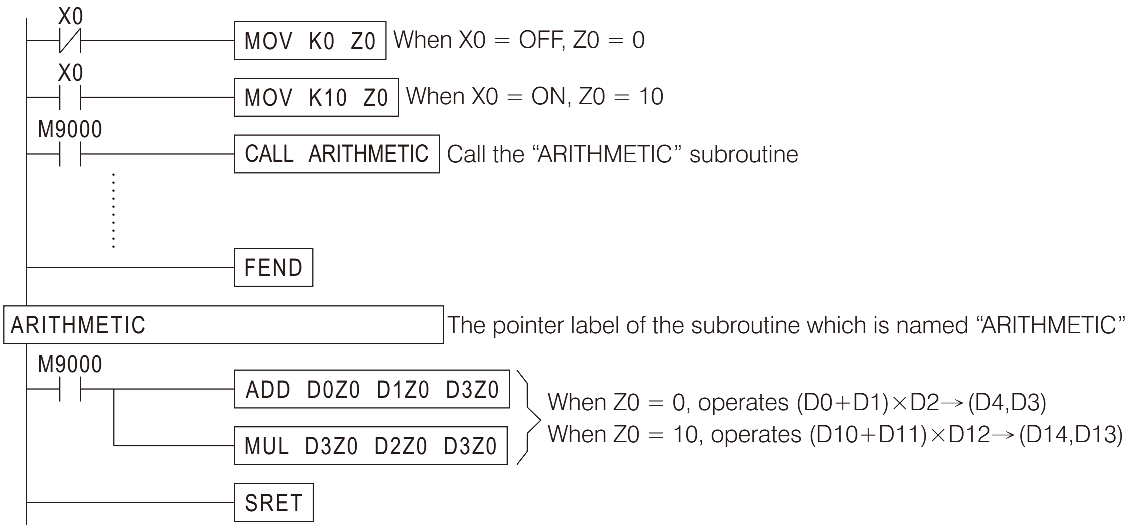
- Here provide some examples that Index registers V, Z are used to modify operands:

  ① 16-bit instructions, when Z0 = 4,
  Y27Z0 = Y33 (The Y is named by the octal number system)
  T5Z0 = T9
  D0Z0 = D4
  K4M8Z0 = K4M12
  U1G0Z0 = U1G4

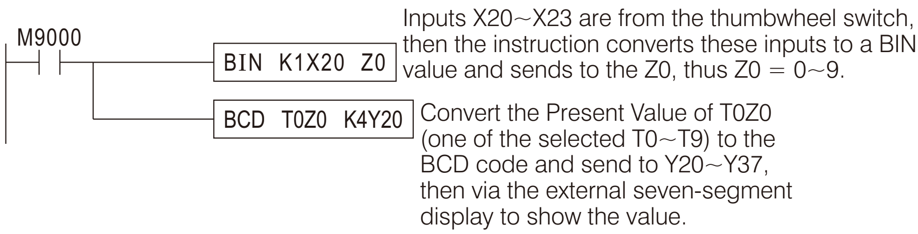
  ② 32-bit instructions(will occupyV,Zregisters), when (V1, Z1)=8,
  X20Z1 = X30 (The X is named by the octal number system)
  M0Z1 = M8
  D0Z1 = D8
  K8M40Z1 = K8M48
  R10Z1 = R18

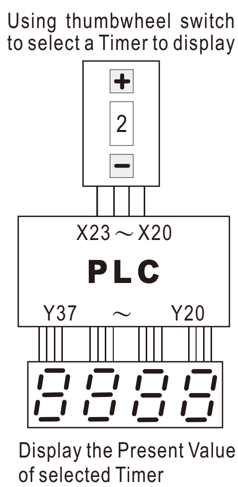
2-9-3 Demonstration Program Using Index Register

- Using index register in a subrountine

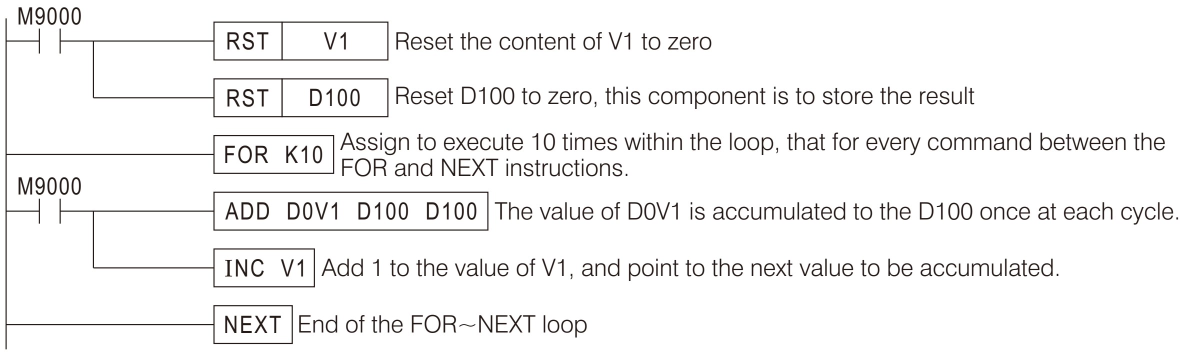


- With the following program and external wiring, you can use the coded inputs to change the value of Z0. Then, can select one of the Present Values in T0~T9 and display the number in the external seven-segment display.





● To add up all values in D0~D9 and store the result into the D10.

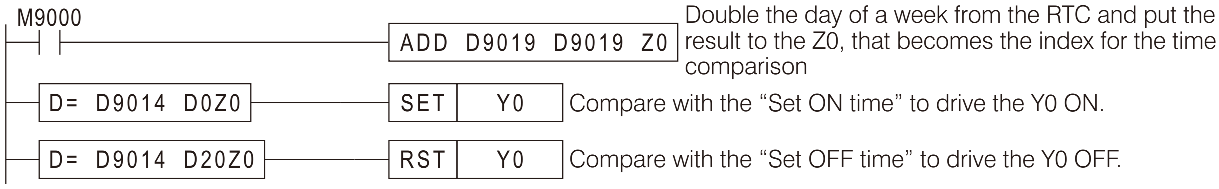


If you want to get the result of the sum (D0~D99) at the D100 instead in the above program, only need to change the K10 in FOR K10 to K100.

● The sprinkler system sets different opening and closing times from Sundays to Saturdays. The following example uses the set ON time at D0~D13 and the set OFF time at D20~D33 to drive the sprinkler motor by the output Y0. The time data of the Real-Time Clock (RTC) is compared with the setting value of the time schedule to determine the ON / OFF of the sprinkler motor.

Set ON time Hr.	Set ON time Min.		Set OFF time Hr.	Set OFF time Min.
D1	D0	Sun.	D21	D20
D3	D2	Mon.	D23	D22
D5	D4	Tue.	D25	D24
D7	D6	Wed.	D27	D26
D9	D8	Thu.	D29	D28
D11	D10	Fri.	D31	D30
D13	D12	Sat.	D33	D32

The time data of RTC:
D9019 —— 0 (Sun.)~6 (Sat.)
D9014 —— Minute
D9015 —— Hour



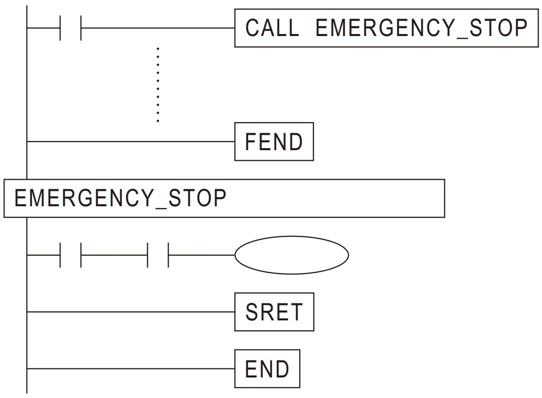
The simple program above can achieve the requirement easily, isn't it? In fact, dealing with a large number of regular data, the most important thing is to select an appropriate data structure (method for data storage). Then, with the characteristics of the index register, you can compile efficient programs.

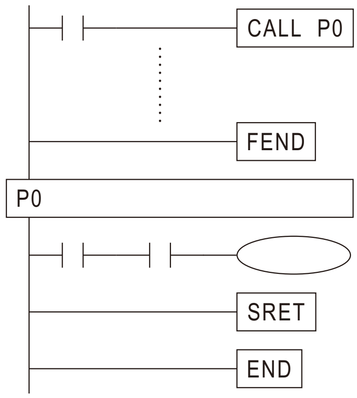
2-10 Mark Pointer and Branch Pointer (P)

The purpose of the Mark Pointer and Branch Pointer (P) is to specify a specific point in the program. Usually they are used to indicate the destination of the CJ instruction, or the start position of a subroutine of the CALL instruction.

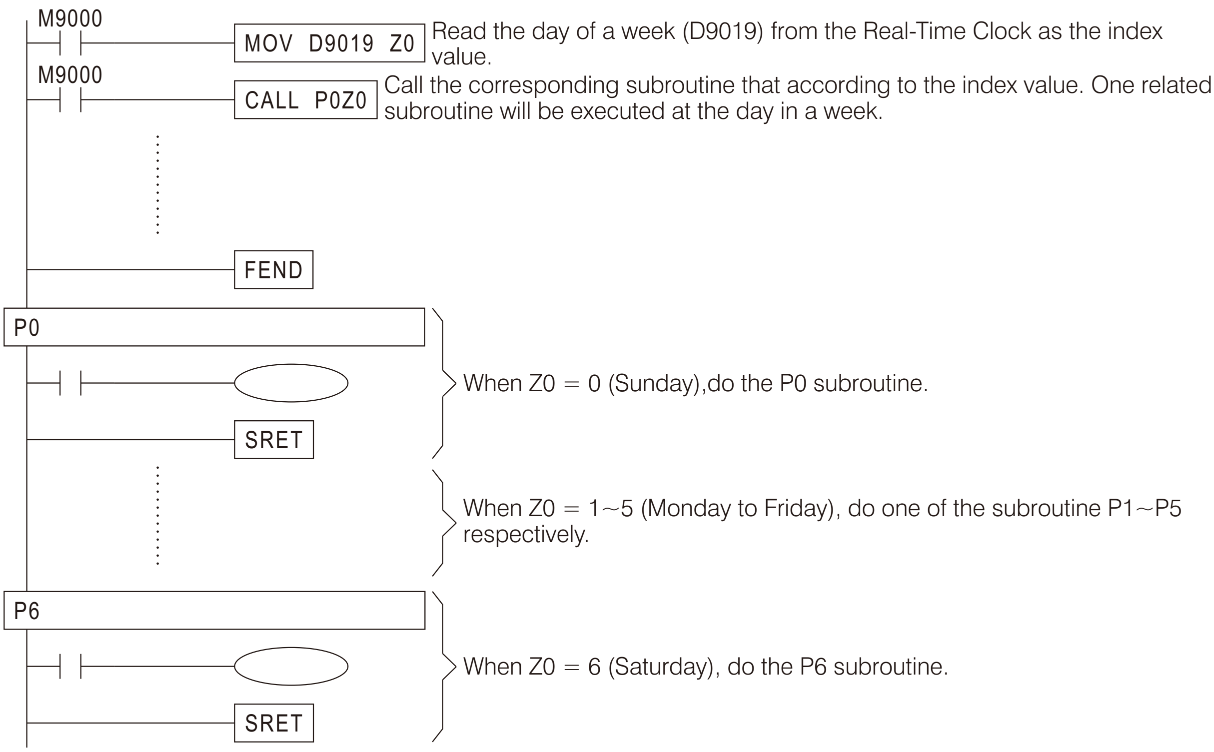
In the past, the VB series PLC only had Branch Pointer P, which indicates a specific location with indication number, but the program's readability was disadvantageous. Therefore, the VS series PLC newly enhances the Mark Pointer indicator function to enable programmers use illustrative text to indicate a specific address, thus greatly increases the readability.

- The ID numbers of Branch Pointer P in the VS1, VS2, VSM and VS3 series are exactly the same: P0~P1023, 1024 points in total.
  - The Branch Pointer P63 or P255 is equal to the position of END in the program.
- The Mark Pointer is made up of 16 English characters or numbers, and its use is exactly the same as the Branch Pointer P.
- A maximum of 1024 Pointers P (including the Mark / Branch Pointers and Interrupt Pointers) in the program can be used.
- The ID number at Branch Pointer P can be modified by the V, Z Index Register; texts in a Mark Pointer cannot use the V, Z Index Register. By the characteristics, choose the suitable one at the program.
- Among the exemplary programs are the Mark Pointer (left) and Branch Pointer P (right), the left one that uses the Mark Pointer can acquire better readability.





- An example of using Branch Pointer P and combine with Index Registers V, Z.



2-11 Table Nickname and Table Code (Q)

PLC users sometimes have to set up a lot of data in order to cope with needs such as formulas, control parameters and communication commands.

In the past, people often set up the necessary reference materials through programming.

Not only a lot of program capacity and manpower were consumed, but also the data content established mostly had poor readability, and were difficult to change and maintenance.

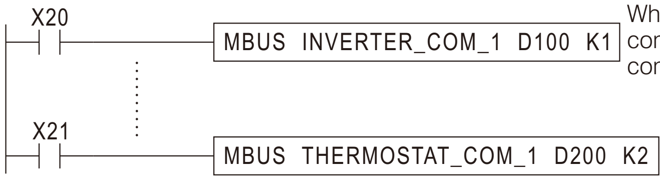
In order to meet the demands, the VS series PLC has provided the “Table” as a data source form. The table is a data set by collecting data of relevant characteristics, such as data table, MBUS communication table, LINK communication table. Each table has its own specific type and purpose. We use the computer programming software to provide the tables' editing interfaces, so these tables are easy to establish and manage, furthermore can significantly improve programming efficiency. The editing method of the table will be described in the programming software.

Those tables will occupy the memory space in the PLC. When the programming software writes a project to a PLC, tables will be written into the memory with the program together. Besides, to read a project from a PLC, the tables are read out from the its memory with the program together. The tables are a part of the PLC project.

At the VS series PLC, every table has its own Table Nickname or Table Code Q to identify. Then, the related application instruction can use that Table Nickname or Table Code Q to get a particular table.

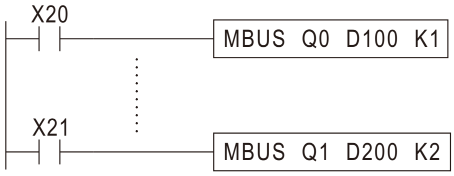
Table Codes Q0~Q31 are marked with index numbers; while the Table Nicknames are composed of 16 English characters or numerals. The Table Nicknames have better readability. The drawback is that they cannot modify by Index Register V, Z. In contrast, Table Codes Q0 to Q31 have less readability but they are irreplaceable when used with the Index Register V, Z.

- The ID numbers of Table Code in the VS1, VS2, VSM and VS3 series are exactly the same.
- The Table Nicknames are made up of 16 English characters or numbers.
- A maximum of 32 Tables in the program can be used.
- Among the exemplary programs are making use of the Table Nickname and Table Code, the upper one that uses the Table Nickname can acquire better readability.



When X20 = ON, CP1 executes the MODBUS communication of which the content is based on the communication table of “INVERTER_COM_1”

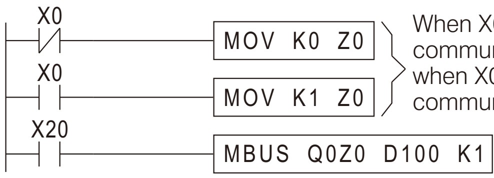
When X21 = ON, CP2 executes the MODBUS communication of which the content is based on the communication table of “THERMOSTAT_COM_1”



When X20 = ON, CP1 executes the MODBUS communication of which the content is based on the communication table at the Q0.

When X21 = ON, CP2 executes the MODBUS communication of which the content is based on the communication table at the Q1.

- Example of using Table Code Q and Index Register V, Z.



When X0 = OFF, the following MBUS instruction s to execute the communication table at the Q0;
when X0 = ON, the following MBUS instruction is to execute the communication table at the Q1.

CP1 executes the MODBUS communication. With the index modify function, it can choose an expected table to execute the communication.

2-12 Interrupt Pointer (I)

In general, the PLC executes user program in the way of sequential scanning. Even the system itself also follows a sequence of execution (receive external inputs → process the user program → output the computed results). However, such an ordinary operation is occasionally unable to meet the needs for quick control responses. Therefore, the interruption function is generated which meets the required of to cut-in a processing sequence immediate.

The interrupt, as the name suggests, is to break the sequentially executed program, and then insert a program section to be processed immediately. In the VS series PLC, every Interrupt Pointer is bound up with its interrupt subroutine to deal with the user's requirement for the insertion of interruption.

- The purpose of the Interrupt Pointer is to specify the start position of interrupt subroutine in the program.
- The ID numbers of Interrupt Pointer I in the VS1, VS2, VSM and VS3 series are exactly the same:

External Interrupt			Timer Interrupt		Software HSC Interrupt		Hardware HSC Interrupt	
External Input	Interrupt Pointer	Inhibit Flag	Interrupt Pointer	Inhibit Flag	Interrupt Pointer	Inhibit Flag	Interrupt Pointer	Inhibit Flag
X0	IX0□	M9050	ITA□□	M9058	IHC0	M9061	IHHC1	M9062
X1	IX1□	M9051			IHC1			
X2	IX2□	M9052			IHC2			
X3	IX3□	M9053	ITB□□	M9059	IHC3			
X4	IX4□	M9054			IHC4		IHHC2	M9063
X5	IX5□	M9055			IHC5			
X6	IX6□	M9056	ITC□□	M9060	IHC6			
X7	IX7□	M9057			IHC7			
□=P means interrupt on the rising edge □=F means interrupt on the falling edge			□□=01~99 means the interval time is 1~99ms		The interrupt is processed with the DHSCS instruction.		The interrupt is processed, when the Present Value reaches to the Set Value.	

- Each interrupt pointer has an inhibitory special relay to control the interrupt, the user can avoid the interrupt by activating the corresponding special relay.
- By the characteristics, those interrupt pointers can be divided into external interrupt, timer interrupt, software high-speed counter interrupt and hardware high-speed counter interrupt.

① External Interrupt:

The rising or falling signal from the specific input point (X0 to X7) generates an interrupt signal to interrupt the program in execution, jump to a designated interrupt pointer (IX0 □ to IX7 □) and execute a corresponding interrupt subroutine. The External Interrupt at the VS series PLC also has the delay action function. Please refer to “2-15-1 External Interrupt” for more details.

② Timer Interrupt:

When the timer interrupt pointer (ITA □□, ITB □□ or ITC □□) is written into the program, the PLC will automatically interrupt the program execution at a given time (as defined by the □□ in the interrupt pointer). Its procedure jumps to the appointed interrupt pointer and executes the interrupt subroutine.
Timer interrupt is mainly used to generate a repeated interrupt that could execute fixed and rapid period timing subroutine. When a certain section is requiring an execution cycle shorter than the PLC's scan time or a fixed time cycle, the timer interrupt is considered.
For example, the HKY (FNC71) and SEGL (FNC74) instructions can use the PLC's scan time as the scan cycle of the instructions. However, too long or too short scan time may cause fault. In this case, a timer interrupt subroutine can be used to perform appropriate scan operation.
Additionally, the RAMP (FNC67) instruction generally depends the scan time of the program to bring the movement of ramp steps forward. Usually, the scan time is unregulated, thus the generated ramp result will become irregular. In this case, a subroutine of timer interrupt can be used for the RAMP instruction to depart it from the scan time, that may stabilize the movement of step thus to produce the regular ramp result.

③ Software High Speed Counter Interrupt:

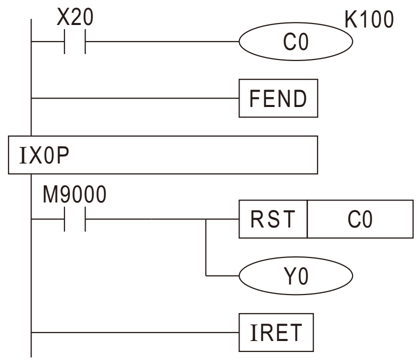
The result of a comparison instruction by the FNC53 (DHSCS) high-speed counter can be assigned to execute an interrupt subroutine. When the DHSCS instruction is assigned to execute a certain interrupt subroutine (IHC0~IHC7) and the comparison result is equal, then PLC will jump to the specified interrupt pointer and execute the interrupt subroutine. Please refer to the FNC53 (DHSCS) Instruction for more details.

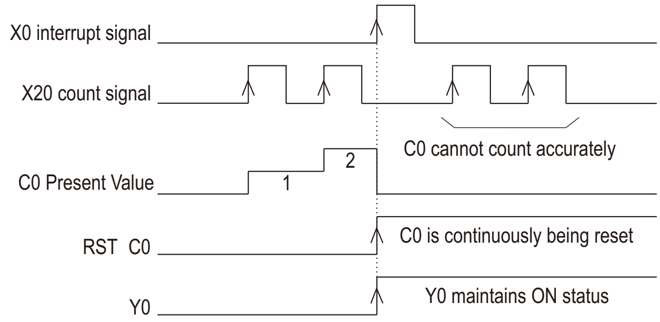
④ Hardware High Speed Counter Interrupt:

When the hardware high-speed counter HHSC1 reaches its Set Value by the external input, the IHHC1 interrupt can be generated; also, the HHSC2 can generate the IHHC2 interrupt. Please refer to “2-15-4 Hardware High-Speed Counter”.

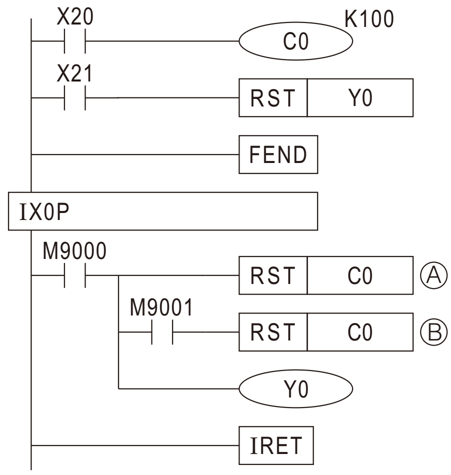
- The interrupt subroutine often needs to immediately read the external state or immediately drive the external load. In this case, the I / O update instruction REF (FNC 50) should be used in the interrupt subroutine to update the I / O status promptly.
- In the interrupt subroutine, should use the Timer T192~T199 when a timer is necessary.

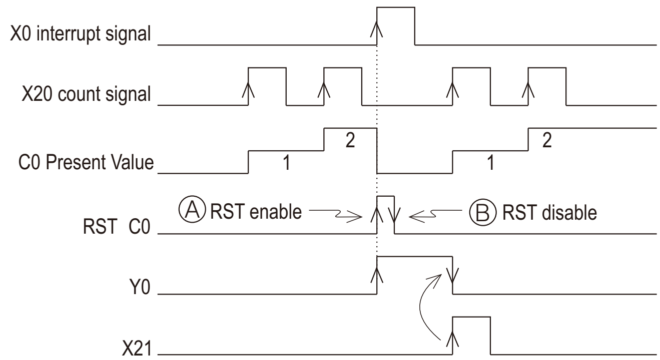
- The interruption is a special mechanism for to break the rules of regular operation; an interrupt subroutine is not regularly implemented. Therefore, special attention must be paid to the components that are driven in the interrupt subroutine.
  - If a step relay is for to active a part of the SFC, it is not allowed to be driven in an interrupt subroutine.
  - The following example shows that the elements driven in the interrupt subroutine remain in their state.





  - The following example shows the modified program.





- The applications of Interrupt Pointer and the concepts of interrupt subroutine are explained in detaile in the section of Application Instructions IRET, EI and DI.

2-13 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”.

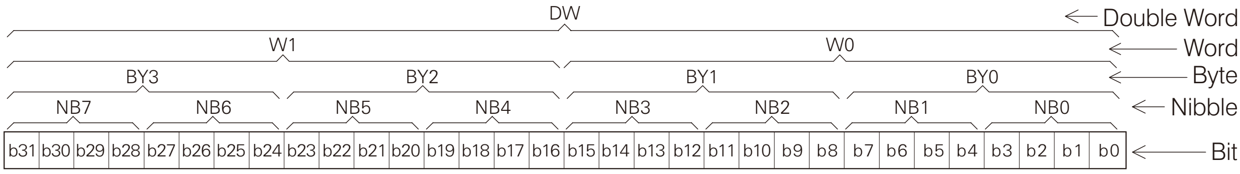
② Nibble: composed of 4 sequential bits.
For example, b3 ~ b0 can express an one-Nibble hex value: 0 ~ F.

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

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

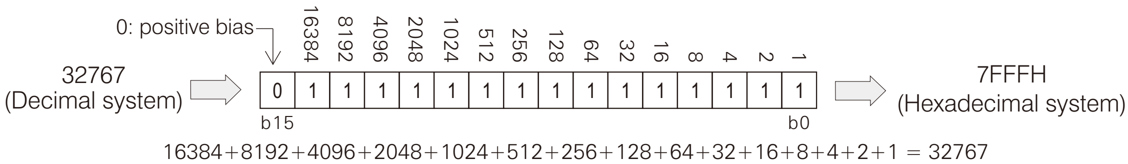
⑤ Double Word: composed of 2 sequential words, 4 sequential bytes or 32 sequential bits.
For example, b31 ~ b0 can express an eight-Nibble hex value: 00000000 ~ FFFFFFFF.

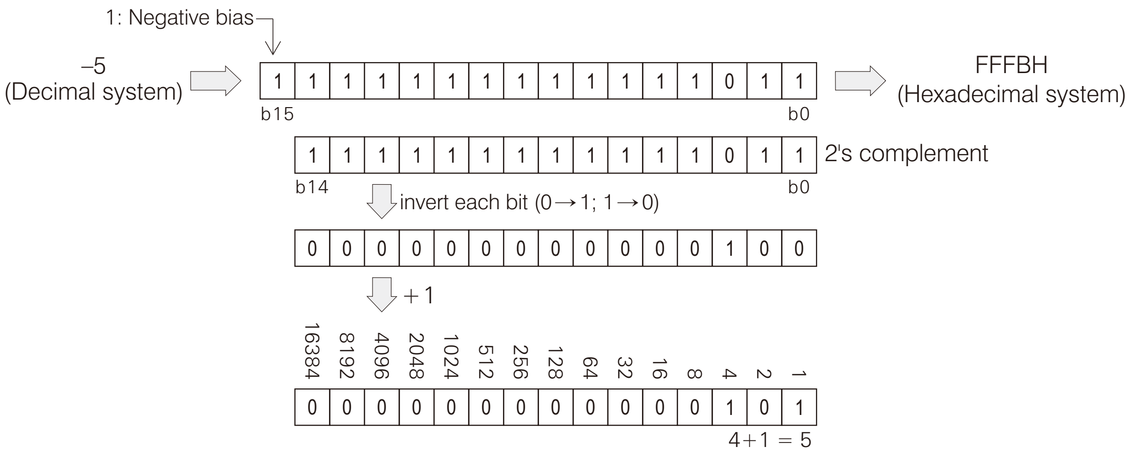
⑥ The relations between every binary Bit, Nibble, Byte, Word and Double Word:



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 quantity of the value.
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) Octal Number (OCT)

The assigned numbers of PLC's external input and output terminals are displayed by the octal system.
For example,
The external input ports: X0 ~ X7, X10 ~ X17
The 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. For example, K123 indicates a decimal number where the value is 123.
Application occasions of Decimal Number:

① Used as the Set Value of the T and C, for example, K10

② Used as the component number of the M, S, T or 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:

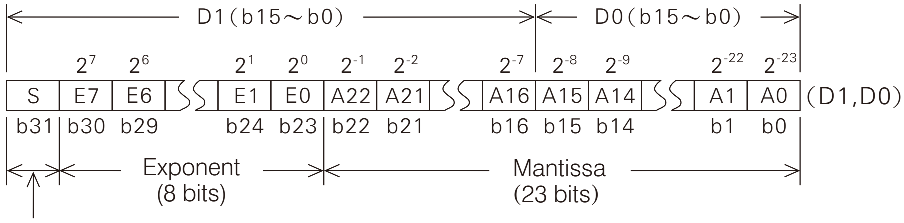
OCT	DEC	HEX	BIN		BCD	
0	0	00	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	1	01	0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1
2	2	02	0 0 0 0	0 0 1 0	0 0 0 0	0 0 1 0
3	3	03	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 1
4	4	04	0 0 0 0	0 1 0 0	0 0 0 0	0 1 0 0
5	5	05	0 0 0 0	0 1 0 1	0 0 0 0	0 1 0 1
6	6	06	0 0 0 0	0 1 1 0	0 0 0 0	0 1 1 0
7	7	07	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 (Real number)

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.



Mantissa Sing bit (1=Negative, 0=Positive)

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:

  $-1.0\times2^{128}\sim-1.0\times2^{-126}$, 0, $1.0\times2^{-126}\sim1.0\times2^{128}$
  Minimum modulus: 1.175×10^{-38} Maximum modulus: 3.402×10^{38}

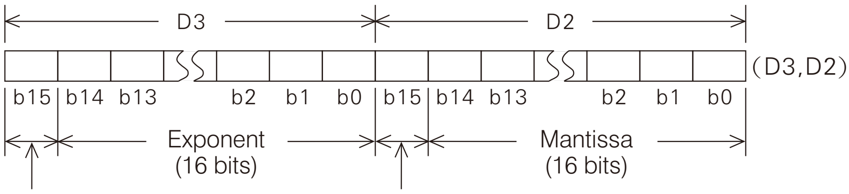
  In the PLC program, the real constant value is usually preceded by the "E". The effective range of real constants is $-3.402\times10^{38}\sim3.402\times10^{38}$, which can be expressed by the decimal point or exponent.

  By the decimal point: Directly use decimal point to express the value, for example, "E102.35" represents "102.35".

  By the exponent: Use the mathematical exponent to express, for example, "E1.0235+2" represents "102.35"; "+2" means to multiply by 10^{+2}.

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.



Exponent Sing bit (1=Negative, 0=Positive) MSB

Mantissa Sing bit (1=Negative, 0=Positive) MSB

Decimal Floating Point Number's value = (Mantissa) $\times10^{(\text{Exponent})}$
Mantissa $=\pm(1000\sim9999)$ Exponent $=-41\sim+35$

- 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:
  Minimum modulus: 1175×10^{-41} Maximum modulus: 3402×10^{35}

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

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

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

2-14 Special Relay and Special Register

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

2-14-1 Table of Special Relay

Relay ID No.	Description	Series			
PLC Operation Status		VS1	VS2	VSM	VS3
■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 either one of the error flag relay M9066, M9067 is "ON", M9004="ON" .	○	○	○	○
Clock Pulse or RTC		VS1	VS2	VSM	VS3
■M9011	Alternate pulse by the period of 10ms cycle time. "ON" 5ms/"OFF" 5ms Pulse	○	○	○	○
■M9012	Alternate pulse by the period of 100ms cycle time. "ON" 50ms/"OFF" 50ms Pulse	○	○	○	○
■M9013	Alternate pulse by the period of 1sec. cycle time. "ON" 0.5sec./"OFF" 0.5sec. Pulse	○	○	○	○
■M9014	Alternate pulse by the period of 1min. cycle time. "ON" 30sec./"OFF" 30sec. Pulse	○	○	○	○
M9015	Pause the RTC and write the values in D9013~D9019 to the RTC	○	○	○	○
M9016	Stop reading time data from the RTC	○	○	○	○
M9017	Modify RTC ±30sec.	○	○	○	○
■M9018	M9018="ON" when RTC is installed in the Main Unit.	○	○	○	○
■M9019	Write wrong data onto the RTC	○	○	○	○
Flag		VS1	VS2	VSM	VS3
■M9020	Zero Flag. M9020="ON" if the result of an arithmetic instruction is "0". (except the MUL and DIV)	○	○	○	○
■M9021	Borrow Flag. M9021="ON" if any "Borrow" occurs by the addition or subtraction instruction.	○	○	○	○
M9022	Carry Flag. M9022="ON" if any "Carry" occurs by the arithmetic, rotary.... instruction.	○	○	○	○
■M9023	Zero Flag. M90023="ON" if the result of multiplication (MUL) or division (DIV) is "0".	○	○	○	○
■M9025	Overflow Flag. M90025="ON" if the result of division (DIV) is overflowed.	○	○	○	○
■M9029	Instruction execution completed flag. M9029="ON" when the executions of some applied instructions are completed (please refer to the relevant instructions).	○	○	○	○
■M9090	All bits "ON" flag at the result of a block data comparison BKCMP (FNC194~FNC199) instruction.				○
■M9162	To indicate the completion of HSCT instruction.	○	○	○	○
■M9163	External Interrupt delay time set-up flag. Use this flag contact to active the interrupt delay function.	○	○	○	○
PLC System Operation Mode		VS1	VS2	VSM	VS3
M9031	Clear the Non-Latched area memory. / States and contents of devices are reset at the "END". All Coils Y, M, S, T, C turn "OFF" and content values of T, C, D become "0"; But, the Special M and D will not be changed.	○	○	○	○
M9032	Clear the Latched area memory. / States and contents of devices are reset at the "END". All Coils Y, M, S, T, C turn "OFF" and content values of T, C, D become "0"; But, the Special M and D will not be changed.	○	○	○	○
M9033	When M9033="ON" and RUN→STOP, the content value and statuses of T, C, D are retained.	○	○	○	○
M9034	All the outputs are disable. When M9034="ON", all external outputs are forced to "OFF" but the Y contacts for the program still operate normally.	○	○	○	○
M9039	To fix PLC's Scan Time duration. When M9039="ON", the PLC within a constant scan duration and the period is allocated by the D9039.	○	○	○	○
Assigning Specification of Applied Operation Instructions Mode		VS1	VS2	VSM	VS3
M9024	Assign the BMOV moving direction. When M9024="OFF", S → D; when M9024="ON", S ← D.	○	○	○	○
M9026	Assign the RAMP operating mode. When M9026="OFF", a series of ramp process will be executed; when M9026="ON", one trigger signal will ramp once only.	○	○	○	○
M9027	Assign the PR operating mode. Please refer to PR (FNC 77) Instruction for details.	○	○	○	○
M9028	To protect not to operate the FROM/TO instruction repeatedly. When M9028="OFF", disallows interrupt during FROM/TO is in operation. When M9028="ON", FROM/TO in an interrupt subroutine is ineffective.		○	○	○
M9035	Assign the PWM operating time base. When M9035="OFF", the parameters for the PWM are by the unit of 1ms.; when M9035="ON", by the unit of 0.1ms.	○	○	○	○

Relay ID No.	Description	Series			
Assigning Specification of Applied Operation Instructions Mode		VS1	VS2	VSM	VS3
M9091	Assign the BINDA operating mode. 16-bit instruction: If M9091=“OFF”, will add the end of string 0000H after the result. If M9091=“ON”, will only convert the data without to add the end of string. 32-bit instruction: If M9091=“OFF”, will add the end of string 00H at the result's last upper 8 bits. If M9091=“ON”, will add the end of string 20H at the result's last upper 8 bits.				○
M9160	Assign the XCH to execute the SWAP function.	○	○	○	○
M9161	Assign the 8-bit or 16-bit operating mode for the related instructions. When M9161=“OFF”, those instructions are processed by the 16-bit mode; when M9161=“ON”, are by the 8-bit mode.	○	○	○	○
M9165	Assign the SORT2 instruction operating mode. When M9165=“OFF”, the sort is by ascending order; when M9165=“ON”, by descending order.	○	○	○	○
M9167	Assign the HKY instruction operating mode. When M9167=“OFF”, by the “DEC” numeric mode; when M9167=“ON”, by the “HEX” numeric mode.	○	○	○	○
M9168	Assign the SMOV instruction operating mode. When M9168=“OFF”, by the “DEC” numeric mode; when M9168=“ON”, by the “HEX” numeric mode.	○	○	○	○
Step Ladder Instruction Correlated Flags		VS1	VS2	VSM	VS3
M9040	To prevent the step transfer. When M9040=“ON”, the STL state transfer function is disabled.	○	○	○	○
■M9046	STL step is working. When M9047=“ON” and any relay 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 Prohibit Flag		VS1	VS2	VSM	VS3
M9050	To prevent the external interrupt from IX0. The IX0P or IX0F is prohibited.	○	○	○	○
M9051	To prevent the external interrupt from IX1. The IX1P or IX1F is prohibited.	○	○	○	○
M9052	To prevent the external interrupt from IX2. The IX2P or IX2F is prohibited.	○	○	○	○
M9053	To prevent the external interrupt from IX3. The IX3P or IX3F is prohibited.	○	○	○	○
M9054	To prevent the external interrupt from IX4. The IX4P or IX4F is prohibited.	○	○	○	○
M9055	To prevent the external interrupt from IX5. The IX5P or IX5F is prohibited.	○	○	○	○
M9056	To prevent the external interrupt from IX6. The IX6P or IX6F is prohibited.	○	○	○	○
M9057	To prevent the external interrupt from IX7. The IX7P or IX7F is prohibited.	○	○	○	○
M9058	To prevent the timer interrupt ITA. The ITA□□ for timer interrupt is prohibited.	○	○	○	○
M9059	To prevent the timer interrupt ITB. The ITB□□ for timer interrupt is prohibited.	○	○	○	○
M9060	To prevent the timer interrupt ITC. The ITC□□ for timer interrupt is prohibited.	○	○	○	○
M9061	To prevent the SHSCs' interrupt. Software High Speed Counter interrupts IHC0~7 are prohibited.	○	○	○	○
M9062	To prevent the HHSC1's interrupt. Hardware High Speed Counter interrupt IHHC1 is prohibited.	○	○	○	○
M9063	To prevent the HHSC2's interrupt. Hardware High Speed Counter interrupt IHHC2 is prohibited.	○	○	○	○
Error Message		VS1	VS2	VSM	VS3
■M9066	Program CHECK SUM error will cause the PLC stop, M9066=“ON” and “ERR” indicator 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 error occur will be latched in D9068,D9069.	○	○	○	○
Loop Counter		VS1	VS2	VSM	VS3
M9072	Start the 32-bit up-count loop counter 0~2,147,483,647. (unit: ms)	○	○	○	○
M9099	Start the 16-bit high-speed-up-count loop counter 0~32,767. (unit: 0.1ms)	○	○	○	○
Pulse Measurement		VS1	VS2	VSM	VS3
■M9075	Pulse measurement setting-up flag. Use this flag contact to active the pulse width / period measurement function at the X0, X1, X3 or X4.	○	○	○	○
M9076	To start the X0 for pulse measurement.	○	○	○	○
M9077	To start the X1 for pulse measurement.	○	○	○	○
M9078	To start the X3 for pulse measurement.	○	○	○	○
M9079	To start the X4 for pulse measurement.	○	○	○	○
M9080	To set the mode of X0's pulse measurement. “OFF”: pulse width measurement, “ON”: pulse period measurement	○	○	○	○

Relay ID No.	Description	Series			
Pulse Measurement		VS1	VS2	VSM	VS3
M9081	To set the mode of X1's pulse measurement. “OFF”: pulse width measurement, “ON”: pulse period measurement	○	○	○	○
M9082	To set the mode of X3's pulse measurement. “OFF”: pulse width measurement, “ON”: pulse period measurement	○	○	○	○
M9083	To set the mode of X4's pulse measurement. “OFF”: pulse width measurement, “ON”: pulse period measurement	○	○	○	○
CP1 Communication		VS1	VS2	VSM	VS3
M9100	CP1 RS instruction data sending out request flag.	○	○	○	○
M9101	CP1 RS instruction data receive completed flag.	○	○	○	○
M9102	CP1 RS instruction data receive time-out flag.	○	○	○	○
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.	○	○	○	○
■M9104	CP1 LINK / MBUS instruction on execution table complete once flag.	○	○	○	○
CP2 Communication		VS1	VS2	VSM	VS3
M9110	CP2 RS instruction data sending out request flag.	○	○	○	○
M9111	CP2 RS instruction data receive completed flag.	○	○	○	○
M9112	CP2 RS instruction data receive time-out flag.	○	○	○	○
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.	○	○	○	○
■M9114	CP2 LINK / MBUS instruction on execution table complete once flag.	○	○	○	○
CP3 Communication		VS1	VS2	VSM	VS3
M9120	CP3 RS instruction data sending out request flag.	○	○	○	○
M9121	CP3 RS instruction data receive completed flag.	○	○	○	○
M9122	CP3 RS instruction data receive time-out flag.	○	○	○	○
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.	○	○	○	○
■M9124	CP3 LINK / MBUS instruction on execution table complete once flag.	○	○	○	○
CP4 Communication		VS1	VS2	VSM	VS3
M9130	CP4 RS instruction data sending out request flag.				○
M9131	CP4 RS instruction data receive completed flag.				○
M9132	CP4 RS instruction data receive time-out flag.				○
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.				○
■M9134	CP4 LINK / MBUS instruction on execution table complete once flag.				○
CP5 Communication		VS1	VS2	VSM	VS3
M9140	CP5 RS instruction data sending out request flag.				○
M9141	CP5 RS instruction data receive completed flag.				○
M9142	CP5 RS instruction data receive time-out flag.				○
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.				○
■M9144	CP5 LINK / MBUS instruction on execution table complete once flag.				○
Input Pulse Capture Flag		VS1	VS2	VSM	VS3
M9170	X0 input signal captured flag.	○	○	○	○
M9171	X1 input signal captured flag.	○	○	○	○
M9172	X2 input signal captured flag.	○	○	○	○
M9173	X3 input signal captured flag.	○	○	○	○
M9174	X4 input signal captured flag.	○	○	○	○
M9175	X5 input signal captured flag.	○	○	○	○
M9176	X6 input signal captured flag.	○	○	○	○
M9177	X7 input signal captured flag.	○	○	○	○
Hardware High Speed Counter		VS1	VS2	VSM	VS3
■M9196	HHSC1's counting direction flag. When M9196=“OFF”, up counting; when “ON”, down counting.	○	○	○	○
■M9197	HHSC2's counting direction flag. When M9197=“OFF”, up counting; when “ON”, down counting.	○	○	○	○

Relay ID No.	Description	Series			
The 32-bit Counter Count Direction Control		VS1	VS2	VSM	VS3
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 Flag of Software High Speed Counter Count Direction		VS1	VS2	VSM	VS3
M9235 ~ M9245	When M92□□ =“OFF”, the C2□□ is operated as a up counter. When M92□□ =“ON”, the C2□□ is operated as a down counter.	○	○	○	○
Monitoring Flag of Software High Speed Counter Count Direction		VS1	VS2	VSM	VS3
■M9246 ~ ■M9255	When C2□□ is operated a up count, M92□□ =“OFF”. When C2□□ is operated a down count, M92□□ =“ON”.	○	○	○	○
Working Area of the EC1 Expansion Card		VS1	VS2	VSM	VS3
M9260 ~ M9279	EC1 Expansion Card's working area. M9260~M9267= EC1X0~EC1X7; M9270~M9277=EC1Y0~EC1Y7.	○	○	○	○
Working Area of the EC2 Expansion Card		VS1	VS2	VSM	VS3
M9280 ~ M9299	EC2 Expansion Card's working area. M9280~M9287= EC2X0~EC2X7; M9290~M9297=EC2Y0~EC2Y7.	○	○	○	○
Working Area of the EC3 Expansion Card		VS1	VS2	VSM	VS3
M9300 ~ M9319	EC3 Expansion Card's working area. M9300~M9307= EC3X0~EC3X7; M9310~M9317=EC3Y0~EC3Y7.	○	○	○	○
DUTY Instruction Output		VS1	VS2	VSM	VS3
M9330	Destination #1 for the timing sequence pulse generative instruction DUTY (FNC186).				○
M9331	Destination #2 for the timing sequence pulse generative instruction DUTY (FNC186).				○
M9332	Destination #3 for the timing sequence pulse generative instruction DUTY (FNC186).				○
M9333	Destination #4 for the timing sequence pulse generative instruction DUTY (FNC186).				○
M9334	Destination #5 for the timing sequence pulse generative instruction DUTY (FNC186).				○
Y0 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9340	Y0 axis's status. “OFF” means the Y0 is in the READY status, it is available for a positioning instruction; while “ON” = BUSY, the Y0 has been occupying.	○	○	○	○
■M9341	Y0 axis's pulse output monitor. “ON” means pulse is generating.	○	○	○	○
■M9342	Y0 axis's positioning completed flag.	○	○	○	○
■M9343	Y0 axis's positioning abnormal stop flag.	○	○	○	○
■M9344	Y0 axis's zero home positioning has been completed.	○	○	○	○
M9345	Y0 axis's stop flag (with gradually slow down).	○	○	○	○
M9346	Y0 axis's immediately stop flag.	○	○	○	○
M9347	Y0 axis's table positioning start signal.	○	○	○	○
■M9348	Y0 axis's M-code active flag.	○	○	○	○
M9349	Y0 axis's M-code clear command.	○	○	○	○
M9350	Y0 axis's external interrupt trigger type. When M9350=“OFF”, the interrupt is triggered by a rising edge; when M9350=“ON”, that is triggered by a falling edge.	○	○	○	○
Y1 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9360	Y1 axis's status. “OFF” means the Y1 is in the READY status, it is available for a positioning instruction; while “ON” = BUSY, the Y1 has been occupying.	○	○	○	○
■M9361	Y1 axis's pulse output monitor. “ON” means pulse is generating.	○	○	○	○
■M9362	Y1 axis's positioning completed flag.	○	○	○	○
■M9363	Y1 axis's positioning abnormal stop flag.	○	○	○	○
■M9364	Y1 axis's zero home positioning has been completed.	○	○	○	○
M9365	Y1 axis's stop flag (with gradually slow down).	○	○	○	○
M9366	Y1 axis's immediately stop flag.	○	○	○	○
M9367	Y1 axis's table positioning start signal.	○	○	○	○
■M9368	Y1 axis's M-code active flag.	○	○	○	○
M9369	Y1 axis's M-code clear command.	○	○	○	○
M9370	Y1 axis's external interrupt trigger type. When M9370=“OFF”, the interrupt is triggered by a rising edge; when M9370=“ON”, that is triggered by a falling edge.	○	○	○	○

Relay ID No.	Description	Series			
Y2 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9380	Y2 axis's status. "OFF" means the Y2 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y2 has been occupying.	○	○	○	○
■M9381	Y2 axis's pulse output monitor. "ON" means pulse is generating.	○	○	○	○
■M9382	Y2 axis's positioning completed flag.	○	○	○	○
■M9383	Y2 axis's positioning abnormal stop flag.	○	○	○	○
■M9384	Y2 axis's zero home positioning has been completed.	○	○	○	○
M9385	Y2 axis's stop flag (with gradually slow down).	○	○	○	○
M9386	Y2 axis's immediately stop flag.	○	○	○	○
M9387	Y2 axis's table positioning start signal.	○	○	○	○
■M9388	Y2 axis's M-code active flag.	○	○	○	○
M9389	Y2 axis's M-code clear command.	○	○	○	○
M9390	Y2 axis's external interrupt trigger type. When M9390="OFF", the interrupt is triggered by a rising edge; when M9390="ON", that is triggered by a falling edge.	○	○	○	○
Y3 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9400	Y3 axis's status. "OFF" means the Y3 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y3 has been occupying.	○	○	○	○
■M9401	Y3 axis's pulse output monitor. "ON" means pulse is generating.	○	○	○	○
■M9402	Y3 axis's positioning completed flag.	○	○	○	○
■M9403	Y3 axis's positioning abnormal stop flag.	○	○	○	○
■M9404	Y3 axis's zero home positioning has been completed.	○	○	○	○
M9405	Y3 axis's stop flag (with gradually slow down).	○	○	○	○
M9406	Y3 axis's immediately stop flag.	○	○	○	○
M9407	Y3 axis's table positioning start signal.	○	○	○	○
■M9408	Y3 axis's M-code active flag.	○	○	○	○
M9409	Y3 axis's M-code clear command.	○	○	○	○
M9410	Y3 axis's external interrupt trigger type. When M9410="OFF", the interrupt is triggered by a rising edge; when M9410="ON", that is triggered by a falling edge.	○	○	○	○

2-14-2 Instruction Table of Special Register

Register ID No.	Description		Series			
PLC Operation Status			VS1	VS2	VSM	VS3
D9000	Time Setting of Watch Dog Timer. The WDT default value is 200ms (unit: 1ms)		○	○	○	○
■D9004	Error code. When M9004="ON", this content value will show an error report. Please refer to the "2-14-3 Error Code Description" for more detail.		○	○	○	○
D9010	Current operation scan time (unit: 0.1ms)		○	○	○	○
D9011	Min. scan time (unit: 0.1ms)		○	○	○	○
D9012	Max. scan time (unit: 0.1ms)		○	○	○	○
System Status			VS1	VS2	VSM	VS3
■D9001	The PLC's model and version. { VS1 series:10, VS2 series:11, VSM series:12, VS3 series:13, VSM-28ML :14 } 1 0 1 0 0 — Version: V1.00		○	○	○	○
■D9002	Capacity size of memory. "16" = 16K words, "32" = 32K words, "64" = 64K words.		○	○	○	○
■D9003	Type of Memory. 00H indicates using the built-in Flash Memory of PLC. 10H indicates using the extend Flash Memory Card.		○	○	○	○
D9020	Input points (X0~X7) filter response time setting. (unit: 1ms) The default value is 10ms and the available range is 0~60ms.		○	○	○	○
D9039	To set the fixed period of the PLC's Scan Time. The default value is 0ms (unit: 1ms).		○	○	○	○
Time Data of Real Time Clock (RTC)			VS1	VS2	VSM	VS3
D9013	Seconds value. (0~59)		○	○	○	○
D9014	Minute value. (0~59)		○	○	○	○
D9015	Hour value. (0~23)		○	○	○	○
D9016	Date value. (1~31)		○	○	○	○
D9017	Month value. (1~12)		○	○	○	○
D9018	Year value. (2000~20YY, 4 digits)		○	○	○	○
D9019	Day of week. 0 (Sun.) ~ 6 (Sat.)		○	○	○	○
VS-3AV-EC Voltage I/O Brief Expansion Card			VS1	VS2	VSM	VS3
■D9030	The AD converted value of VI1 at the VS-3AV-EC, 0~10V = 0~4000		○	○	○	○
■D9031	The AD converted value of VI2 at the VS-3AV-EC, 0~10V = 0~4000		○	○	○	○
D9032	The DA digital input value for the VO at the VS-3AV-EC, 0~1000 = 0~10V		○	○	○	○
Step Ladder Instruction Correlated			VS1	VS2	VSM	VS3
■D9040	1st active STL step	When M9047="ON", the active STL step ID numbers will be stored in D9040~D9047, where the D9040 will store the lowest active ID number, the second lowest one will store in the 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			VS1	VS2	VSM	VS3
■D9067	Error code to identify the operation error.		○	○	○	○
D9068	Lower 16 bits	Latched the step address number of the operation error.	○	○	○	○
D9069	Upper 16 bits					
D9070	Lower 16 bits	Step address number of the operation error.	○	○	○	○
D9071	Upper 16 bits					
Loop Counter			VS1	VS2	VSM	VS3
D9072	Lower 16 bits	The current value of 32-bit up-count loop counter 0~2,147,483,647. (unit: ms)	○	○	○	○
D9073	Upper 16 bits					
D9099	The current value of 16-bit high-speed-up-count loop counter 0~32,767. (unit: 0.1ms)		○	○	○	○

Register ID No.		Description	Series			
Pulse Measurement			VS1	VS2	VSM	VS3
D9074	Lower 16 bits	The X0's rising edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9075	Upper 16 bits					
D9076	Lower 16 bits	The X0's falling edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9077	Upper 16 bits					
D9078	Lower 16 bits	The X0's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, min. Pulse Period: 20μs	○	○	○	○
D9079	Upper 16 bits					
D9080	Lower 16 bits	The X1's rising edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9081	Upper 16 bits					
D9082	Lower 16 bits	The X1's falling edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9083	Upper 16 bits					
D9084	Lower 16 bits	The X4's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, min. Pulse Period: 20μs	○	○	○	○
D9085	Upper 16 bits					
D9086	Lower 16 bits	The X3's rising edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9087	Upper 16 bits					
D9088	Lower 16 bits	The X3's falling edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9089	Upper 16 bits					
D9090	Lower 16 bits	The X4's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, min. Pulse Period: 20μs	○	○	○	○
D9091	Upper 16 bits					
D9092	Lower 16 bits	The X4's rising edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9093	Upper 16 bits					
D9094	Lower 16 bits	The X4's falling edge to catch the current value of loop counter. (unit: 1/6 μs)	○	○	○	○
D9095	Upper 16 bits					
D9096	Lower 16 bits	The X4's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, min. Pulse Period: 20μs	○	○	○	○
D9097	Upper 16 bits					
CP1 Communication Port			VS1	VS2	VSM	VS3
D9100	The CP1's station number.		○	○	○	○
■D9101	The CP1's amount of residual data to be sent out by the instruction RS.		○	○	○	○
■D9102	The CP1's amount of the data already received by the instruction RS.		○	○	○	○
CP2 Communication Port			VS1	VS2	VSM	VS3
D9110	The CP2's station number.		○	○	○	○
■D9111	The CP2's amount of residual data to be sent out by the instruction RS.		○	○	○	○
■D9112	The CP2's amount of the data already received by the instruction RS.		○	○	○	○
CP3 Communication Port			VS1	VS2	VSM	VS3
D9120	The CP3's station number.		○	○	○	○
■D9121	The CP3's amount of residual data to be sent out by the instruction RS.		○	○	○	○
■D9122	The CP3's amount of the data already received by the instruction RS.		○	○	○	○
CP4 Communication Port			VS1	VS2	VSM	VS3
D9130	The CP4's station number.					○
■D9131	The CP4's amount of residual data to be sent out by the instruction RS.					○
■D9132	The CP4's amount of the data already received by the instruction RS.					○
CP5 Communication Port			VS1	VS2	VSM	VS3
D9140	The CP5's station number.					○
■D9141	The CP5's amount of residual data to be sent out by the instruction RS.					○
■D9142	The CP5's amount of the data already received by the instruction RS.					○

Register ID No.	Description			Series		
RND, HSCT, INT		VS1	VS2	VSM	VS3
D9160	Lower 16 bits: Providing a number for the RND (FNC184) instruction to produce a random value. Initial value: K1	○	○	○	○
D9161	Upper 16 bits				
■D9162	The number of count is activated at the table of HSCT instruction.	○	○	○	○
D9163	External interrupt delay time Set Value (Unit: ms).	○	○	○	○
Index Register V, Z		VS1	VS2	VSM	VS3
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	○	○	○	○
Hardware High Speed Counter		VS1	VS2	VSM	VS3
D9224	HHSC1 counting mode selection. "0" is to disable the HHSC1; "1" ~ "6" represent different modes.	○	○	○	○
D9225	HHSC2 counting mode selection. "0" is to disable the HHSC2; "1" ~ "6" represent different 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 Set Value of HHSC1.	○	○	○	○
D9231	Upper 16 bits				
D9232	Lower 16 bits: The Set Value of HHSC2.	○	○	○	○
D9233	Upper 16 bits				
Working Area of the EC1 Expansion Card		VS1	VS2	VSM	VS3
D9260 ~ D9279	EC1 Expansion Card's working area. D9260~D9279=EC1D0~EC1D19.	○	○	○	○
Working Area of the EC2 Expansion Card		VS1	VS2	VSM	VS3
D9280 ~ D9299	EC2 Expansion Card's working area. D9280~D9299=EC2D0~EC2D19.	○	○	○	○
Working Area of the EC3 Expansion Card		VS1	VS2	VSM	VS3
D9300 ~ D9319	EC2 Expansion Card's working area. D9300~D9319=EC3D0~EC3D19.	○	○	○	○
DUTY Instruction Output		VS1	VS2	VSM	VS3
■D9330	Cycle counter #1 for the timing sequence pulse generative instruction DUTY (FNC186).				○
■D9331	Cycle counter #2 for the timing sequence pulse generative instruction DUTY (FNC186).				○
■D9332	Cycle counter #3 for the timing sequence pulse generative instruction DUTY (FNC186).				○
■D9333	Cycle counter #4 for the timing sequence pulse generative instruction DUTY (FNC186).				○
■D9334	Cycle counter #5 for the timing sequence pulse generative instruction DUTY (FNC186).				○

Register ID No.		Description	Series			
Y0 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9340	Lower 16 bits	The Y0's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz.	○	○	○	○
D9341	Upper 16 bits					
D9342		The Y0's bias speed. Available range: 0~20kHz.	○	○	○	○
D9343		The Y0's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.	○	○	○	○
D9344		The Y0's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.	○	○	○	○
D9345		The Y0's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.	○	○	○	○
D9346	Lower 16 bits	The Y0's preSet Value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9347	Upper 16 bits					
D9348		The Y0's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).	○	○	○	○
■D9349		The Y0's M-code register.	○	○	○	○
■D9350	Lower 16 bits	The Y0's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9351	Upper 16 bits					
D9352	Lower 16 bits	The Y0's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9353	Upper 16 bits					
D9354	Lower 16 bits	The Y0's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9355	Upper 16 bits					
Y1 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9360	Lower 16 bits	The Y1's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz.	○	○	○	○
D9361	Upper 16 bits					
D9362		The Y1's bias speed. Available range: 0~20kHz.	○	○	○	○
D9363		The Y1's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.	○	○	○	○
D9364		The Y1's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.	○	○	○	○
D9365		The Y1's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.	○	○	○	○
D9366	Lower 16 bits	The Y1's preSet Value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9367	Upper 16 bits					
D9368		The Y1's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).	○	○	○	○
■D9369		The Y1's M-code register.	○	○	○	○
■D9370	Lower 16 bits	The Y1's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9371	Upper 16 bits					
D9372	Lower 16 bits	The Y1's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9373	Upper 16 bits					
D9374	Lower 16 bits	The Y1's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9375	Upper 16 bits					

Register ID No.		Description	Series			
Y2 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9380	Lower 16 bits	The Y2's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz	○	○	○	○
D9381	Upper 16 bits					
D9382	The Y2's bias speed. Available range: 0~20kHz.		○	○	○	○
D9383	The Y2's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9384	The Y2's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9385	The Y2's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.		○	○	○	○
D9386	Lower 16 bits	The Y2's preSet Value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9387	Upper 16 bits					
D9388	The Y2's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).		○	○	○	○
■D9389	The Y2's M-code register.		○	○	○	○
■D9390	Lower 16 bits	The Y2's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9391	Upper 16 bits					
D9392	Lower 16 bits	The Y2's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9393	Upper 16 bits					
D9394	Lower 16 bits	The Y2's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9395	Upper 16 bits					
Y3 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9400	Lower 16 bits	The Y3's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz	○	○	○	○
D9401	Upper 16 bits					
D9402	The Y3's bias speed. Available range: 0~20kHz.		○	○	○	○
D9403	The Y3's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9404	The Y3's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9405	The Y3's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.		○	○	○	○
D9406	Lower 16 bits	The Y3's preSet Value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9407	Upper 16 bits					
D9408	The Y3's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).		○	○	○	○
■D9409	The Y3's M-code register.		○	○	○	○
■D9410	Lower 16 bits	The Y3's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9411	Upper 16 bits					
D9412	Lower 16 bits	The Y3's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9413	Upper 16 bits					
D9414	Lower 16 bits	The Y3's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9415	Upper 16 bits					

2-14-3 Error Code Description

System error information (use the contents of D9004)

Error Code	Error Cause	Opportunity to Detect the Error	PLC Status	Status of the ERR Indicator
9064	PLC ID ≠ Project ID	STOP→RUN	STOP	Twinkling by 1Hz
9065	PLC model is incorrect	STOP→RUN	STOP	Twinkling by 1Hz
9066	Check Sum incorrect	STOP→RUN	STOP	Twinkling by 1Hz
9067	Operation error	During the program processing	RUN	OFF
6100	ROM access error	Power OFF→ON	STOP	ON
6101	RAM access error	Power OFF→ON	STOP	ON
6105	Watch Dog Timer exceeding	During the program processing	STOP	Twinkling by 1Hz

Operation error information (use the contents of D9067)

Error Code	Error Cause
0	Normal, no processing error
6701	The destination pointer of the CJ or CALL instruction is not exist. Usually the pointer is modified by an incorrect index register.
6702	More than 5 Levels of the CALL instruction have been nested together.
6704	More than 5 Levels of the FOR / NEXT loop have been nested together.
6705	An incompatible device has been specified as an operand for an application instruction.
6706	Specified devices or contents at the operand has been exceeded the range for the application instruction.
6708	The source or destination for the FROM / TO instruction is not existed.
6710	The assigned parameter does not match with the application instruction.

The code below is for the PID instruction

Error Code	Error Cause	Effect to the Instruction
6730	The setting value of Sampling Time (Ts) is beyond the range ($Ts < 1$).	The PID instruction will stop to operate
6732	The setting value of Input Filter constant (α) is beyond the range ($\alpha < 0$ or $\alpha \geq 100$).	
6733	The setting value of Proportional Gain constant (KP) is beyond the range ($KP < 1$).	
6734	The setting value of Integral Time constant (TI) is beyond the range ($TI < 0$)	
6735	The setting value of Differential Gain constant (KD) is beyond the range ($KD < 0$ or $KD > 100$).	
6736	The setting value of Derivative Time constant (TD) is beyond the range ($TD < 0$).	
6740	The Sampling Time <= The Scan Time of PLC	The PID instruction will continue to operate
6742	The change of the measured Present 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 Proportional Gain (KP) is too large, it cause the calculating value of proportion which exceeds the range.	
6746	The calculating value of Differential process exceeds −32768~32767.	
6747	The calculating result value of the PID instruction which exceeds −32768~32767.	

2-15 The X0~X7 High Speed Input Function Description

The input points X0~X7 of the VS series PLC have the abilities to respond to high-speed input and to execute many high speed functions. The functions of these 8 output points are listed as follows.

Function	External Input Point								Ref. Page
	X0	X1	X2	X3	X4	X5	X6	X7	
Common Input	Use D9020 to adjust the filter time as 0~60ms.								183
Frequency Meter	Use the SPD instruction to perform the speed detection function								190
Software High Speed Counter	Become the input of the C235~C255 1-phase/2-phase/AB phase counter, hereby could make the IHC0~IHC7 interrupt								25
External Interrupt	IX0P/F	IX1P/F	IX2P/F	IX3P/F	IX4P/F	IX5P/F	IX6P/F	IX7P/F	54
Pulse Capture	M9170	M9171	M9172	M9173	M9174	M9175	M9176	M9177	55
Hardware High Speed Counter	HHSC1, to make the IHHC1 interrupt		—	HHSC2, to make the IHHC2 interrupt		—	—	—	58
Pulse Measurement	Width / Period measurement		—	Width / Period measurement		—	—	—	56
Positioning Control	Can be the input points of the positioning control's DOG, PG0, INT signals or for the handwheel								427

- Common Input is available to work with an other advanced function.
- When one of the X0~X7 performs a mentioned advanced function above, this input point is not reusable with another function. However, External Interrupt could cooperate with Pulse Measurement function. (For details, please see the specific function description.)
- When a mentioned advanced function above performs, the filter time of this input point will be automatically adjusted to 0 ms (deviates form D9020). To prevent noise interference, input points X0~X7 are also mounted with hardware RC filters.
  Thus, the filter time of 0 ms is not true 0 ms. In addition, the response time of input points X0~X7 is varied depending on the series of VS1, VS2, VSM or VS3.
  When the filter time is adjusted to 0 ms, the minimum ON or OFF signal width of each input point is listed as follows.

Series	External Input Point							
	X0	X1	X2	X3	X4	X5	X6	X7
VS1	50μs	50μs	50μs	50μs	50μs	50μs	50μs	50μs
VS2	10μs	10μs	10μs	10μs	10μs	10μs	10μs	10μs
VSM	2.5μs	2.5μs	10μs	2.5μs	2.5μs	10μs	10μs	10μs
VSM-28ML	0.5μs	0.5μs	10μs	0.5μs	0.5μs	10μs	10μs	10μs
VS3	2.5μs	2.5μs	10μs	2.5μs	2.5μs	10μs	10μs	10μs

- When a multi-function input point is performing an advanced function, the reaction speed of the point should be very fast. On the other hand, the consequence is relatively sensitive (higher frequency means more sensitive), which makes the input vulnerability to noise interference. Therefore, it's necessary to pay special attention to the external wiring. Keep away from interference sources, or even use isolation cables.
- As the adjustable input filter time, speed detection and the functions of high-speed counters have been described above, the sections below will describe the functions of external interrupt, pulse wave capture, hardware high speed input and pulse measurement.

2-15-1 External Interrupt

The VS series PLC has 8 external interrupt input points X0~X7. The External Interrupt function can be employed when the external input signal that needs instant response, not to be affected by scan time or to read a narrow width pulse.

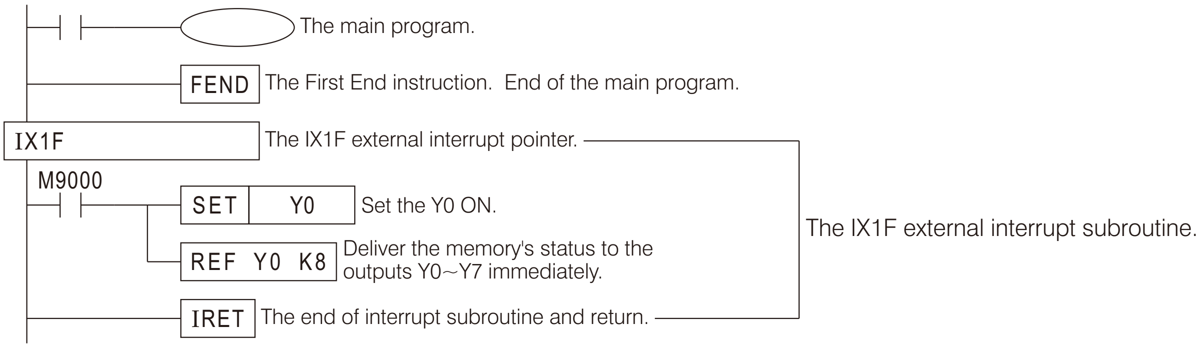
The External Interrupt signal works with its corresponding interrupt subroutine to execute an external interrupt.

The External Interrupt pointers are listed in the following table:

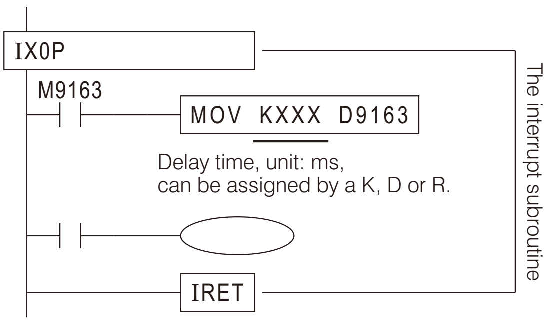
External Input Point	Rising Edge Interrupt Pointer	Falling Edge Interrupt Pointer	Interrupt Prohibit Flag
X0	IX0P	IX0F	M9050
X1	IX1P	IX1F	M9051
X2	IX2P	IX2F	M9052
X3	IX3P	IX3F	M9053
X4	IX4P	IX4F	M9054
X5	IX5P	IX5F	M9055
X6	IX6P	IX6F	M9056
X7	IX7P	IX7F	M9057

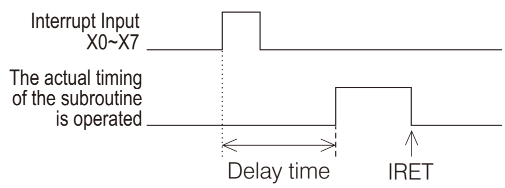
※ One external input point only corresponds to one external interrupt pointer. Such as either the IX0P or IX0F can be used in a program; either the IX1P or IX1F can be used in a program, and so forth.

- Program example:
  When the external input point X1 turns OFF, the IX1F interrupt subroutine is executed and Y0 is set to ON. And the status of Y0 ON is immediately sent to the output port via the I / O refresh command REF (FNC 50).



- The VS series PLC external interrupt has the function of delay interrupt. That delay time is by the unit of 1ms. This feature allows the user to change the starting of the interrupt subroutine by the parameter adjustments, without to change the external detector's location where the interrupt signal has occurred.
  This interrupt delay time setting is by a series of particular pattern below the interrupt pointer. The pattern format of this standard program can not be changed.





The tables below list the special relay and register related to this function:
■ Represents that component is read only.

Relay ID No.	Description
■M9163	External Interrupt delay time set-up flag. Use this flag contact to active the interrupt delay function.

Register ID No.	Description
D9163	External interrupt delay time Set Value (Unit: ms).

2-15-2 Pulse Capture

The function of pulse capture is to get the input signal which the width of ON is narrow. If an input point X0~X7 is not to use an input special function, its pulse capture is active automatically.

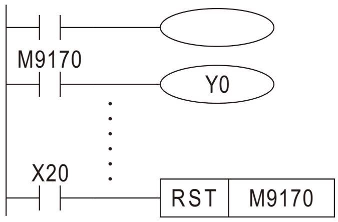
The pulse signals which X0~X7 input points capture will reflect to the special relays M9170~M9177. For each captured point, its special relay can only be cleared by using the RST instruction. And after the END instruction is executed, the input point is able to capture the next signal.

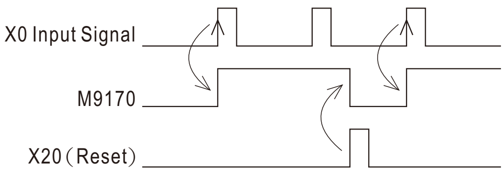
Only one signal can be captured at a scan time. Therefore, if there is a demand for to capture more signals at a short time, the external interrupt function can be used for this purpose.

The table below lists the special relays related to this function:

Relay ID No.	Description
M9170	X0 input signal captured flag.
M9171	X1 input signal captured flag.
M9172	X2 input signal captured flag.
M9173	X3 input signal captured flag.
M9174	X4 input signal captured flag.
M9175	X5 input signal captured flag.
M9176	X6 input signal captured flag.
M9177	X7 input signal captured flag.

- Program example:
  During program operation, when the external input point X0 changes from OFF to ON, the M9170 is set to ON in an interrupt manner. Subsequently, the M9170 will remain its ON status no matter the X0 changes OFF or ON.
  After the X20 is turned ON to reset the M9170 to OFF, the M9170 is able to turned ON again by the next time that X0 changes from OFF to ON.





2-15-3 Pulse Measurement

With the pulse measurement function, the VS series PLC can measure X0, X1, X3, X4 input pulse signal's ON width or cycle period.
The pulse measurement function uses an 1/6 μs loop counter to store the count value to the specific special registers at the rising edge and falling edge of the input signal respectively. And then calculate the pulse width or pulse period of the input signal according to the contents of the registers. The calculated width or cycle is stored in the corresponding special registers in the unit of 10 μs.

The tables below list the special relays and registers related to this function:
■ Represents that component is read only.

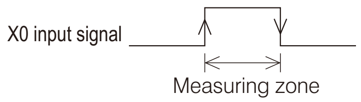
Relay ID No.	Description
■M9075	Pulse measurement setting-up flag. Use this flag contact to active the pulse width / period measurement function at the X0, X1, X3 or X4.
M9076	To start the X0 for pulse measurement.
M9077	To start the X1 for pulse measurement.
M9078	To start the X3 for pulse measurement.
M9079	To start the X4 for pulse measurement.
M9080	To set the mode of X0's pulse measurement. "OFF": pulse width measurement, "ON": pulse period measurement
M9081	To set the mode of X1's pulse measurement. "OFF": pulse width measurement, "ON": pulse period measurement
M9082	To set the mode of X3's pulse measurement. "OFF": pulse width measurement, "ON": pulse period measurement
M9083	To set the mode of X4's pulse measurement. "OFF": pulse width measurement, "ON": pulse period measurement

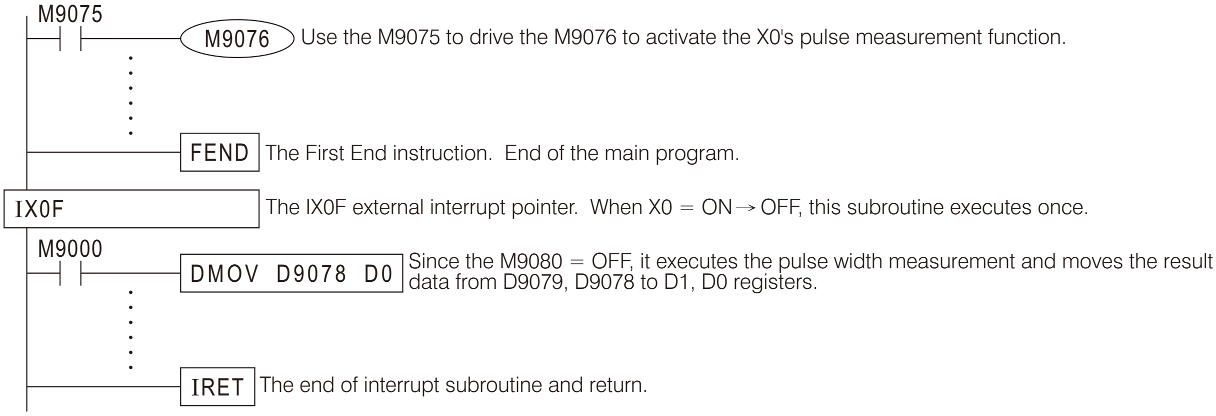
Register ID No.	Description	
D9074	Lower 16 bits	The X0's rising edge to catch the current value of loop counter. (unit: 1/6μs)
D9075	Upper 16 bits	
D9076	Lower 16 bits	The X0's falling edge to catch the current value of loop counter. (unit: 1/6μs)
D9077	Upper 16 bits	
D9078	Lower 16 bits	The X0's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, minimum Pulse Period: 20μs
D9079	Upper 16 bits	
D9080	Lower 16 bits	The X1's rising edge to catch the current value of loop counter. (unit: 1/6μs)
D9081	Upper 16 bits	
D9082	Lower 16 bits	The X1's falling edge to catch the current value of loop counter. (unit: 1/6μs)
D9083	Upper 16 bits	
D9084	Lower 16 bits	The X1's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, minimum Pulse Period: 20μs
D9085	Upper 16 bits	
D9086	Lower 16 bits	The X3's rising edge to catch the current value of loop counter. (unit: 1/6μs)
D9087	Upper 16 bits	
D9088	Lower 16 bits	The X3's falling edge to catch the current value of loop counter. (unit: 1/6μs)
D9089	Upper 16 bits	
D9090	Lower 16 bits	The X3's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, minimum Pulse Period: 20μs
D9091	Upper 16 bits	
D9092	Lower 16 bits	The X4's rising edge to catch the current value of loop counter. (unit: 1/6μs)
D9093	Upper 16 bits	
D9094	Lower 16 bits	The X4's falling edge to catch the current value of loop counter. (unit: 1/6μs)
D9095	Upper 16 bits	
D9096	Lower 16 bits	The X4's Pulse Width/Period Measurement cached value. (unit: 10μs). The measurable range of Width: 10μs~100s, minimum Pulse Period: 20μs
D9097	Upper 16 bits	

- Must use the M9075 to drive M9076~M9079 to activate the pulse measurement function at the corresponding input point X0, X1, X3 or X4.
- The pulse measurement and external interrupt functions can use an external input point at the same time.

● Program example 1:

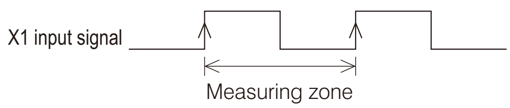
Pulse width measurement at the X0 input point.

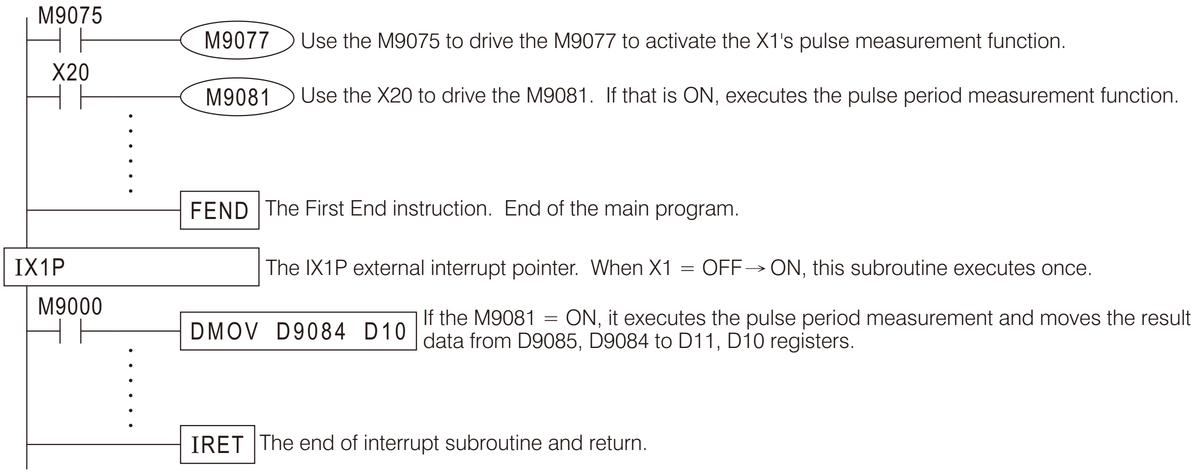


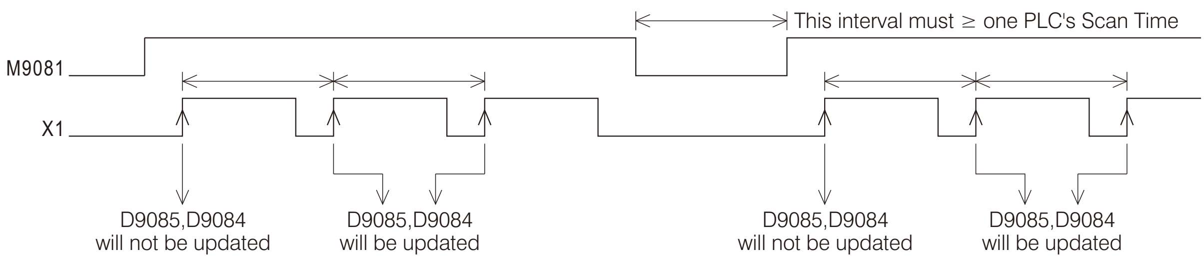


● Program example 2:

Pulse period measurement at the X1 input point.







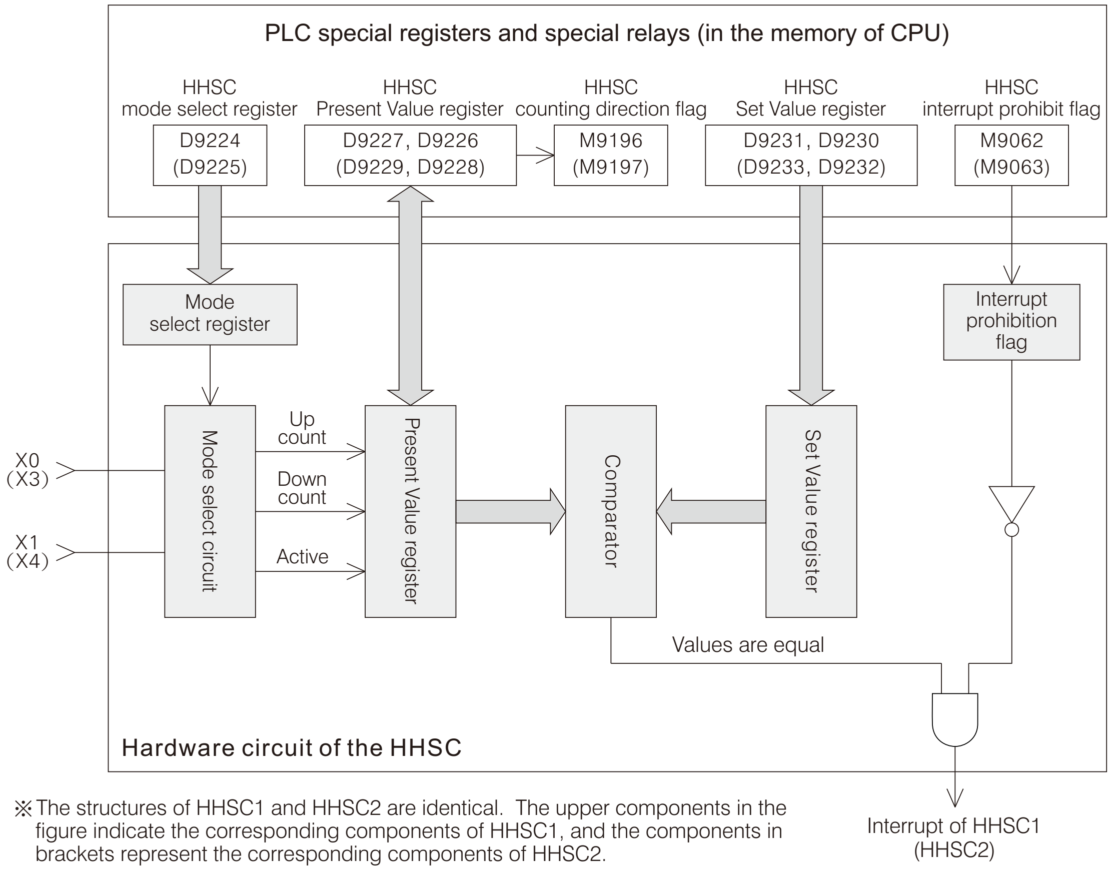
As shown in the figure above, after the period measurement mode (M9081 = OFF→ON) is selected, the measured value of the period in D9085 and D9084 will not be updated at the first rising edge until the next rising edge is appeared.

2-15-4 Hardware High Speed Counter

The VS series PLC has two sets of Hardware High-Speed Counter: HHSC1 and HHSC2.
To reach the purposes of high-speed counting, the HHSC uses its hardware circuit to get high-speed pulse input. Therefore, in the counting process, HHSC will not affect the efficiency of CPU implementation. When planning a control system, one can make good use of HHSC function.

The HHSC is a 32-bit up/down count counter with a latched function. Also, with the function of Set Value comparison, when a pulse input to make its Present Value and Set Value are equal, it starts the IHHC interrupt.

The figure below shows the configuration of HHSC:



※The structures of HHSC1 and HHSC2 are identical. The upper components in the figure indicate the corresponding components of HHSC1, and the components in brackets represent the corresponding components of HHSC2.

- From the figure above, HHSC possesses both CPU's memory registers and hardware circuit registers. When executing the END instruction, the PLC system program automatically writes the HHSC's mode register, set value register and interrupt prohibit flag to the hardware circuit. That also reads the HHSC's Present Value from the hardware circuit and stores it in the CPU's Present Value register.

  In order to meet the needs of fast and prompt process, the VS PLCs are also designed with the “Hardware high-speed counter data move, HHCMV (FNC 189)” instruction. With the HHCMV instruction, the Present Value in the HHSC hardware circuit can be read out immediately or the Set Value can be written to the HHSC hardware circuit.

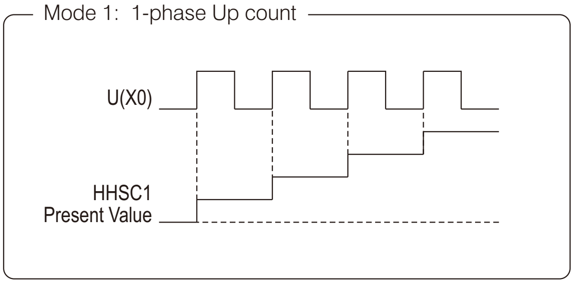
- Table of HHSC Working Modes:

HHSC No.	Input Point	HHSC Operating Mode					
		1-Phase		2-Phase	AB-Phase ×1	AB-Phase × 2	AB-Phase × 4
		1	2	3	4	5	6
HHSC1	X0	U	U/D	U	A	A	A
	X1		DIR	D	B	B	B
HHSC2	X3	U	U/D	U	A	A	A
	X4		DIR	D	B	B	B

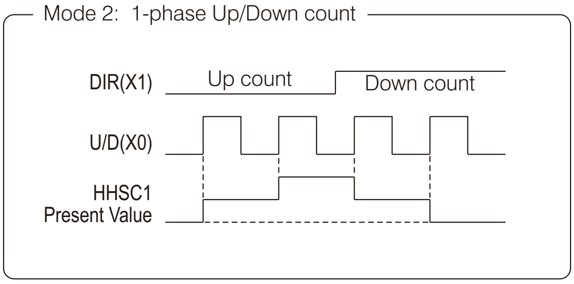
U: Up count input D: Down count input A: A-phase input B: B-phase input
U/D: Up/Down count pulse input DIR: Up/Down directional selector input

- The operating modes of HHSC are illustrated by using the HHSC1.

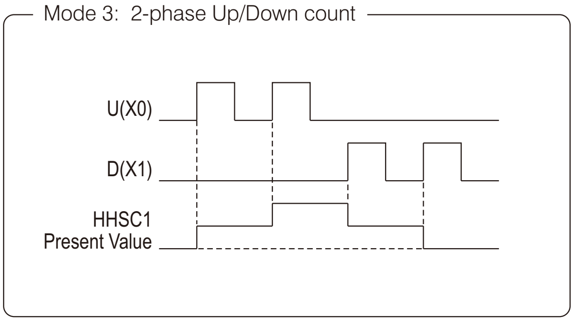
Mode 1: 1-phase Up count



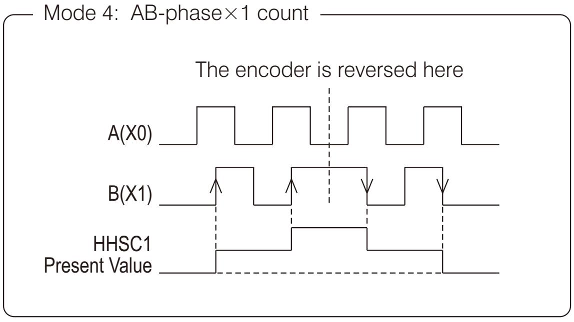
Mode 2: 1-phase Up/Down count



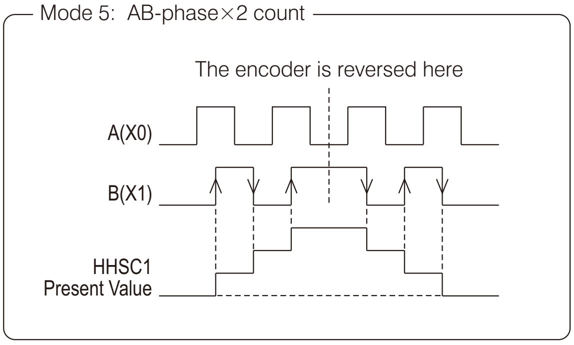
Mode 3: 2-phase Up/Down count



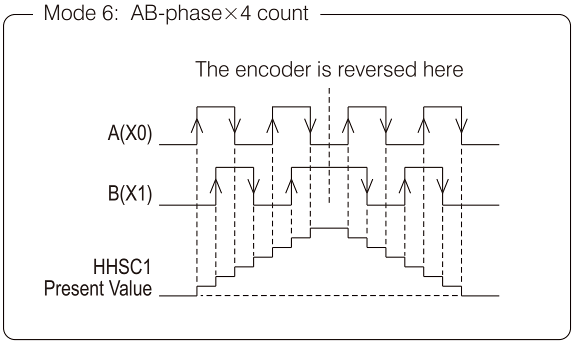
Mode 4: AB-phase×1 count



Mode 5: AB-phase×2 count



Mode 6: AB-phase×4 count



- The tables below list the special relays and registers related to the HHSCs:
  ■ Represents that component is read only.

Relay ID No.	Description
M9062	To prevent the HHSC1's interrupt. Hardware High Speed Counter interrupt IHHC1 is prohibited.
M9063	To prevent the HHSC2's interrupt. Hardware High Speed Counter interrupt IHHC2 is prohibited.
■M9196	HHSC1's counting direction flag. When M9196="OFF", up counting; when "ON", down counting.
■M9197	HHSC2's counting direction flag. When M9197="OFF", up counting; when "ON", down counting.

Register ID No.	Description	
D9224	HHSC1 counting mode selection. "0" is to disable the HHSC1; "1" ~ "6" represent different modes.	
D9225	HHSC2 counting mode selection. "0" is to disable the HHSC2; "1" ~ "6" represent different 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 Set Value of HHSC1.
D9231	Upper 16 bits	
D9232	Lower 16 bits	The Set Value of HHSC2.
D9233	Upper 16 bits	

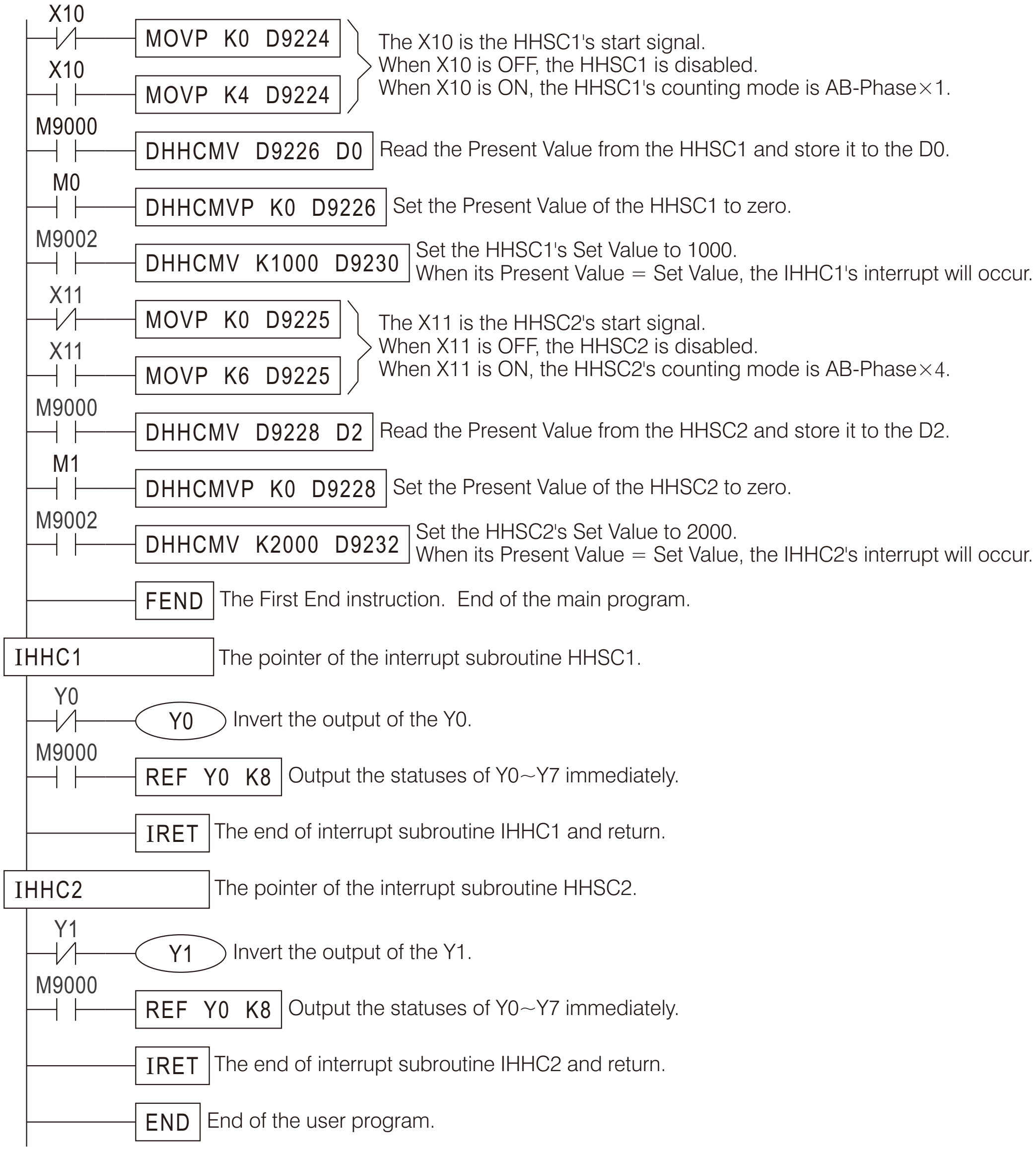
- Program example:

This exemplary program is mainly used to describe the actual usage of the HHSC1 and HHSC2.

To use the HHSC, only need to set the counting mode at the special register, then the HHSC can start to count obediently. When the PLC is executing the END instruction, the PLC system automatically reads the HHSC's Present Value from the hardware circuit, and store it in the Present Value register.

To get the most precise Present Value during the user program is in executing, can use the DHHCMV instruction to read that from the hardware circuit immediately. When the DHHCMV instruction is executing, the PLC system firstly reads out the HHSC hardware circuit's Present Value and stores it in the CPU's Present Value register, then transports this value to the destination register of the DHHCMV instruction. Besides, to reset the Present Value of the HHSC instantly, must use the DHHCMV instruction rather than use the RST instruction.

In addition, according to the quick response of the application requirement, can use its hardware comparison function to immediately generate an interrupt, that could avoid the delay from the PLC's Scan Time.



2-16 Expansion Card Related Components

The Expansion Card Sockets are designed for flexible expansions, on the upper side of the VS series PLC. Which are available to install DIO expansion cards to increase a small number of control points in a cost effective way. Also can install the communication port (CP) expansion card to expand communication capabilities for linking with external accessories of communication control. In addition, the special function (SF) expansion card is capable to perform various special controls, such as position inspection, speed control, temperature control, etc. to present a complicated, high-level control system.

The list of Expansion Cards for the VS series Main Unit:

Item	Model Name	Main Specification
DIO Expansion Card	VS-4XY★-EC	DIO Expansion Card: 2 DI (DC 24V); 2 DO ★; I/O by screw-clamp terminal
	VS-4X-EC	DI Expansion Card: 4 DI (DC 24V); output by screw-clamp terminal
	VS-4Y★-EC	DO Expansion Card: 4 DO ★; output by screw-clamp terminal
	VS-8XY★-EC	DIO Expansion Card: 4 DI (DC 24V); 4 DO ★; I/O by screw-clamp terminal
	VS-8X-EC	DI Expansion Card: 8 DI (DC 24V); input by screw-clamp terminal
	VS-8YT-EC	DO Expansion Card: 8 DO (DC 24V, 300mA NPN transistor); output by screw-clamp terminal
	VS-8XI-EC	DI Expansion Card: 8 DI (DC 24V); input by IDC connector
	VS-8YTI-EC	DO Expansion Card: 8 DO (DC 24V, 100mA NPN transistor); output by IDC connector
Comm. Expansion Card	VS-485-EC	RS-485 Comm. Expansion Card: One non-isolated RS-485 port with TX / RX indicators; dist. 50m Max.
	VS-485A-EC	RS-485 Comm. Expansion Card: One isolated RS-485 port with TX / RX indicators; dist. 1000m Max.
	VS-D485-EC	RS-485 Comm. Expansion Card: Dual non-isolated RS-485 ports with TX / RX indicators; dist. 50m Max.
	VS-D485A-EC	RS-485 Comm. Expansion Card: Dual isolated RS-485 ports with TX / RX indicators; dist. 1000m Max.
	VS-D232-EC	RS-232C Communication Expansion Card: Dual non-isolated RS-232 ports with TX / RX indicators; dist. 15m Max.; wiring by the RX / TX / SG terminals
	VS-D52A-EC	RS-485 + RS-232C Communication Expansion Card: One isolated RS-485 port (1000m) & one non-isolated RS-232C port (15m), both with TX / RX indicators and wiring by terminals
	VS-ENET-EC	Ethernet + RS-485 Communication Expansion Card: One Ethernet port (with additional non-isolated RS-485, dist. 50m) & one non-isolated RS-485 port (dist. 50m), both with TX / RX indicators
Special Function Card	VS-3AV-EC	Brief Voltage I/O Card: 2 channel (0~10V, 12-bit) inputs; 1 channel (0~10V, 10-bit) output; with a calibrated DC 10V output; non-isolated
	VS-4AD-EC	Analog Input Card: 4 channel (12-bit) inputs, each channel could output either 0~10V, 4~20mA or 0~20mA; non-isolated
	VS-2DA-EC	Analog Output Card: 2 channel (12-bit) outputs, each channel could input either 0~10V, 4~20mA or 0~20mA; non-isolated
	VS-4A-EC	Analog I/O Card: 2 channel (12-bit) inputs + 2 channel (12-bit) outputs, each channel could input/output either 0~10V, 4~20mA or 0~20mA; non-isolated
	VS-3ISC-EC	Inverter Speed Control Card: 3 channel (0.1% resolution) voltage outputs; totally isolated for each channel
	VS-2TC-EC	Thermocouple Temperature Input Card: 2 channel (K, J, R, S, T, E, B or N type thermocouple, 0.2~0.3℃ resolution) inputs; non-isolated
	VS-4TC-EC	Thermocouple Temperature Input Card: 4 channel (K, J, R, S, T, E, B or N type thermocouple, 0.2~0.3℃ resolution) inputs; non-isolated
	VS-1PT-EC	PT-100 Temperature Input Card: 1 channel (3-wire PT-100, 0.1℃ resolution) input; non-isolated
	VS-2PT-EC	PT-100 Temperature Input Card: 2 channel (3-wire PT-100, 0.1℃ resolution) inputs; non-isolated

★ Selectable output: R: 2A Relay; T: 0.3A NPN transistor

The VS series PLCs are equipped with one to three Expansion Card Sockets depending on the model. There are configured with 20 special relays and 20 special registers for each Expansion Card Socket as the work area of the card. For helping memory and convenience in use, every special component of the Expansion Card at a socket is given a corresponding “Simple Code”. Hereafter, the “Simple Code” will be used in the following documents.

Expansion Card Socket	Working Area		Available Card Type for the Socket		
	Special Component	Simple Code	DIO Card	CP Card	SF Card *
EC1	M9260～M9269	EC1X0～EC1X7	○	○	○
	M9270～M9279	EC1Y0～EC1Y7			
	D9260 ～ D9279	EC1D0～EC1D19			
EC2	M9280～M9289	EC2X0～EC2X7	○	—	○
	M9290～M9299	EC2Y0～EC2Y7			
	D9280 ～ D9299	EC2D0～EC2D19			
EC3	M9300～M9309	EC3X0～EC3X7	○	○ (at the VS3 only)	○
	M9310～M9319	EC3Y0～EC3Y7			
	D9300 ～ D9319	EC3D0～EC3D19			

＊The VS-3AV-EC can ONLY be installed at the EC2 expansion socket otherwise it is ineffective. Except for the VS-3AV-EC, the VS1 series can install one SF card only.

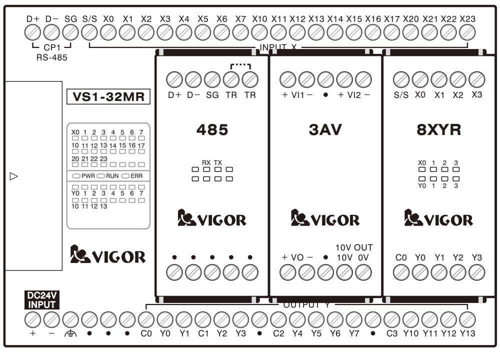
2-16-1 The DIO Expansion Card Related Components

When a DIO Expansion Card is installed in the Main Unit, those components' X/Y numbers at this card are correspond to their Simple Codes respectively. (★ Selectable output: R: Relay; T: NPN transistor)

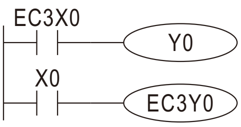
Model Name of Expansion Card	Expansion Card Socket	Simple Code at the DIO Card							
		X0	X1	X2	X3	Y0	Y1	Y2	Y3
VS-4XY★-EC	EC1	EC1X0	EC1X1	—	—	EC1Y0	EC1Y1	—	—
	EC2	EC2X0	EC2X1	—	—	EC2Y0	EC2Y1	—	—
	EC3	EC3X0	EC3X1	—	—	EC3Y0	EC3Y1	—	—
VS-4X-EC	EC1	EC1X0	EC1X1	EC1X2	EC1X3	—	—	—	—
	EC2	EC2X0	EC2X1	EC2X2	EC2X3	—	—	—	—
	EC3	EC3X0	EC3X1	EC3X2	EC3X3	—	—	—	—
VS-4Y★-EC	EC1	—	—	—	—	EC1Y0	EC1Y1	EC1Y2	EC1Y3
	EC2	—	—	—	—	EC2Y0	EC2Y1	EC2Y2	EC2Y3
	EC3	—	—	—	—	EC3Y0	EC3Y1	EC3Y2	EC3Y3
VS-8XY★-EC	EC1	EC1X0	EC1X1	EC1X2	EC1X3	EC1Y0	EC1Y1	EC1Y2	EC1Y3
	EC2	EC2X0	EC2X1	EC2X2	EC2X3	EC2Y0	EC2Y1	EC2Y2	EC2Y3
	EC3	EC3X0	EC3X1	EC3X2	EC3X3	EC3Y0	EC3Y1	EC3Y2	EC3Y3

Model Name of Expansion Card	Expansion Card Socket	Simple Code at the DIO Card							
		X0	X1	X2	X3	X4	X5	X6	X7
VS-8X-EC VS-8XI-EC	EC1	EC1X0	EC1X1	EC1X2	EC1X3	EC1X4	EC1X5	EC1X6	EC1X7
	EC2	EC2X0	EC2X1	EC2X2	EC2X3	EC2X4	EC2X5	EC2X6	EC2X7
	EC3	EC3X0	EC3X1	EC3X2	EC3X3	EC3X4	EC3X5	EC3X6	EC3X7

Model Name of Expansion Card	Expansion Card Socket	Simple Code at the DIO Card							
		Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
VS-8YT-EC VS-8YTI-EC	EC1	EC1Y0	EC1Y1	EC1Y2	EC1Y3	EC1Y4	EC1Y5	EC1Y6	EC1Y7
	EC2	EC2Y0	EC2Y1	EC2Y2	EC2Y3	EC2Y4	EC2Y5	EC2Y6	EC2Y7
	EC3	EC3Y0	EC3Y1	EC3Y2	EC3Y3	EC3Y4	EC3Y5	EC3Y6	EC3Y7



As shown on the left, a VS-8XYR-EC is installed in the EC3 Expansion Card Socket.
Those X0~X3 on the VS-8XYR-EC are correspond to EC3X0~EC3X3 and Y0~Y3 are correspond to EC3Y0~EC3Y3.



Send the VS-8XYR-EC's X0 input status out via the Y0 output point in the Main Unit.

Send the Main Unit's X0 input status out via the Y0 output point in the VS-8XYR-EC.

2-16-2 The Communication Expansion Card Related Components

At the VS1, VS2 or VSM series PLC, it has one built-in communication port CP1 already, moreover at its EC1 socket is available to install one CP card. By to install a single port CP card can have the expanded CP2, or by a dual port CP card to get the CP2 and CP3. As a result, the VS1, VS2 or VSM series PLC could have three communication ports CP1~CP3.

At the VS3 series PLC, its EC1 can install a dual port card to provide the CP2 and CP3, furthermore the EC3 can install a dual port card to provide the CP4 and CP5. Thus the VS3 could have 5 communication ports CP1~CP5. However, if the CP5 at a VS3 PLC is required, its EC2 socket may install a VS-3AV-EC card or not to use; any DIO or SF Card at EC2 will cause the CP5 at EC3 ineffective.

A communication port at CP card is not related to the working area of DIO expansion or SF card, to operate the port is directly by the installed setting and program.

- The special relays and registers related to the CP1: (■ means read only)

Component ID No.	Description
M9100	CP1 RS instruction data sending out request flag.
M9101	CP1 RS instruction data receive completed flag.
M9102	CP1 RS instruction data receive time-out flag.
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
■M9104	CP1 LINK / MBUS instruction on execution table complete once flag.
D9100	The CP1's station number.
■D9101	The CP1's amount of residual data to be sent out by the instruction RS.
■D9102	The CP1's amount of the data already received by the instruction RS.

- The special relays and registers related to the CP2: (■ means read only)

Component ID No.	Description
M9110	CP2 RS instruction data sending out request flag.
M9111	CP2 RS instruction data receive completed flag.
M9112	CP2 RS instruction data receive time-out flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
■M9114	CP2 LINK / MBUS instruction on execution table complete once flag.
D9110	The CP2's station number.
■D9111	The CP2's amount of residual data to be sent out by the instruction RS.
■D9112	The CP2's amount of the data already received by the instruction RS.

- The special relays and registers related to the CP3: (■ means read only)

Component ID No.	Description
M9120	CP3 RS instruction data sending out request flag.
M9121	CP3 RS instruction data receive completed flag.
M9122	CP3 RS instruction data receive time-out flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
■M9124	CP3 LINK / MBUS instruction on execution table complete once flag.
D9120	The CP3's station number.
■D9121	The CP3's amount of residual data to be sent out by the instruction RS.
■D9122	The CP3's amount of the data already received by the instruction RS.

- The special relays and registers related to the CP4: (■ means read only)

Component ID No.	Description
M9130	CP4 RS instruction data sending out request flag.
M9131	CP4 RS instruction data receive completed flag.
M9132	CP4 RS instruction data receive time-out flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
■M9134	CP4 LINK / MBUS instruction on execution table complete once flag.
D9130	The CP4's station number.
■D9131	The CP4's amount of residual data to be sent out by the instruction RS.
■D9132	The CP4's amount of the data already received by the instruction RS.

● The special relays and registers related to the CP5: (■ means read only)

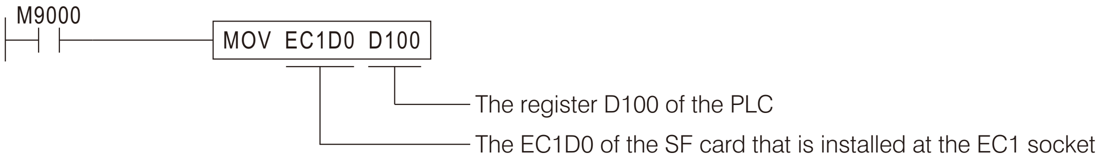
Component ID No.	Description
M9140	CP5 RS instruction data sending out request flag.
M9141	CP5 RS instruction data receive completed flag.
M9142	CP5 RS instruction data receive time-out flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.
■M9144	CP5 LINK / MBUS instruction on execution table complete once flag.
D9140	The CP5's station number.
■D9141	The CP5's amount of residual data to be sent out by the instruction RS.
■D9142	The CP5's amount of the data already received by the instruction RS.

2-16-3 The Special Function Expansion Card Related Components

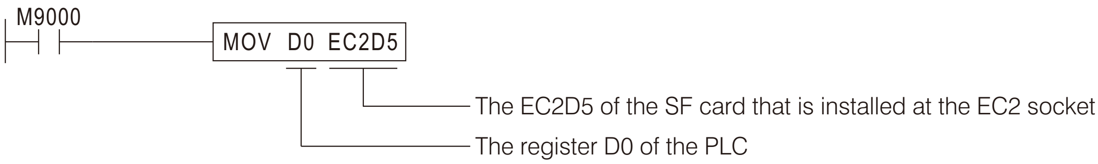
For convenience, every EC1~EC3 expansion card socket at a VS series PLC will possess 20 special registers that is the working area of the installed expansion card. When a special card is installed in the socket, the PLC can access related data for the respective device on the card through its working area.
For easy memorize and convenient application, every special register of each Special Expansion Card is given a "Simple Code". The "Simple Code" will be used in the following documents.

Expansion Card Socket	Expansion Card Working Area	
	Simple Code	Special Register
EC1	EC1D0～EC1D19	D9260 ～ D9279
EC2	EC2D0～EC2D19	D9280 ～ D9299
EC3	EC3D0～EC3D19	D9300 ～ D9319

The instruction diagram below will move the content value from the EC1D0 of special card to the register D100 of PLC. (that card is installed at the EC1 socket)



The instruction diagram below will move the content value from the register D0 of PLC to the EC2D5 of special card. (that card is installed at the EC2 socket)



- Special Register related to the VS-3AV-EC (For this brief card the Simple Code EC2Dn is useless)

  The VS-3AV-EC can ONLY be installed at the EC2 expansion socket otherwise it is ineffective. Also, to access this expansion card is through 3 specific special registers below that instead of expansion card working area.
  ■Represents that component is read only.

Register ID No.	Description
■D9030	The AD converted value of VI1 at the VS-3AV-EC, 0~10V = 0~4000
■D9031	The AD converted value of VI2 at the VS-3AV-EC, 0~10V = 0~4000
D9032	The DA digital input value for the VO at the VS-3AV-EC, 0~1000 = 0~10V

- EC Card Register (Simple Code) related to the VS-4AD-EC

EC1	EC2	EC3	Component Description
EC1D0	EC2D0	EC3D0	To assign the analog input modes of AI1~AI4.
EC1D1	EC2D1	EC3D1	Converted digital value of AI1, 0~4000 or 0~3200.
EC1D2	EC2D2	EC3D2	Converted digital value of AI2, 0~4000 or 0~3200.
EC1D3	EC2D3	EC3D3	Converted digital value of AI3, 0~4000 or 0~3200.
EC1D4	EC2D4	EC3D4	Converted digital value of AI4, 0~4000 or 0~3200.
EC1D18	EC2D18	EC3D18	Identification code: K101 (If code = K240, means connecting error between Main Unit and card)
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □.□)

To appoint the modes of analog inputs: (the sliding switch should also consistent with the modes)

b15 … b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
AI4	AI3	AI2	AI1

To assign input modes:

If the nibble = 0, the channel is assigned for (0~10V) voltage input.
If the nibble = 1, the channel is assigned for (4~20mA) current input.
If the nibble = 2, the channel is assigned for (0~20mA) current input.
If the nibble is any number other than 0, 1 or 2, the channel is disabled.

Example: If VS- 4AD- EC is installed in EC1, and EC1D0 is set to be H3210, then
AI1: voltage input (0~10V) AI2: current input (4~20mA) AI3: current input (0~20mA) AI4: disabled

- EC Card Register (Simple Code) Related to the VS-2DA-EC

EC1	EC2	EC3	Component Description
EC1D10	EC2D10	EC3D10	To assign the analog output modes of AO1~AO2.
EC1D11	EC2D11	EC3D11	Digital Set Value for AO1, 0~4000 or 0~3200.
EC1D12	EC2D12	EC3D12	Digital Set Value for AO2, 0~4000 or 0~3200.
EC1D18	EC2D18	EC3D18	Identification code: K102 (If code = K240, means connecting error between Main Unit and card)
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □.□)

To appoint the modes of analog outputs:

b15 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AO2	AO1

$\underbrace{\text{AO2 AO1}}_{\text{To assign output modes:}}$

If the nibble = 0, the channel is assigned for (0~10V) voltage output.
If the nibble = 1, the channel is assigned for (4~20mA) current output.
If the nibble = 2, the channel is assigned for (0~20mA) current output.
If the nibble is any number other than 0, 1 or 2, the channel is disabled.

Example: If a VS-2DA-EC is installed at the EC1, and its EC1D10 is set to be H10, then
AO1: voltage output (0~10V) AO2: current output (4~20mA)

- EC Card Register (Simple Code) related to the VS-4A-EC

EC1	EC2	EC3	Component Description
EC1D0	EC2D0	EC3D0	To assign the input modes of AI1~AI2.
EC1D1	EC2D1	EC3D1	Read value of AI1, 0~4000 or 0~3200.
EC1D2	EC2D2	EC3D2	Read value of AI2, 0~4000 or 0~3200.
EC1D10	EC2D10	EC3D10	To assign the output modes of AO1~AO2.
EC1D11	EC2D11	EC3D11	Write value of AO1, 0~4000 or 0~3200.
EC1D12	EC2D12	EC3D12	Write value of AO2, 0~4000 or 0~3200.
EC1D18	EC2D18	EC3D18	Identification code: K103 (If code = K240, means connecting error between Main Unit and card)
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □.□)

To appoint the modes of analog inputs: (the sliding switch should also consistent with the modes)

b15 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AI2	AI1

To assign input modes:

If the nibble = 0, the channel is assigned for (0~10V) voltage input.
If the nibble = 1, the channel is assigned for (4~20mA) current input.
If the nibble = 2, the channel is assigned for (0~20mA) current input.
If the nibble is any number other than 0, 1 or 2, the channel is disabled.

Example: If a VS-4A-EC is installed at the EC1, and its EC1D0 is set to be H10, then
AI1: voltage input (0~10V) AI2: current input (4~20mA)

To appoint the modes of analog inputs:

b15 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AO2	AO1

To assign output modes:

If the nibble = 0, the channel is assigned for (0~10V) voltage output.
If the nibble = 1, the channel is assigned for (4~20mA) current output.
If the nibble = 2, the channel is assigned for (0~20mA) current output.
If the nibble is any number other than 0, 1 or 2, the channel is disabled.

Example: If VS-4A-EC is installed in EC1, and EC1D10 is set to be H10, then
AO1: voltage output (0~10V) AO2: current output (4~20mA)

● EC Card Register (Simple Code) Related to the VS-3ISC-EC

EC1	EC2	EC3	Component Description	
EC1D0	EC2D0	EC3D0	VO Set Value of CH1, 0~1000	The output ratio (0% ~ 100.0%) at the VO terminal that connect to the analog speed control point of inverter. This VO Set Value is the percentage of 0 to "VO Max." If the Set Value <0, the output ratio = 0 (0 %). If the Set Value >1000, the output ratio = 1000 (100.0 %).
EC1D1	EC2D1	EC3D1	VO Set Value of CH2 ,0~1000	
EC1D2	EC2D2	EC3D2	VO Set Value of CH3 , 0~1000	
EC1D3	EC2D3	EC3D3	V+ measured voltage value of CH1.	Measure the external control use power from inverter by a voltage meter and fill in the result value here. If the result is 10V, then fill in 1000 (by unit of 0.01V). If the filled result is not between 400 and 1200, then the VO point will output 0V.
EC1D4	EC2D4	EC3D4	V+ measured voltage value of CH2.	
EC1D5	EC2D5	EC3D5	V+ measured voltage value of CH3.	
EC1D6	EC2D6	EC3D6	VO Max. of CH1.	Fill in the control input voltage of the maximum speed for the inverter. If its effective range is 0~10V, then fill in 1000 (by unit of 0.01V). If the filled value is not in the range between 0 to "V+ measured voltage", then the VO point will output 0V.
EC1D7	EC2D7	EC3D7	VO Max. of CH2.	
EC1D8	EC2D8	EC3D8	VO Max. of CH3.	
EC1D18	EC2D18	EC3D18	Identification code: K104 (If code = K240, means connecting error between Main Unit and card)	
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □ . □)	

● EC Card Register (Simple Code) Related to the VS-2TC-EC

EC1	EC2	EC3	Component Description	
EC1D0	EC2D0	EC3D0	To assign the thermocouple types for TC1~TC2.	
EC1D1	EC2D1	EC3D1	To assign the unit of temperature measurement. 0:℃; 1:℉; other values: ℃.	
EC1D2	EC2D2	EC3D2	Converted temperature value of TC1, with unit as 0.1 ℃ or 0.1 ℉.	
EC1D3	EC2D3	EC3D3	Converted temperature value of TC2, with unit as 0.1 ℃ or 0.1 ℉.	
EC1D6	EC2D6	EC3D6	To set the average times of TC1	Usable Set Value is 1~32767; other values = 5.
EC1D7	EC2D7	EC3D7	To set the average times of TC2	
EC1D17	EC2D17	EC3D17	Status and error flag	
EC1D18	EC2D18	EC3D18	Identification code: K105 (If code = K240, means connecting error between Main Unit and card)	
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □.□)	

Assign Thermocouple Type:

b15 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	TC2	TC1

Value of Nibble	0	1	2	3	4	5	6	7
Thermocouple Type	K	J	R	S	T	E	B	N

If Value of Nibble is not 0~7, the channel is disabled.

Example: If a VS-2TC-EC is installed at the EC1, and its EC1D0 is set to be H0010, then TC1: K Type of thermocouple input, TC2: J Type thermocouple input.

Status and Error Flag:

b15~b5	b4	b3	b2	b1	b0

TC1 disconnected or converted value exceeds the range

TC2 disconnected or converted value exceeds the range

The hardware error flag of this card

● EC Card Register (Simple Code) Related to the VS-4TC-EC

EC1	EC2	EC3	Component Description	
EC1D0	EC2D0	EC3D0	To assign the thermocouple types for TC1~TC4.	
EC1D1	EC2D1	EC3D1	To assign the unit of temperature measurement. 0:℃; 1:℉; other values:℃.	
EC1D2	EC2D2	EC3D2	Converted temperature value of TC1, with unit as 0.1 ℃ or 0.1 ℉.	
EC1D3	EC2D3	EC3D3	Converted temperature value of TC2, with unit as 0.1 ℃ or 0.1 ℉.	
EC1D4	EC2D4	EC3D4	Converted temperature value of TC3, with unit as 0.1 ℃ or 0.1 ℉.	
EC1D5	EC2D5	EC3D5	Converted temperature value of TC4, with unit as 0.1 ℃ or 0.1 ℉.	
EC1D6	EC2D6	EC3D6	To set the average times of TC1	Usable Set Value is 1~32767; other values = 5.
EC1D7	EC2D7	EC3D7	To set the average times of TC2	
EC1D8	EC2D8	EC3D8	To set the average times of TC3	
EC1D9	EC2D9	EC3D9	To set the average times of TC4	
EC1D17	EC2D17	EC3D17	Status and error flag	
EC1D18	EC2D18	EC3D18	Identification code: K106 (If code = K240, means connecting error between Main Unit and card)	
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □.□)	

To appoint the types of thermocouples:

b15			b0
Nibble #4	Nibble #3	Nibble #2	Nibble #1
TC4	TC3	TC2	TC1

Value of Nibble	0	1	2	3	4	5	6	7
Thermocouple Type	K	J	R	S	T	E	B	N

If Value of Nibble is not 0~7, the channel is disabled.

Example: If a VS-4TC-EC is installed at the EC1, and its EC1D0 is set to be H8100, then
TC1 & TC2: K Type of thermocouple input, TC3: J Type thermocouple input, TC4: disabled

Status and Error Flag:

b15~b5	b4	b3	b2	b1	b0

b0 — TC1 is disconnected or converted value exceeds the range
b1 — TC2 is disconnected or converted value exceeds the range
b2 — TC3 is disconnected or converted value exceeds the range
b3 — TC4 disconnected or converted value exceeds the range
b4 — The hardware error flag of this card

● EC Card Register (Simple Code) Related to the VS-1PT-EC

EC1	EC2	EC3	Component Description
EC1D0	EC2D0	EC3D0	To select the frequency of power noise to be filtered out. 0: 60Hz, 1: 50Hz; other values: 60Hz. Reduce the influence of noise from power lines. Always set the value as 1 for 50Hz AC system.
EC1D1	EC2D1	EC3D1	To assign the unit of temperature measurement. 0: ℃; 1: ℉; other values: ℃.
EC1D2	EC2D2	EC3D2	Converted temperature value of PT1, with unit as 0.1 ℃ or 0.1 ℉.
EC1D6	EC2D6	EC3D6	To set the average times of PT1. Usable Set Value is 1~32767; other values = 1.
EC1D17	EC2D17	EC3D17	Status and error flag
EC1D18	EC2D18	EC3D18	Identification code: K107 (If code = K240, means connecting error between Main Unit and card)
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □.□)

Status and Error Flag:

b15~b5	b4	b3	b2	b1	b0

b0 — TC1 is disconnected or converted value exceeds the range
b4 — The hardware error flag of this card

- EC Card Register (Simple Code) Related to the VS-2PT-EC

EC1	EC2	EC3	Component Description	
EC1D0	EC2D0	EC3D0	To select the frequency of power noise to be filtered out. 0: 60Hz, 1: 50Hz; other values: 60Hz. Reduce the influence of noise from power lines. Always set the value as 1 for 50Hz AC system.	
EC1D1	EC2D1	EC3D1	To assign the unit of temperature measurement. 0: ℃ ; 1: ℉ ; other values: ℃ .	
EC1D2	EC2D2	EC3D2	Converted temperature value of PT1, with unit as 0.1 ℃ or 0.1 ℉ .	
EC1D3	EC2D3	EC3D3	Converted temperature value of PT2, with unit as 0.1 ℃ or 0.1 ℉ .	
EC1D6	EC2D6	EC3D6	To set the average times of PT1.	Usable Set Value is 1~32767; other values = 1.
EC1D7	EC2D7	EC3D7	To set the average times of PT2.	
EC1D17	EC2D17	EC3D17	Status and error flag.	
EC1D18	EC2D18	EC3D18	Identification code: K108 (If code = K240, means connecting error between Main Unit and card)	
EC1D19	EC2D19	EC3D19	The version number of this card. (the content value □□ indicates Ver. □ . □)	

Status and Error Flag:

b15~b5	b4	b3	b2	b1	b0

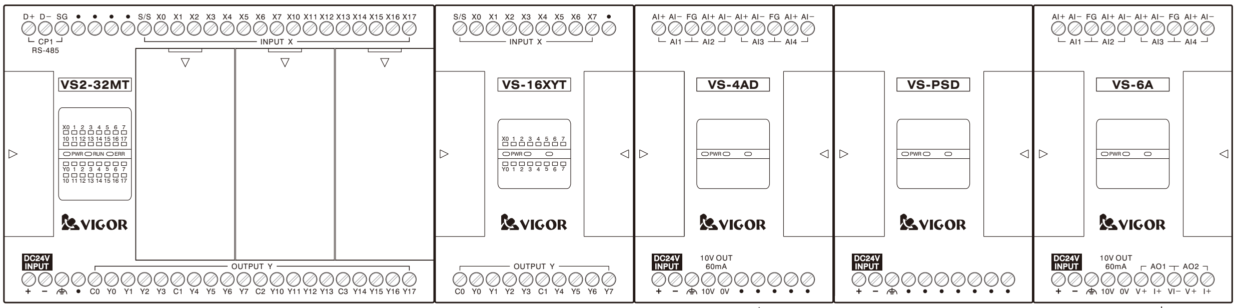
- b0: PT1 is disconnected or converted value exceeds the range
- b1: PT2 is disconnected or converted value exceeds the range
- b4: The hardware error flag of this card

2-17 Special Function Module

The VS series PLC offers various Special Function Models, such as analog input/output and temperature input. The following is the list of selectable special models.

Item	Model Name	Specifications
Special Function Module	VS-4AD	Analog Input Module: 4 channel (16-bit) inputs, each channel could input either −10~+10V, 4~20mA or −20~+20mA; isolated; with an accurate calibration DC 10V output
	VS-2DA	Analog Output Module: 2 channel (16-bit) outputs, each channel could output either −10~+10V, 4~20mA or −20~+20mA; isolated
	VS-3A	Analog I/O Module: 2 channel (16-bit) inputs + 1 channel (16-bit) output, each channel could input/output either −10~+10V, 4~20mA or −20~+20mA; isolated; with an accurate calibration DC 10V output
	VS-6A	Analog I/O Module: 4 channel (16-bit) inputs + 2 channel (16-bit) outputs, each channel could input/output either −10~+10V, 4~20mA or −20~+20mA; isolated; with an accurate calibration DC 10V output
	VS-4TC	Thermocouple Temperature Input Module: 4 channel thermocouple (K, J, R, S, T, E, B or N type) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-8TC	Thermocouple Temperature Input Module: 8 channel thermocouple (K, J, R, S, T, E, B or N type) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-2PT	PT-100 Temperature Input Module: 2 channel (3-wire PT-100) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-4PT	PT-100 Temperature Input Module: 4 channel (3-wire PT-100) inputs, 0.1℃ / 0.1℉ resolution ; isolated
	VS-2PG	Pulse Generator Module: 2 sets of 200 kHz high speed pulse outputs for 2 axes position control
	VS-4PG	Pulse Generator Module: 4 sets of 200 kHz high speed pulse outputs for 4 axes position control

All the Special and DIO Expansion Modules are serial connected on the right side of the Main Unit. The connecting sequence is without reserve. The closest Special Module is designated as the 1$^{st.}$ Special Module. Then on its right side, the following Special Module is the 2$^{nd.}$, and so on. But, the DIO Expansion Module or Power Module will not interfere to the ranking. Please pay attention to the power consumption, appropriately add the VS-PSD power repeater module is required as the picture below if the power is insufficient.



The 1$^{st.}$ Special Module
The Power Repeater Module
The 2$^{nd.}$ Special Module

There are some Buffer Memories (BFM) built-in at every Special Function Module to store the related data.
The VS series Main Unit uses the FROM/TO instruction to read/write the data in the module's BFM thus can achieve the purpose of data transfer across each other. The FROM instruction is used to read BFMs data from the designated special module. The TO instruction is used to write data into the designated BFMs at the special module. For detailed information about the FROM and TO instructions, please refer to the following pages.

Furthermore, can directly use the addressing operation to easily access the data in the special module's BFM.

Operand	Device																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D		●	●	●	○	○												

Above is the example table of Operand devices for an instruction. The device type of UnG at the table is to indicate the BFM at a Special Module could be used by the instruction directly. Thus, if the operand U1G3 is used in an instruction, that means to access the data at the BFM #3 of the 1$^{st.}$ Special Module.

Since the Ladder Master S programming software can not get the real time status from a BFM of Special Module, it is unavailable to monitor the data of operand at the UnG that is used in the instruction.

The program line on the right is to compare the data in the 1$^{st.}$ Special Module's BFM #30 with the constant value K201, and the comparison result is used to drive the coil of M0.
If that U1G30 is equal to K201, the M0 will be turned ON.

= U1G30 K201 — M0

Due to the monitor mode of the Ladder Master S could not get the value from U1G30, the in-line comparison itself in the ladder diagram would not show the result, but the real output of this line is not affected by the display.

The program line on the right is to move the value H1100 into the BFM #0 of the 2$^{nd.}$ Special Module.

M9000 — MOV H1100 U2G0

2-17-1 Buffer Memory BFM in the VS-4AD Module

BFM No.	Component Description	
#0	To assign the analog input modes of AI1~AI4. When the power is turned from OFF to ON, the default value is H0000.	
#1	To set the average times of AI1.	When the power is turned from OFF to ON, the default value is 10. The available range is 1~32,767, otherwise it is equivalent to 10.
#2	To set the average times of AI2.	
#3	To set the average times of AI3.	
#4	To set the average times of AI4.	
#5	Converted digital value of AI1 (the average times is designated by BFM #1).	
#6	Converted digital value of AI2 (the average times is designated by BFM #2).	
#7	Converted digital value of AI3 (the average times is designated by BFM #3).	
#8	Converted digital value of AI4 (the average times is designated by BFM #4).	
#30	Identification code: VS-4AD = K201 (can use the FROM instruction to check whether the place is this module or not)	
#31	The version number of this module. (the content value □□ indicates Ver. □ . □)	

BFM#0 To appoint the modes of analog inputs: (the sliding switch should also consistent with the modes)

b15	BFM#0		b0
Nibble #4	Nibble #3	Nibble #2	Nibble #1
AI4	AI3	AI2	AI1

To assign input modes

Value of Nibble	Analog Input Mode	
0	−10V~+10V voltage input	Converted digital value:−32000~+32000
1		Converted digital value:−10000~+10000
2	4mA~20mA current input	Converted digital value: 0~+16000
3	−20mA~+20mA current input	Converted digital value:−16000~+16000
4		Converted digital value:−20000~+20000
Other	Disabled	

Example: If the BFM #0 of a VS-4AD is set to be H5420, then

AI1: For −10V~+10V voltage input, that will be converted to the value −32,000~+32,000 at this mode.
AI2: For 4mA~20mA current input, that will be converted to the value 0~+16,000 at this mode.
AI3: For −20mA~+20mA current input, that will be converted to the value −20,000~+20,000 at this mode.
AI4: Disabled

2-17-2 Buffer Memory BFM in the VS-2DA Module

BFM No.	Component Description	
#20	To assign the analog output modes of AO1~AO2. When the power is turned from OFF to ON, the default value is H00.	
#21	The digital Set Value of AO1.	When the power is turned from OFF to ON, the default value is 0.
#22	The digital Set Value of AO2.	
#23	To assign the holding modes of AO1~AO2. When the power is turned from OFF to ON, the default value is H00.	
#30	Identification code: VS-2DA = K202 (can use the FROM instruction to check whether the place is this module or not)	
#31	The version number of this module. (the content value □□ indicates Ver. □ . □)	

BFM#20 To appoint the modes of analog outputs:

b15 BFM#20 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AO2	AO1

To assign output modes

Value of Nibble	Analog Output Mode	
0	–10V~+10V voltage output	Digital Set Value: −32000~+32000
1		Digital Set Value: −10000~+10000
2	4mA~20mA current output	Digital Set Value: 0~+32000
3	–20mA~+20mA current output	Digital Set Value: −32000~+32000
4		Digital Set Value: −20000~+20000
Other	Disabled	

Example: If the BFM #20 of a VS-2DA is set to be H20, then

AO1: For –10V~+10V voltage output, that will use the digital Set Value –32,000~+32,000 at this mode.

AO2: For 4mA~20mA current output, that will use the digital Set Value 0~+32,000 at this mode.

BFM#23 To appoint the output holding mode: (for the PLC status turns from RUN to STOP)

b15 BFM#23 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AO2	AO1

If the value in the nibble = 0, the channel will keep the last output, even PLC is STOP.

If the value in the nibble ≠ 0, the channel will change its digital Set Value = 0 at STOP.

2-17-3 Buffer Memory BFM in the VS-3A Module

BFM No.	Component Description	
#0	To assign the analog input modes of AI1~AI2. When the power is turned from OFF to ON, the default value is H00.	
#1	To set the average times of AI1.	When the power is turned from OFF to ON, the default value is 10. The available range is 1~32,767, otherwise it is equivalent to 10.
#2	To set the average times of AI2.	
#5	Converted digital value of AI1 (the average times is designated by BFM #1).	
#6	Converted digital value of AI2 (the average times is designated by BFM #2).	
#20	To assign the analog output mode of AO1. When the power is turned from OFF to ON, the default value is H0.	
#21	The digital Set Value of AO1.	When the power is turned from OFF to ON, the default value is 0.
#23	To assign the holding mode of AO1. When the power is turned from OFF to ON, the default value is H0.	
#30	Identification code: VS-3A = K203 (can use the FROM instruction to check whether the place is this module or not)	
#31	The version number of this module. (the content value □□ indicates Ver. □ . □)	

BFM#0 To appoint the modes of analog inputs: (the sliding switch should also consistent with the modes)

b15 BFM#0 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AI2	AI1

$\underbrace{\text{AI2} \quad \text{AI1}}_{\text{To assign input modes}}$

Value of Nibble	Analog Input Mode	
0	−10V~+10V voltage input	Converted digital value: −32000~+32000
1		Converted digital value: −10000~+10000
2	4mA~20mA current input	Converted digital value: 0~+16000
3	−20mA~+20mA current input	Converted digital value: −16000~+16000
4		Converted digital value: −20000~+20000
Other	Disabled	

Example: If the BFM #0 of a VS-3A is set to be H20, then

AI1: For −10V~+10V voltage input, that will be converted to the value −32,000~+32,000 at this mode.

AI2: For 4mA~20mA current input, that will be converted to the value 0~+16,000 at this mode.

BFM#20 To appoint the mode of analog output:

b15 BFM#20 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	Null	AO1

Value of Nibble	Analog Output Mode	
0	−10V~+10V voltage output	Digital Set Value: −32000~+32000
1		Digital Set Value: −10000~+10000
2	4mA~20mA current output	Digital Set Value: 0~+32000
3	−20mA~+20mA current output	Digital Set Value: −32000~+32000
4		Digital Set Value: −20000~+20000
Other	Disabled	

Example: If the BFM #20 of a VS-3A is set to be H2, then

AO1: For 4mA~20mA current output, that will use the digital Set Value 0~+32,000 at this mode.

BFM#23 To appoint the output holding mode: (for the PLC status turns from RUN to STOP)

b15 BFM#23 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	Null	AO1

If the value in the nibble = 0, the channel will keep the last output, even PLC is STOP.

If the value in the nibble ≠ 0, the channel will change its digital Set Value = 0 at STOP.

2-17-4 Buffer Memory BFM in the VS-6A Module

BFM No.	Component Description	
#0	To assign the analog input modes of AI1~AI4. When the power is turned from OFF to ON, the default value is H0000.	
#1	To set the average times of AI1.	When the power is turned from OFF to ON, the default value is 10. The available range is 1~32,767, otherwise it is equivalent to 10.
#2	To set the average times of AI2.	
#3	To set the average times of AI3.	
#4	To set the average times of AI4.	
#5	Converted digital value of AI1 (the average times is designated by BFM #1).	
#6	Converted digital value of AI2 (the average times is designated by BFM #2).	
#7	Converted digital value of AI3 (the average times is designated by BFM #3).	
#8	Converted digital value of AI4 (the average times is designated by BFM #4).	
#20	To assign the analog output modes of AO1~AO2. When the power is turned from OFF to ON, the default value is H00.	
#21	The digital Set Value of AO1.	When the power is turned from OFF to ON, the default value is 0.
#21	The digital Set Value of AO2.	
#23	To assign the holding modes of AO1~AO2. When the power is turned from OFF to ON, the default value is H00.	
#30	Identification code: VS-6A = K204 (can use the FROM instruction to check whether the place is this module or not)	
#31	The version number of this module. (the content value □□ indicates Ver. □.□)	

BFM#0 To appoint the modes of analog inputs: (the sliding switch should also consistent with the modes)

b15 BFM#0 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
AI4	AI3	AI2	AI1

To assign input modes

Value of Nibble	Analog Input Mode	
0	−10V~+10V voltage input	Converted digital value: −32000~+32000
1		Converted digital value: −10000~+10000
2	4mA~20mA current input	Converted digital value: 0~+16000
3	−20mA~+20mA current input	Converted digital value: −16000~+16000
4		Converted digital value: −20000~+20000
Other	Disabled	

Example: If the BFM #0 of a VS-6A is set to be H5420, then

AI1: For −10V~+10V voltage input, that will be converted to the value −32,000~+32,000 at this mode.
AI2: For 4mA~20mA current input, that will be converted to the value 0~+16,000 at this mode.
AI3: For −20mA~+20mA current input, that will be converted to the value −32,000~+32,000 at this mode.
AI4: Disabled

BFM#20 To appoint the mode of analog output:

b15 BFM#20 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AO2	AO1

To assign output modes

Value of Nibble	Analog Output Mode	
0	−10V~+10V voltage output	Digital Set Value: −32000~+32000
1		Digital Set Value: −10000~+10000
2	4mA~20mA current output	Digital Set Value: 0~+32000
3	−20mA~+20mA current output	Digital Set Value: −32000~+32000
4		Digital Set Value: −20000~+20000
Other	Disabled	

Example: If the BFM #20 of a VS-6A is set to be H20, then

AO1: For −10V~+10V voltage output, that will use the digital Set Value −32,000~+32,000 at this mode.
AO2: For 4mA~20mA current output, that will use the digital Set Value 0~+32,000 at this mode.

BFM#23 To appoint the output holding mode: (for the PLC status turns from RUN to STOP)

b15 BFM#23 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
Null	Null	AO2	AO1

If the value in the nibble = 0, the channel will keep the last output, even PLC is STOP.

If the value in the nibble ≠ 0, the channel will change its digital Set Value = 0 at STOP.

2-17-5 Buffer Memory BFM in the VS-4TC Module

BFM No.	Component Description	
#0	To assign the thermocouple types for TC1~TC4. When the power is turned from OFF to ON, the default value is H0000.	
#2	To assign the scale of temperature measurement. 0: ℃ ; 1: ℉ ; other values: ℃ . When the power is turned from OFF to ON, the default value is 0.	
#3	To set the average times of TC1.	When the power is turned from OFF to ON, the default value is 1. The available range is 1~32,767, otherwise it is equivalent to 1.
#4	To set the average times of TC2.	
#5	To set the average times of TC3.	
#6	To set the average times of TC4.	
#11	Converted temperature value of TC1, with unit as 0.1 ℃ or 0.1 ℉.	
#12	Converted temperature value of TC2, with unit as 0.1 ℃ or 0.1 ℉.	
#13	Converted temperature value of TC3, with unit as 0.1 ℃ or 0.1 ℉.	
#14	Converted temperature value of TC4, with unit as 0.1 ℃ or 0.1 ℉.	
#29	Status and error flag.	
#30	Identification code: VS-4TC = K205 (can use the FROM instruction to check whether the place is this module or not)	
#31	The version number of this module. (the content value □□ indicates Ver. □ . □)	

BFM #0 To appoint the types of thermocouples:

b15 BFM#0 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
TC4	TC3	TC2	TC1

Value of Nibble	0	1	2	3	4	5	6	7
Thermocouple Type	K	J	R	S	T	E	B	N

If Value of Nibble is not 0~7, the channel is disabled.

Example: If the BFM #0 of a VS-4TC is set to be H8100, then
TC1 & TC2: K Type thermocouple input;
TC3: J Type thermocouple input; TC4: disabled.

BFM #29 Status and Error Flag: (0: normal; 1: error)

b15	b14~b4	b3	b2	b1	b0

b15: Module's external DC 24V supply is missing or has a hardware error

b0: TC1 is disconnected or converted value exceeds the range

b1: TC2 is disconnected or converted value exceeds the range

b2: TC3 is disconnected or converted value exceeds the range

b3: TC4 is disconnected or converted value exceeds the range

2-17-6 Buffer Memory BFM in the VS-8TC Module

BFM No.	Component Description	
#0	To assign the thermocouple types for TC1~TC4. When the power is turned from OFF to ON, the default value is H0000.	
#1	To assign the thermocouple types of TC5~TC8. When the power is turned from OFF to ON, the default value is H0000.	
#2	To assign the scale of temperature measurement. 0: ℃ ; 1: ℉ ; other values: ℃ . When the power is turned from OFF to ON, the default value is 0.	
#3	To set the average times of TC1.	When the power is turned from OFF to ON, the default value is 1. The available range is 1~32,767, otherwise it is equivalent to 1.
#4	To set the average times of TC2.	
#5	To set the average times of TC3.	
#6	To set the average times of TC4.	
#7	To set the average times of TC5.	
#8	To set the average times of TC6.	
#9	To set the average times of TC7.	
#10	To set the average times of TC8.	
#11	Converted temperature value of TC1, with unit as 0.1 ℃ or 0.1 ℉.	
#12	Converted temperature value of TC2, with unit as 0.1 ℃ or 0.1 ℉.	
#13	Converted temperature value of TC3, with unit as 0.1 ℃ or 0.1 ℉.	
#14	Converted temperature value of TC4, with unit as 0.1 ℃ or 0.1 ℉.	
#15	Converted temperature value of TC5, with unit as 0.1 ℃ or 0.1 ℉.	
#16	Converted temperature value of TC6, with unit as 0.1 ℃ or 0.1 ℉.	
#17	Converted temperature value of TC7, with unit as 0.1 ℃ or 0.1 ℉.	
#18	Converted temperature value of TC8, with unit as 0.1 ℃ or 0.1 ℉.	
#29	Status and error flag.	
#30	Identification code: VS-8TC = K206 (can use the FROM instruction to check whether the place is this module or not)	
#31	The version number of this module. (the content value □□ indicates Ver. □ . □)	

BFM #0 & BFM #1 To appoint the types of thermocouples:

b15 BFM#0 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
TC4	TC3	TC2	TC1

b15 BFM#1 b0

Nibble #4	Nibble #3	Nibble #2	Nibble #1
TC8	TC7	TC6	TC5

Value of Nibble	0	1	2	3	4	5	6	7
Thermocouple Type	K	J	R	S	T	E	B	N

If Value of Nibble is not 0~7, the channel is disabled.

Example: If the BFM #0 of a VS-8TC is set to be H8100, then
TC1 & TC2: K Type thermocouple input;
TC3: J Type thermocouple input; TC4: disabled.

BFM #29 Status and Error Flag: (0: normal; 1: error)

b15	b14~b8	b7	b6	b5	b4	b3	b2	b1	b0

- b15: Module's external DC 24V supply is missing or has a hardware error
- b7: TC8 is disconnected or converted value exceeds the range
- b6: TC7 is disconnected or converted value exceeds the range
- b5: TC6 is disconnected or converted value exceeds the range
- b4: TC5 is disconnected or converted value exceeds the range
- b3: TC4 is disconnected or converted value exceeds the range
- b2: TC3 is disconnected or converted value exceeds the range
- b1: TC2 is disconnected or converted value exceeds the range
- b0: TC1 is disconnected or converted value exceeds the range

2-17-7 Buffer Memory BFM in the VS-2PT Module

<table>
<tr><th>BFM No.</th><th colspan="2">Component Description</th></tr>
<tr><td>#2</td><td colspan="2">To assign the scale of temperature measurement. 0: ℃ ; 1: ℉ ; other values: ℃ . When the power is turned from OFF to ON, the default value is 0.</td></tr>
<tr><td>#3</td><td>To set the average times of PT1.</td><td rowspan="2">When the power is turned from OFF to ON, the default value is 1.
The available range is 1～32,767, otherwise it is equivalent to 1.</td></tr>
<tr><td>#4</td><td>To set the average times of PT2.</td></tr>
<tr><td>#11</td><td colspan="2">Converted temperature value of PT1, with unit as 0.1 ℃ or 0.1 ℉.</td></tr>
<tr><td>#12</td><td colspan="2">Converted temperature value of PT2, with unit as 0.1 ℃ or 0.1 ℉.</td></tr>
<tr><td>#29</td><td colspan="2">Status and error flag.</td></tr>
<tr><td>#30</td><td colspan="2">Identification code: VS-2PT = K207 (can use the FROM instruction to check whether the place is this module or not)</td></tr>
<tr><td>#31</td><td colspan="2">The version number of this module. (the content value □□ indicates Ver. □ . □)</td></tr>
</table>

BFM #29 Status and Error Flag: (0: normal; 1: error)

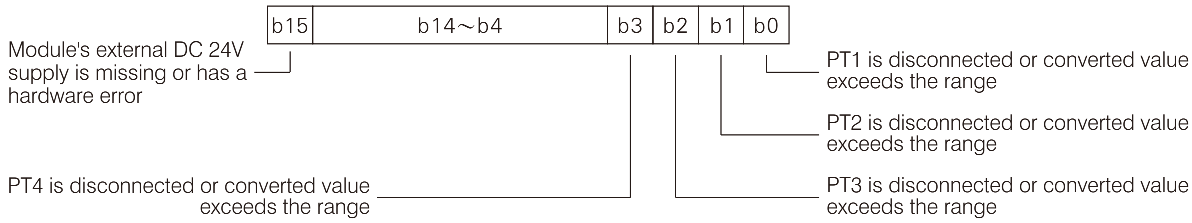
b15	b14～b2	b1	b0

- b15: Module's external DC 24V supply is missing or has a hardware error
- b1: PT2 is disconnected or converted value exceeds the range
- b0: PT1 is disconnected or converted value exceeds the range

2-17-8 Buffer Memory BFM in the VS-4PT Module

<table>
<tr><th>BFM No.</th><th colspan="2">Component Description</th></tr>
<tr><td>#2</td><td colspan="2">To assign the scale of temperature measurement. 0: ℃ ; 1: ℉ ; other values: ℃ . When the power is turned from OFF to ON, the default value is 0.</td></tr>
<tr><td>#3</td><td>To set the average times of PT1.</td><td rowspan="4">When the power is turned from OFF to ON, the default value is 1.
The available range is 1~32,767, otherwise it is equivalent to 1.</td></tr>
<tr><td>#4</td><td>To set the average times of PT2.</td></tr>
<tr><td>#5</td><td>To set the average times of PT3.</td></tr>
<tr><td>#6</td><td>To set the average times of PT4.</td></tr>
<tr><td>#11</td><td colspan="2">Converted temperature value of PT1, with unit as 0.1 ℃ or 0.1 ℉.</td></tr>
<tr><td>#12</td><td colspan="2">Converted temperature value of PT2, with unit as 0.1 ℃ or 0.1 ℉.</td></tr>
<tr><td>#13</td><td colspan="2">Converted temperature value of PT3, with unit as 0.1 ℃ or 0.1 ℉.</td></tr>
<tr><td>#14</td><td colspan="2">Converted temperature value of PT4, with unit as 0.1 ℃ or 0.1 ℉.</td></tr>
<tr><td>#29</td><td colspan="2">Status and error flag.</td></tr>
<tr><td>#30</td><td colspan="2">Identification code: VS-4PT = K208 (can use the FROM instruction to check whether the place is this module or not)</td></tr>
<tr><td>#31</td><td colspan="2">The version number of this module. (the content value □□ indicates Ver. □ . □)</td></tr>
</table>

BFM #29 Status and Error Flag: (0: normal; 1: error)



2-17-9 Buffer Memory BFM in the VS-2PG / VS-4PG Module

The VS-2PG / VS-4PG module uses the BFM to communicate with the VS Main Unit for the parameter setting and value access. The VS-2PG provides PG1 and PG2, the VS-4PG provides PG1~PG4.

At the list below, a number with the " ■ " symbol means it is a read only BFM.
The BFM#0~31 are shared by all axes.
The BFM#100~137 are specifically for the operation of PG1, the BFM#200~237 are for the PG2, the BFM#300~337 are for the PG3 and the BFM#400~437 are for the PG4.
Since the definitions of BFMs for each axis are equal, at the list below only shows the BFMs of PG1.

The BFM#150~163 are specifically for the linear interpolation operation at the paired PG1 and PG2.
The BFM#350~363 are specifically for the linear interpolation operation at the paired PG3 and PG4.
Since the definitions of BFMs for each linear interpolation group are equal, at the list below only shows the BFMs for the paired PG1 and PG2.

<table>
<tr><th>BFM #</th><th>Title</th><th colspan="2">Component Description</th><th>Default Value</th><th>Unit</th></tr>
<tr><td>1,0</td><td>MPG's Input Current Position</td><td colspan="2">32-bit data</td><td>0</td><td>Pulse</td></tr>
<tr><td>■ 2</td><td>MPG's Input Frequency</td><td colspan="2">16-bit data</td><td>0</td><td>Hz</td></tr>
<tr><td>3</td><td>MPG's Gear Ratio Numerator</td><td colspan="2" rowspan="2">1～32,767; over the range will be regarded as 1.
MPG's output pulses=Input pulses×Numerator/Denominator</td><td>1</td><td>—</td></tr>
<tr><td>4</td><td>MPG's Gear Ratio Denominator</td><td>1</td><td>—</td></tr>
<tr><td>5</td><td>MPG's Response Delay Time</td><td colspan="2">1～500ms; over the range will be regarded as 5ms.</td><td>5</td><td>ms</td></tr>
<tr><td rowspan="2">■ 6</td><td rowspan="2">MPG's Handwheel Input Status</td><td colspan="2">b0=Input from the MPG is forward (current value increase)</td><td rowspan="2">H0000</td><td rowspan="2">—</td></tr>
<tr><td colspan="2">b1=Input from the MPG is backward (current value decrease)</td></tr>
<tr><td>■ 20</td><td>PG1 and PG2 Terminal Status</td><td colspan="2">
<table>
<tr><td></td><td></td><td></td><td></td><td colspan="5">PG2</td><td colspan="5">PG1</td><td colspan="2">MPG</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>b0</td></tr>
<tr><td></td><td></td><td></td><td></td><td>CLR</td><td>INT</td><td>PG0</td><td>DOG</td><td>STR</td><td>CLR</td><td>INT</td><td>PG0</td><td>DOG</td><td>STR</td><td>B</td><td>A</td></tr>
</table>
The input terminals' status showing at this BFM will have 10ms time integration delay. That could avoid the interfering noise when it is used for the ordinary purpose input.</td><td>H0000</td><td>—</td></tr>
<tr><td>■ 21</td><td>PG3 and PG4 Terminal Status</td><td colspan="2">
<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4">PG4</td><td></td><td colspan="5">PG3</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>b0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>CLR</td><td>INT</td><td>PG0</td><td>DOG</td><td></td><td>CLR</td><td>INT</td><td>PG0</td><td>DOG</td><td>STR</td></tr>
</table>
The input terminals' status showing at this BFM will have 10ms time integration delay. That could avoid the interfering noise when it is used for the ordinary purpose input.</td><td>H0000</td><td>—</td></tr>
<tr><td>22</td><td>STR, CK, DIR and CLR Terminal Function Select</td><td colspan="2">
<table>
<tr><td></td><td colspan="3">PG4</td><td colspan="4">PG3</td><td colspan="4">PG2</td><td colspan="4">PG1</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>b0</td></tr>
<tr><td></td><td>CLR</td><td></td><td>CK&DIR</td><td>STR</td><td>CLR</td><td></td><td>CK&DIR</td><td>STR</td><td>CLR</td><td></td><td>CK&DIR</td><td>STR</td><td>CLR</td><td></td><td>CK&DIR</td></tr>
</table>
STR:0: for the PG use; 1: not for the PG use. When the STR is for the ordinary purpose input, should assign it not for the PG use.
CK,DIR,CLR:0: for the PG use; 1: operated by the BFM#23. When the CK, DIR or CLR is for the ordinary purpose output, should assign it not for the PG use.</td><td>H0000</td><td>—</td></tr>
<tr><td>23</td><td>CK, DIR and CLR Status Force Command</td><td colspan="2">
<table>
<tr><td></td><td colspan="3">PG4</td><td></td><td colspan="3">PG3</td><td></td><td colspan="3">PG2</td><td></td><td colspan="3">PG1</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>b0</td></tr>
<tr><td></td><td>CLR</td><td>DIR</td><td>CK</td><td></td><td>CLR</td><td>DIR</td><td>CK</td><td></td><td>CLR</td><td>DIR</td><td>CK</td><td></td><td>CLR</td><td>DIR</td><td>CK</td></tr>
</table></td><td>H0000</td><td>—</td></tr>
<tr><td rowspan="2">■ 30</td><td rowspan="2">Identification code</td><td>VS-2PG: K209</td><td rowspan="2">Can use the FROM instruction to check whether the place is this module or not</td><td>209</td><td rowspan="2">—</td></tr>
<tr><td>VS-4PG: K210</td><td>210</td></tr>
<tr><td>■ 31</td><td>Version</td><td colspan="2">Firmware version (the content value □□ indicates Ver. □.□)</td><td>10</td><td>—</td></tr>
</table>

※ The range of a 32-bit data is –2,147,483,648～2,147,483,647.
The range of a 16-bit data is –32,768～32,767.

<table>
<tr><th>BFM #</th><th>Title</th><th>Component Description</th><th>Default Value</th><th>Unit</th></tr>
<tr><td>101,100</td><td>Maximum Speed</td><td>Convert this speed for pulse output it must between 10~200kHz; over the range will be regarded as 200kHz.</td><td>200,000</td><td>User-defined</td></tr>
<tr><td>102</td><td>Bias Speed</td><td>Convert for the real pulse output it must between 1~200kHz; over the range will be regarded as 0.</td><td>0</td><td>User-defined</td></tr>
<tr><td>103</td><td>Acceleration Time</td><td rowspan="2">0~32,000ms
If < 0, will be regarded as 0; if > 32,000, will as 32,000.</td><td>100</td><td>ms</td></tr>
<tr><td>104</td><td>Deceleration Time</td><td>100</td><td>ms</td></tr>
<tr><td>106,105</td><td>JOG Operating Speed</td><td>Convert for the real pulse output it must between 1~200kHz</td><td>10,000</td><td>User-defined</td></tr>
<tr><td>107</td><td>JOG Start Delay Time</td><td>1~32,767ms; over the range will be regarded as 1ms.</td><td>300</td><td>ms</td></tr>
<tr><td>109,108</td><td>Home Positioning Speed</td><td>Convert for the real pulse output it must between 1~200kHz</td><td>200,000</td><td>User-defined</td></tr>
<tr><td>110</td><td>Home Positioning Creep Speed</td><td>Convert for the real pulse output it must between 1~30kHz</td><td>1,000</td><td>User-defined</td></tr>
<tr><td>111</td><td>Input No. of PG0 after DOG</td><td>1~32,767; over the range will be regarded as 1.</td><td>1</td><td>Pulse</td></tr>
<tr><td>113,112</td><td>PreSet Value of Home Position</td><td>Convert this position to the unit of pulse it must fit 32-bit data</td><td>0</td><td>User-defined</td></tr>
<tr><td>114</td><td>Speed Multiple Ratio</td><td>0.1~3,000.0%; over the range will be regarded as 100.0%.</td><td>1,000</td><td>×0.1%</td></tr>
<tr><td>115</td><td>Parameter Setting</td><td>b1,b0=Operating unit
<table>
<tr><td rowspan="2">b1</td><td rowspan="2">b0</td><td rowspan="2">Item</td><td colspan="2">Unit</td></tr>
<tr><td>Position</td><td>Speed</td></tr>
<tr><td>0</td><td>0</td><td>Motor system</td><td>Pulse</td><td>Hz</td></tr>
<tr><td>0</td><td>1</td><td>Machine system</td><td rowspan="2">● μm
● mdeg
● 10^{-4}inch</td><td>● cm/min
● 10deg/min
● inch/min</td></tr>
<tr><td>1</td><td>X</td><td>Combined system</td><td>Hz</td></tr>
</table>
<hr>
b3,b2=Multiple rate of position data
b3b2=00:X1
b3b2=01:X10
b3b2=10:X100
b3b2=11:X1,000
<hr>
b4=Rotational direction
b4=0: Increase Present Value when forward;
b4=1: Increase Present Value when backward
<hr>
b5=Home return direction
b5=0: By the direction of Present Value decreasing;
b5=1: By the direction of Present Value increasing
<hr>
b8~b6=Home return mode
<table>
<tr><td>b8</td><td>b7</td><td>b6</td><td>Home return mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>DOG Rear End home positioning</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>DOG Front End home positioning</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>DOG Rear End with PG0 count home positioning</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>DOG Front End with PG0 count home positioning</td></tr>
<tr><td>1</td><td>X</td><td>X</td><td>Data-set type home return</td></tr>
</table>
<hr>
b9=STR input type b9=0: N/O contact;
b9=1: N/C contact
<hr>
b10=DOG input type b10=0: N/O contact;
b10=1: N/C contact
<hr>
b11=PG0 input type b11=0: N/O contact;
b11=1: N/C contact
<hr>
b12=INT input type b12=0: N/O contact;
b12=1: N/C contact</td><td>H0000</td><td>—</td></tr>
<tr><td>117,116</td><td>Pulse Per Revolution of Motor</td><td>1~999,999; over the range will be regarded as 2,000.</td><td>2,000</td><td>Pulse</td></tr>
<tr><td>119,118</td><td>Distance Per Rev. of Motor</td><td>1~999,999; over the range will be regarded as 2,000.</td><td>2,000</td><td>User-defined</td></tr>
<tr><td>121,120</td><td>Target Position #1</td><td>Convert this position to the unit of pulse it must fit 32-bit data</td><td>0</td><td>User-defined</td></tr>
<tr><td>123,122</td><td>Operation Speed #1</td><td>Convert for the real pulse output it must between 1~200kHz (For the PLSV, the +/– sign of speed is forward/reverse control)</td><td>200,000</td><td>User-defined</td></tr>
<tr><td>125,124</td><td>Target Position #2</td><td>Convert this position to the unit of pulse it must fit 32-bit data</td><td>0</td><td>User-defined</td></tr>
<tr><td>127,126</td><td>Operation Speed #2</td><td>Convert for the real pulse output it must between 1~200kHz</td><td>50,000</td><td>User-defined</td></tr>
</table>

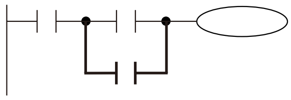
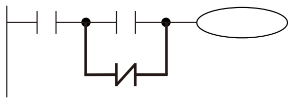
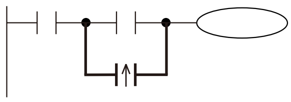
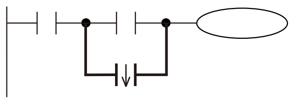
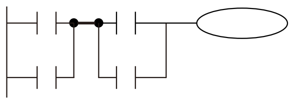
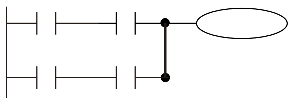
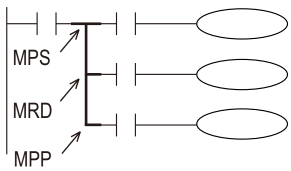
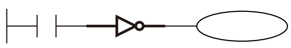
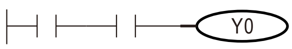
BFM #	Title	Component Description	Default Value	Unit
128	System Command	b0=Error reset, will reset at the rising edge b1=Stop, will gradually slow down and then stop b2=LSF forward limit switch, level detection b3=LSR reverse limit switch, level detection b4=Absolute/Relative positioning b4=0: Absolute positioning; b4=1: Relative positioning b5=Start command, will begin at the rising edge b6=Speed change signal of the DV2I positioning	H0000	—
129	Operation Command	b0=ZRN Home Positioning (Zero Return) b1=JOGF, Jog Forward b2=JOGR, Jog Reverse b3=DRV, Drive to Set Position b4=DRV2, Drive to Set Position by 2 Stages b5=DVIT, Interrupt Constant Quantity Positioning b6=DV2I, 2 Stages Interrupt Constant Quantity Positioning b7=DVS, Interrupt to Stop or Drive to Set Position b8=PLSV, Variable Speed Pulse Output b9=MPG, Handwheel Positioning b10=LI, Linear Interpolation Positioning	H0000	—
■ 131,130	Current Speed	32-bit data	0	User-defined
133,132	Current Location	32-bit data	0	User-defined
135,134	Current Location	32-bit data For input the servo's position, the data is read from the Main Unit's ABS instruction.	0	Pulse
■ 136	Status Information	b0=READY/BUSY 0: READY for a new instruction; 1: BUSY b1=Moving forward b2=Moving backward b3=The home positioning completed flag b4=The Current Location at the BFM#133, 132 is exceeded b5=The error flag. Also, the BFM#137 shows the error code. b6=The positioning completed flag	H0000	—
■ 137	Error Code	K0=No error K□□□1=Setting value is exceeded □□□ shows the number of incorrect BFM K□□□2=Setting value is overflow □□□ shows the number of incorrect BFM K3=More than one operation command is given K4=The LSF or LSR is activated at its watching operation Use the JOGF, JOGR or MPG instruction could relieve the limit switch and clear this error code.	0	—

BFM #	Title	Component Description		Default Value	Unit
150	Linear Interpolation's Composite Initial Speed	Convert for the real pulse output it must between 0~30kHz; over the range will be regarded as 0.		0	User-defined
152,151	Linear Interpolation's Composite Operating Speed	Convert for the real pulse output it must between 10~200kHz; over the range will be regarded as 200kHz.		200,000	User-defined
153	Linear Interpolation's Acceleration/Deceleration Time	0~32,000ms If < 0, will be regarded as 0; if > 32,000, will as 32,000.		100	ms
155,154	Linear Interpolation's Target of X-axis	Convert this position to the unit of pulse it must fit 32-bit data		0	User-defined
157,156	Linear Interpolation's Target of Y-axis	Convert this position to the unit of pulse it must fit 32-bit data		0	User-defined
■ 158	Linear Interpolation's X-axis Bias Speed	16-bit data	Result storage area, those are produced by the interpolation instruction	0	User-defined
■ 160,159	Linear Interpolation's X-axis Operating Speed	32-bit data		0	User-defined
■ 161	Linear Interpolation's Y-axis Bias Speed	16-bit data		0	User-defined
■ 163,162	Linear Interpolation's Y-axis Operating Speed	32-bit data		0	User-defined

MEMO

3. Basic Instruction

3-1 Basic Instruction Table

Mnemonic	Format	Devices	Function
LD (LOAD)		X, Y, M, S, T, C, D.b, R.b	Initial logical operation contact type NO (Normally Open)
LDI (LOAD INVERSE)		X, Y, M, S, T, C, D.b, R.b	Initial logical operation contact type NC (Normally Closed)
LDP (LOAD PULSE)		X, Y, M, S, T, C, D.b, R.b	Initial logical operation Rising edge pulse
LDF (LOAD FALLING PULSE)		X, Y, M, S, T, C, D.b, R.b	Initial logical operation Falling edge pulse
AND (AND)		X, Y, M, S, T, C, D.b, R.b	Serial connection of NO (Normally Open) contact
ANI (AND INVERSE)		X, Y, M, S, T, C, D.b, R.b	Serial connection of NC (Normally Closed) contact
ANDP (AND PULSE)		X, Y, M, S, T, C, D.b, R.b	Serial connection of Rising edge pulse
ANDF (AND FALLING PULSE)		X, Y, M, S, T, C, D.b, R.b	Serial connection of Falling edge pulse
OR (OR)		X, Y, M, S, T, C, D.b, R.b	Parallel connection of NO (Normally Open) contact
ORI (OR INVERSE)		X, Y, M, S, T, C, D.b, R.b	Parallel connection of NC (Normally Closed) contact
ORP (OR PULSE)		X, Y, M, S, T, C, D.b, R.b	Parallel connection of Rising edge pulse
ORF (OR FALLING PULSE)		X, Y, M, S, T, C, D.b, R.b	Parallel connection of Falling edge pulse
ANB (AND BLOCK)		—	Series connection of multiple parallel circuit blocks
ORB (OR BLOCK)		—	Parallel connection of multiple contact circuit blocks
MPS (POINT STORE)		—	Store the current result of the internal PLC operation
MRD (POINT READ)		—	Read the current result of the internal PLC operation
MPP (POINT POP)		—	Pop (recall and remove) the currently stored result
INV (INVERSE)		—	Invert the current result of the internal PLC operation
MEP		—	Conversion of operation result to Rising edge pulse
MEF		—	Conversion of operation result to Falling edge pulse
OUT (OUT)		Y, M, S, T, C, D.b, R.b	Final logical operation type coil drive
SET (SET)		Y, M, S, D.b, R.b	Set component permanently ON
RST (RESET)		Y, M, S, D.b, R.b, T, C, D, R, V, Z	Reset component permanently OFF
PLS (PULSE)		Y, M(Except for special M)	Rising edge pulse
PLF (PULSE FALLING)		Y, M(Except for special M)	Falling edge pulse

Mnemonic	Format	Devices	Function
MC (MASTER CONTROL)	MC N0	N0～N7	Denote the start of a master control block
MCR (MC RESET)	MCR N0	N0～N7	Denote the end of a master control block
END (END)	END	—	Force the current program scan to end
NOP (NO OPERATION)	—	—	No operation or null step

The related device table for the basic instruction

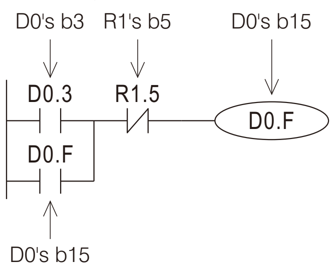
Instru-ction	Devices												
	X	Y	M	SM	S	T	16-bit C	32-bit C	D.b	R.b	D	R	V,Z
LD	●	●	●	●	●	●	●	●	○	○			
LDI	●	●	●	●	●	●	●	●	○	○			
LDP	●	●	●	●	●	●	●	●	○	○			
LDF	●	●	●	●	●	●	●	●	○	○			
AND	●	●	●	●	●	●	●	●	○	○			
ANI	●	●	●	●	●	●	●	●	○	○			
ANDP	●	●	●	●	●	●	●	●	○	○			
ANDF	●	●	●	●	●	●	●	●	○	○			
OR	●	●	●	●	●	●	●	●	○	○			
ORI	●	●	●	●	●	●	●	●	○	○			
ORP	●	●	●	●	●	●	●	●	○	○			
ORF	●	●	●	●	●	●	●	●	○	○			
OUT		●	●	●	●	●	●	○	○	○			
SET		●	●	●	●				○	○			
RST		●	●	●	●	●	●	●	○	○	○	○	○
PLS		●	●										
PLF		●	●										

○ Means it cannot use the V, Z Index Register to modify ● Means it could use the V, Z Index Register to modify

To establish the Set Value of OUT T or OUT C instruction can use the K, D or R, also which can use the Index Register V, Z to modify.

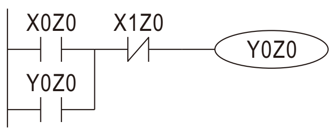
The basic instructions of the VS series PLC provide the “Bitwise Operation” and the “Bit Index” function. That greatly improving the convenience of programming, but also greatly enhances the overall performance.

- Since a register has 16 bits and they are allocated to 0~F (b0~b15) respectively, the “Bitwise Operation” function is to treat the individual bit in a register D or R as a bit component to process, as shown in the example below:



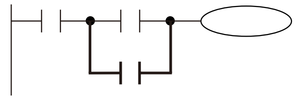
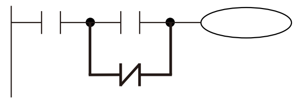
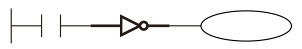
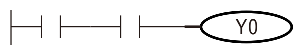
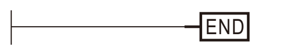
The left diagram shows the self-holding circuit about the b15 of D0. That starts from the b3 of D0 and may release by the b5 of R1, then gets a new output to the b15 of D0.

- The “Bit Index” function is to let the bit operand in the basic instruction has V, Z modifiable capability, as shown in the example below:

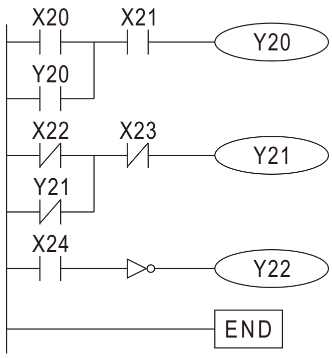


When the contents of the Z0 is 0, the self-holding circuit starts from the X0 and may release by the X1, then gets a new output to the Y0. When the contents of the Z0 is 5, the self-holding circuit starts from the X5 and may release by the X6, then gets a new output to the Y5.

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

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

Ladder Diagram

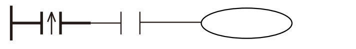
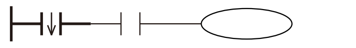
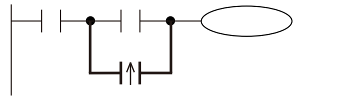
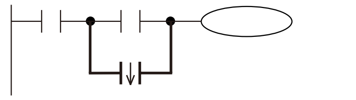


Instruction List

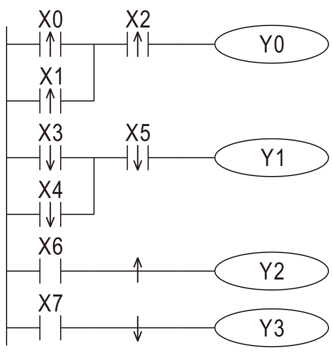
LD	X20	Initial logical operation contact type NO (Normally Open)
OR	Y20	Parallel connection of NO (Normally Open) contact
AND	X21	Serial connection of NO (Normally Open) contact
OUT	Y20	Final logical operation type coil drive
LDI	X22	Initial logical operation contact type NC (Normally Closed)
ORI	Y21	Parallel connection of NC (Normally Closed) contact
ANI	X23	Serial connection of NC (Normally Closed) contact
OUT	Y21	Final logical operation type coil drive
LD	X24	Initial logical operation contact type NO (Normally Open)
INV		Invert the current result of the internal PLC operation
OUT	Y22	Final logical operation type coil drive
END		Force the current program scan to end

- The OUT instruction for the coil of T or C will have an explicit explanation in the section 3-9 "The OUT and RST Instructions for the Timer or Counter".
- When the PLC executes the END instruction, that will force to terminate the current scan and process both output and input component updates. All instructions in the program and subroutines after the END instruction will not be executed.
- The END instruction can be inserted into the middle of the program, it can be used to do the programming debug. The program section after the END instruction is disabled and isolated from the whole program.

3-3 The LDP, LDF, ANDP, ANDF, ORP, OPF, MEP and MEF Instructions

Mnemonic	Format	Devices	Function
LDP (LOAD PULSE)		X, Y, M, S, T, C, D.b, R.b	Initial logical operation Rising edge pulse
LDF (LOAD FALLING PULSE)		X, Y, M, S, T, C, D.b, R.b	Initial logical operation Falling edge pulse
ANDP (AND PULSE)		X, Y, M, S, T, C, D.b, R.b	Serial connection of Rising edge pulse
ANDF (AND FALLING PULSE)		X, Y, M, S, T, C, D.b, R.b	Serial connection of Falling edge pulse
ORP (OR PULSE)		X, Y, M, S, T, C, D.b, R.b	Parallel connection of Rising edge pulse
ORF (OR FALLING PULSE)		X, Y, M, S, T, C, D.b, R.b	Parallel connection of Falling edge pulse
MEP		—	Conversion of operation result to Rising edge pulse
MEF		—	Conversion of operation result to Falling edge pulse

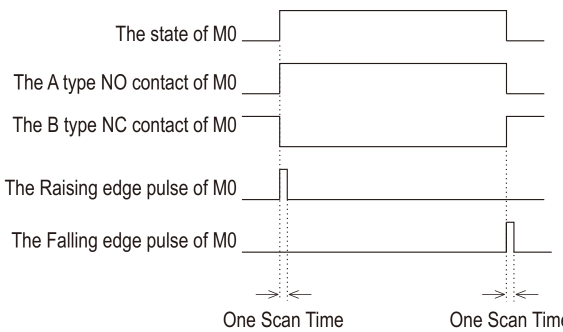
Ladder Diagram



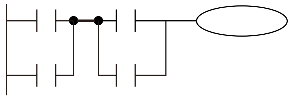
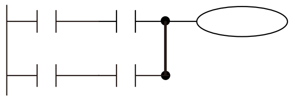
Instruction List

LDP	X0	Initial logical operation Rising edge pulse contact
ORP	X1	Parallel connection of Rising edge pulse contact
ANDP	X2	Serial connection of Rising edge pulse contact
OUT	Y0	Final logical operation type coil drive
LDF	X3	Initial logical operation Falling edge pulse contact
ORF	X4	Parallel connection of Falling edge pulse contact
ANDF	X5	Serial connection of Falling edge pulse contact
OUT	Y1	Final logical operation type coil drive
LD	X6	Initial logical operation contact type NO (Normally Open)
MEP		Conversion of operation result to Rising edge pulse contact
OUT	Y2	Final logical operation type coil drive
LD	X7	Initial logical operation contact type NO (Normally Open)
MEF		Conversion of operation result to Rising edge pulse contact
OUT	Y3	Final logical operation type coil drive

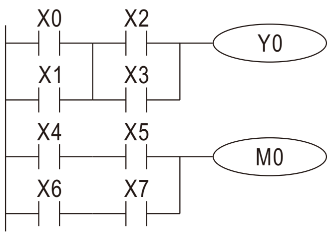
- The right figure takes the bit device M0 as an example to illustrate the relationship between the initial logical operation contact type NO, NC, the Raising and Falling edge pulse contacts.
- The Rising edge contact will be active at the program for one Scan Time when the associated device status changes from “OFF” to “ON”.
- The Falling edge contact will be active at the program for one Scan Time when the associated device status changes from “ON” to “OFF”.



3-4 The ANB and ORB Instructions

Mnemonic	Format	Devices	Function
ANB (AND BLOCK)		—	Series connection of multiple parallel circuit blocks
ORB (OR BLOCK)		—	Parallel connection of multiple contact circuit blocks

Ladder Diagram

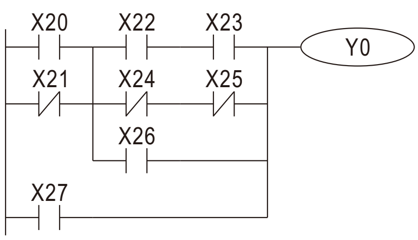


Instruction List

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

- To declare the starting points of the circuit block, please use an LD, LDI, LDP or LDF instruction. After completing the serial circuit block, to connect it to the preceding block by series/parallel connection should use the ANB/ORB instruction.
- No more than 8 of the ANB and ORB instructions in a batch circuit.
- Please refer to the following program example, it is used both the ANB and ORB instructions in a circuit block.

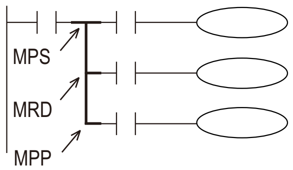
Ladder Diagram



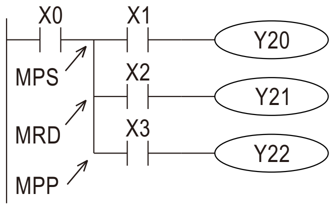
Instruction List

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

3-5 The MPS, MRD and MPP Instructions

Mnemonic	Format	Devices	Function
MPS (POINT STORE)		—	Store the current result of the internal PLC operation
MRD (POINT READ)		—	Read the current result of the internal PLC operation
MPP (POINT POP)		—	Pop (recall and remove) the currently stored result

Ladder Diagram

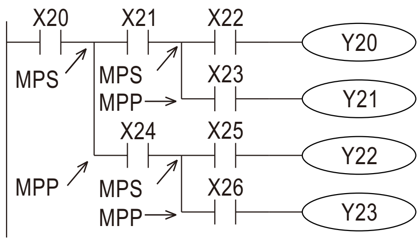


Instruction List

LD	X0	Initial logical operation contact type NO (Normally Open)
MPS		Store the current result of the internal PLC operation
AND	X1	Serial connection of NO (Normally Open) contact
OUT	Y20	Final logical operation type coil drive
MRD		Read the current result of the internal PLC operation
AND	X2	Serial connection of NO (Normally Open) contact
OUT	Y21	Final logical operation type coil drive
MPP		Pop (recall and remove) the currently stored result
AND	X3	Serial connection of NO (Normally Open) contact
OUT	Y22	Final logical operation type coil drive

- The MPS instruction stores the state of a 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).
- The following diagram shows the program example block with three connections:

Ladder Diagram



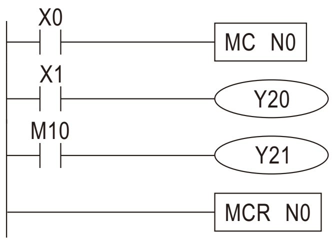
Instruction List

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

3-6 The MC and MCR Instructions

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

Ladder Diagram

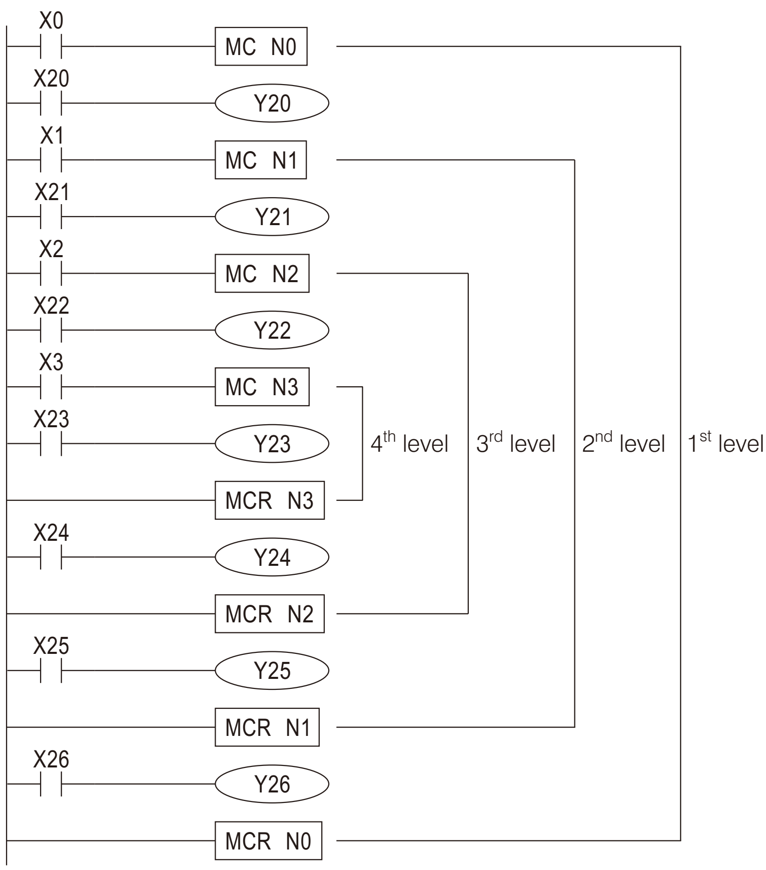


Instruction List

LD	X0	The X0 is the condition contact
MC	N0	Become a master control block which is controlled by the 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 the condition contact X0 is “ON”, all instructions between the MC and MCR instructions will be executed normally.
- When the condition contact X0 is “OFF”, all instructions between the paired MC and MCR instructions will NOT be executed, also all ordinary Timers and the coils which is driven by the OUT instruction will be reset; but the Retentive Timers, Counters and the status of coils which is driven by the SET / RST instruction will be kept.
- Use the MC instruction to shift the bus line (the Initial logical operation) to a point behind the conditional contact and then use the paired MCR instruction to return to the original bus line.
- A master control block allows to contain another master control block inside, which is to form a nest level. This structure at the most can have 8 levels N0~N7. The top nest level shall be the N0, and followed by N1, N2..., so the most inner level shall be the N7.
- A program example with a multiple nest levels structure is shown below:

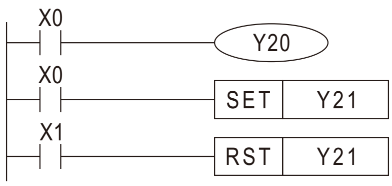
Ladder Diagram



3-7 The SET and RST Instructions

Mnemonic	Format	Devices	Function
SET (SET)	SET Y0	Y, M, S, D.b, R.b	Set component permanently ON
RST (RESET)	RST Y0	Y, M, S, D.b, R.b, T, C, D, R, V, Z	Reset component permanently OFF

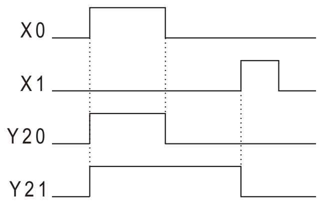
Ladder Diagram



Instruction List

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

Active I/O duration time sheet

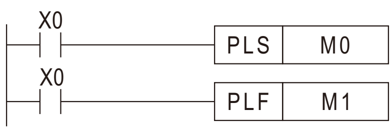


- The SET instruction sets the output coil permanently "ON" when it has been operated.
- The RESET instruction resets the output coil permanently "OFF" and resets the Present Value of a Timer, Counter or Register to zero.
- The SET instruction and the RESET instruction can use the same output coil, and the number of times is unlimited.
- The RST instruction for the device C will have an explicit explanation in the section 3-9 "The OUT and RST Instructions for the Timer or Counter".

3-8 The PLS and PLF Instructions

Mnemonic	Format	Devices	Function
PLS (PULSE)	PLS Y0	Y, M (Except the Special M)	Rising edge pulse
PLF (PULSE FALLING)	PLF Y0	Y, M (Except the Special M)	Falling edge pulse

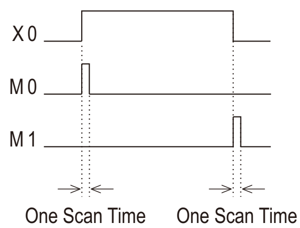
Ladder Diagram



Instruction List

LD	X0
PLS	M0
LD	X0
PLF	M1

Active I/O duration time sheet



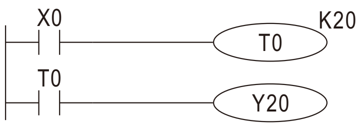
- When X0 = OFF → ON, the M0 will output a pulse for one Scan Time.
- When X0 = ON → OFF, the M1 will output a pulse for one Scan Time.

3-9 The OUT and RST Instructions for the Timer or Counter

If the OUT instruction is used for the coil of the component T or C, input a Set Value is required.

Timer

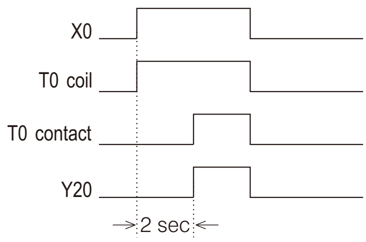
Ladder Diagram



Instruction List

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

Active I/O duration time sheet



- The Set Value of a Timer can be set by either using a K (Constant) or a Data Register D, R (Parameter).
- The available range of the Set Value:

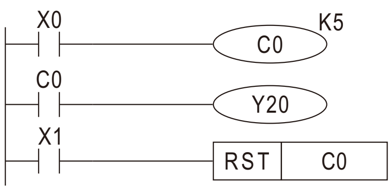
Timer ID No.	Timing Unit	Type of Timer	Available Range	Real Setting Time
T0~T199	100ms	General Timer (non-retentive)	1~32,767 (The Set Value beyond this range will be defaulted to 0) ※	0.1~3276.7 sec.
T200~T245	10ms			0.01~327.67 sec.
T256~T511	1ms			0.001~32.767 sec.
T246~T249	1ms	Retentive Timer		0.001~32.767 sec.
T250~T255	100ms			0.1~3276.7 sec.

- To reset the contact or Present Value of a Retentive Timer T246~T255, must use the RST instruction.

※ If the timer's Set Value is 0, its contact will become activated at the next scan time after its coil is triggered.

Counter

Ladder Diagram



Instruction List

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

Action Description

- When the X0 is turned from “OFF” to “ON”, the C0 executes up count once, until the Present Value of the C0 is equal to 5 that will turn its output contact to “ON”, where the Present Value will not increase anymore and its contact will stay permanently “ON”.
- When X1 = “ON”, the Present Value of C0 will be reset to “0” and the contact of C0 will become “OFF”.

- The Set Value of a Counter can be set by either using a K (Constant) or a Data Register D, R (Parameter).
- The available range of the Set Value:

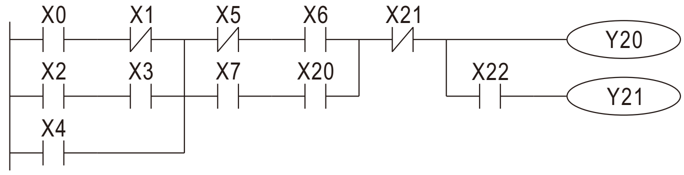
Counter ID No.	Type of the Counter		Available Range
C0~C99	General	16-bit Up count	1~32,767 (The Set Value beyond this range will be defaulted to 1)
C100~C199	Latched		
C200~C219	General	32-bit Up/Down count	−2,147,483,648~2,147,483,647
C220~C234	Latched		
C235~C255	Software High Speed Counter (Latched)		

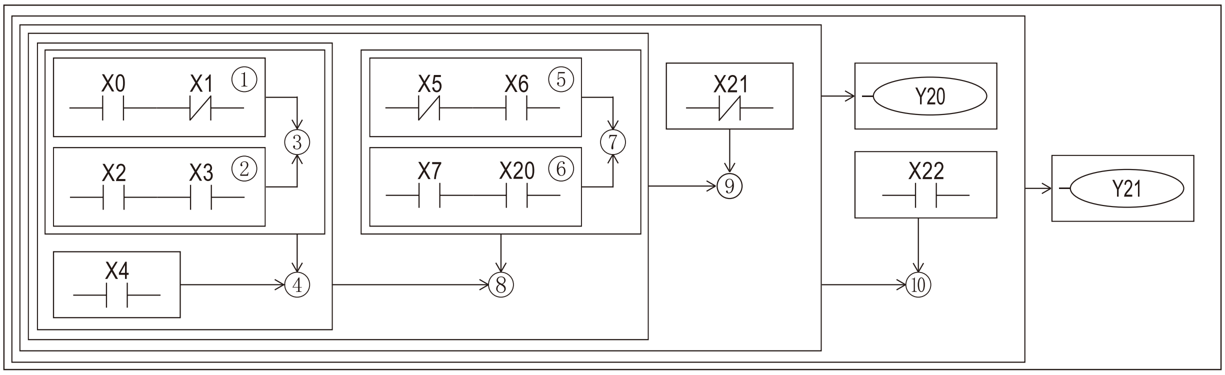
- When using High Speed Counters, please refer to the section 2-7 "Software High Speed Counter".

3-10 Significant Notes for Programming

3-10-1 Convert the Ladder Diagram to the Instruction List

The rule to convert a program from the Ladder Diagram to the Instruction List format should follow the sequence that from left to right and from top to bottom.

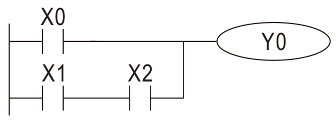
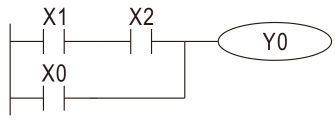




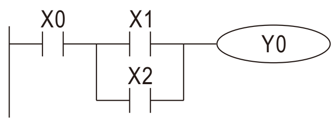
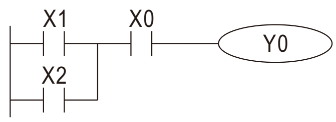
LD	X0	Serial connection Block ①	
ANI	X1		③
LD	X2	Serial connection Block ②	
AND	X3		④
ORB		Parallel connection circuit blocks	
OR	X4	Parallel connection of NO contact	⑧
LDI	X5	Serial connection Block ⑤	
AND	X6		⑦
LD	X7	Serial connection Block ⑥	
AND	X20		⑨
ORB		Parallel connection circuit blocks	
ANB		Serial connection of multiple parallel blocks	
ANI	X21	Serial connection of NC contact	⑩
OUT	Y20	Output	
AND	X22	Serial connection of NO contact	
OUT	Y21	Output	

3-10-2 Programming Techniques

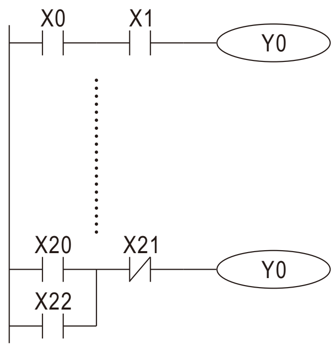
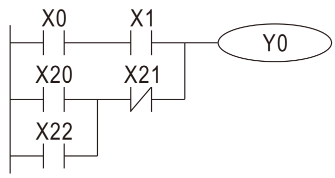
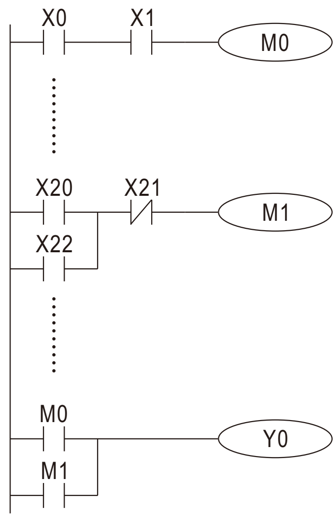
1. Put a section which with longer serial connections of contacts at the upper place of a Parallel Connection Circuit Blocks. This way will make the programming simpler and easier.

Ladder Diagram	Instruction List	Ladder Diagram	Instruction List
	LD X0 LD X1 AND X2 ORB OUT Y0		LD X1 AND X2 OR X0 OUT Y0

2. It is recommended to place a circuit with more parallel connection of contacts on the left side.

Ladder Diagram	Instruction List	Ladder Diagram	Instruction List
	LD X0 LD X1 OR X2 ANB OUT Y0		LD X1 OR X2 AND X0 OUT Y0

3. Although it is not an incorrect syntax to repetitively use an output coil ID number or Double Coil, the last output operation is the real effective coil. Therefore, the structure of conditional signal contacts should be modified or the output coils of the same ID number should be avoided.

Ladder Diagram	Instruction List	Ladder Diagram	Instruction List
	LD X0 AND X1 OUT Y0 ⋮ LD X20 OR X22 ANI X21 OUT Y0	 or 	LD X0 AND X1 LD X20 OR X22 ANI X21 ORB OUT Y0 LD X0 AND X1 OUT M0 ⋮ LD X20 OR X22 ANI X21 OUT M1 ⋮ LD M0 OR M1 OUT Y0

MEMO

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

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, that is quite a difficult job to learn such a complicated sequential control design for machinery engineers, therefor the SFC (Sequential Function Chart) is developed accordingly.

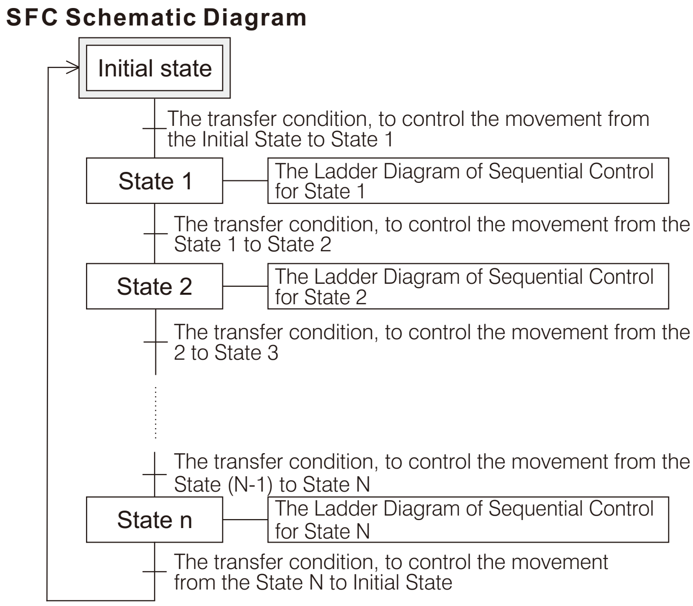
4-1 What is the Sequential Function Chart (SFC)

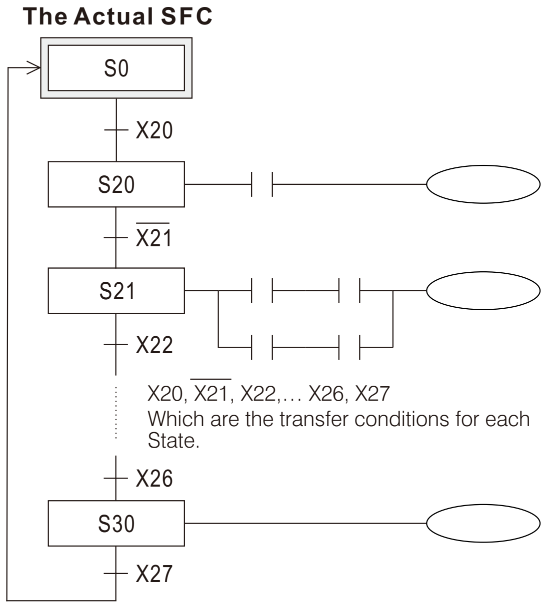
The SFC is designed to create an easy way to understand the machine's movements, which has the following features:

(1) It is not necessary to design the special sequence for stepladders of which state constantly changes, the PLC will automatically execute interlocks and double coils under different states. Only need to simply design the sequence function of every state.

(2) Even a person who is not a machine designer can easily learn all actions and conduct jobs such as trials, adjustment, debug and maintenance.

4-1-1 The Framework of SFC





The left is a Schematic Diagram of SFC and the right diagram is the corresponding actual SFC. The PLC will execute to start from the Initial State, then complete State 1 → State 2 → … → State N in the sequence based on states' transfer 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 State refers to a state which is executive or active during the PLC processing. 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”.

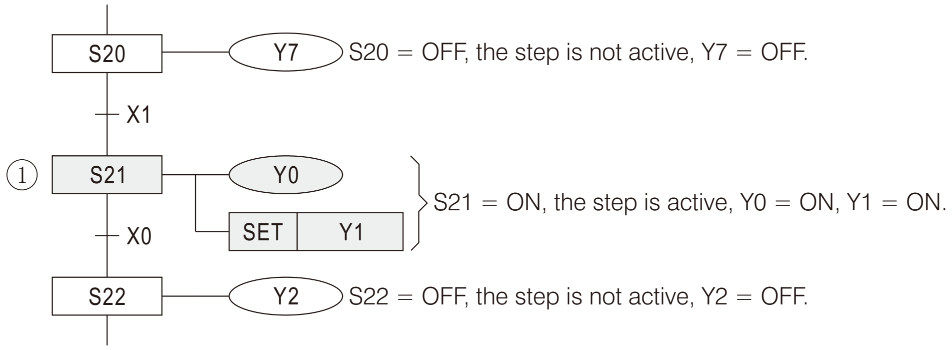
③ Transferring the machine action to the next state when the transfer condition is authorized. In generally there is a connecting line to connect the states, and it indicates the direction of the signal.

2. Transfer Condition

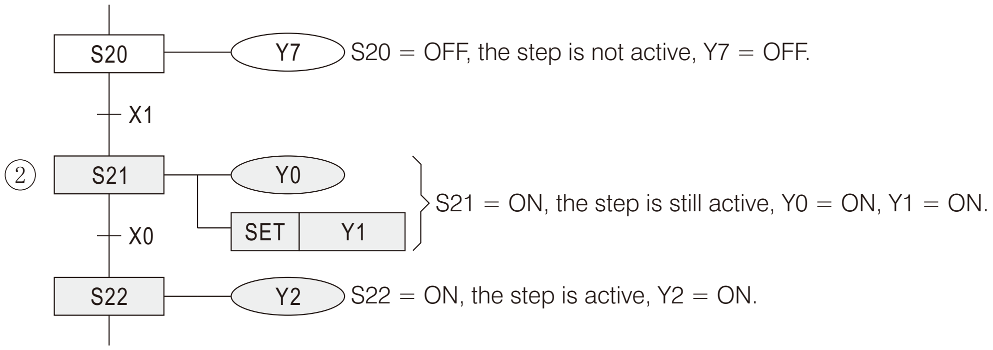
There is a line segment to connect between states, and on the line segment put a perpendicular short line which is used to express the related condition driving the states transfer.

4-1-3 State and Action of SFC

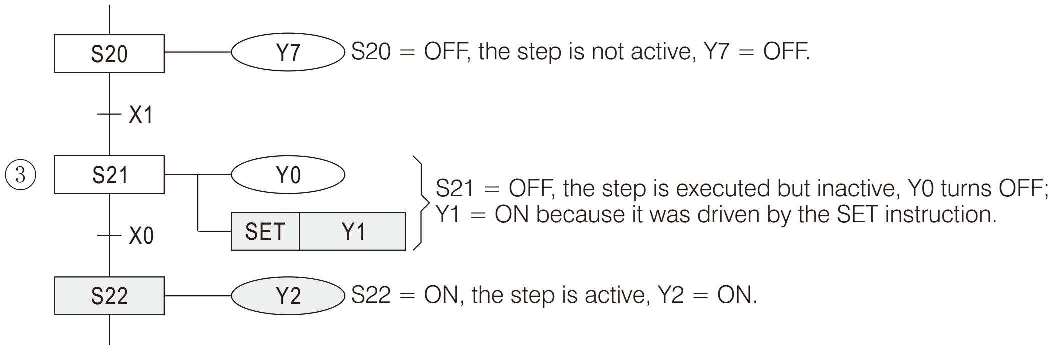
The VS series PLC uses step relay to indicate the state in the SFC. The diagrams below show the action of step relays under various of situations.



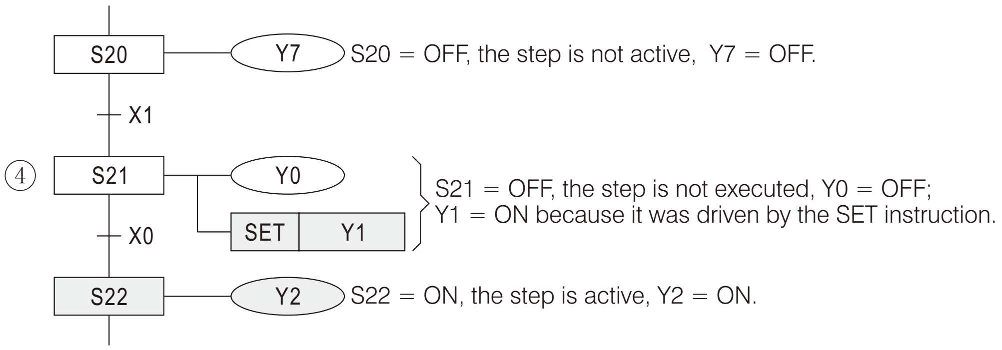
At the moment X0 = OFF→ON, the active step will be transferred from the S21 to S22. The relay S22 truns from OFF to ON, but S21 remains ON temporary.



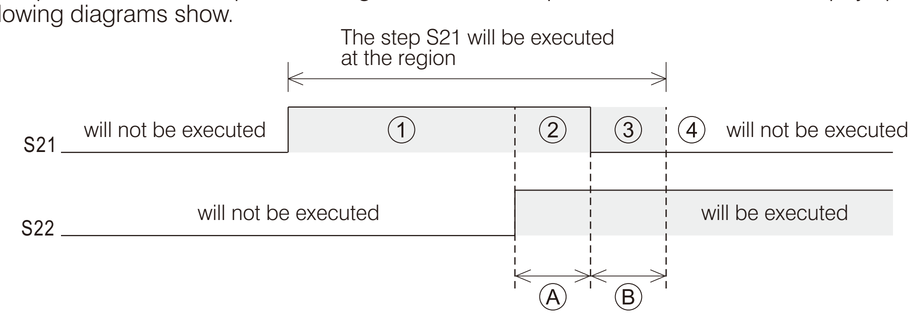
At the next scan time, the step S21 turns from ON to OFF (inactive), but S21 is still executed.



Then the next scan time, the step S21 will not be executed.



In the process of SFC step transferring, there are some phenomena which need to pay special attention to, as the following diagrams show.



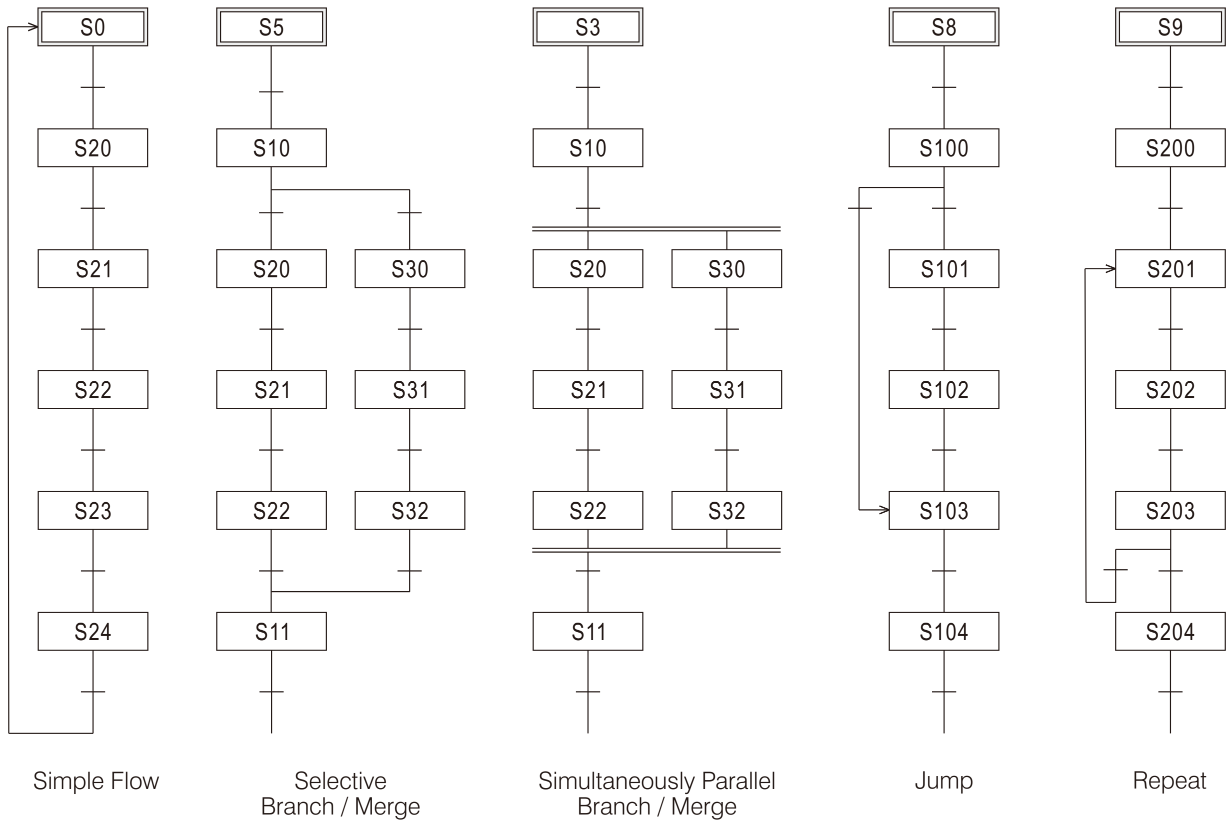
Ⓐ: It is a Scan Time, the activated step is transferred at the scan time. Both S21 and S22 are ON at this scan time.

Ⓑ: Another Scan Time, it is coming after the state transfer. Although the S21 has turned OFF, it will still execute the program within the step; meanwhile, it turns Y0 from ON to OFF.

The devices driven by the OUT instruction in the step will turn OFF (OUT Y0 in the program) for the next scan time after the state transfer; but the devices driven by the SET instruction will remain ON (SET Y1 in the program).

4-1-4 Types of SFC

Based on the flow control methods of SFC, SFC has five basic types:

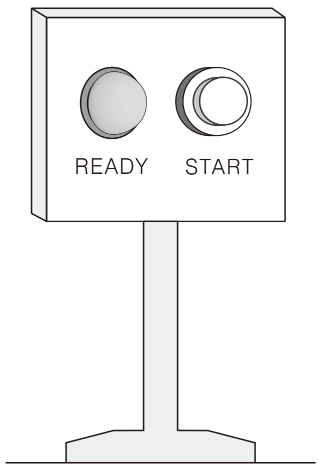


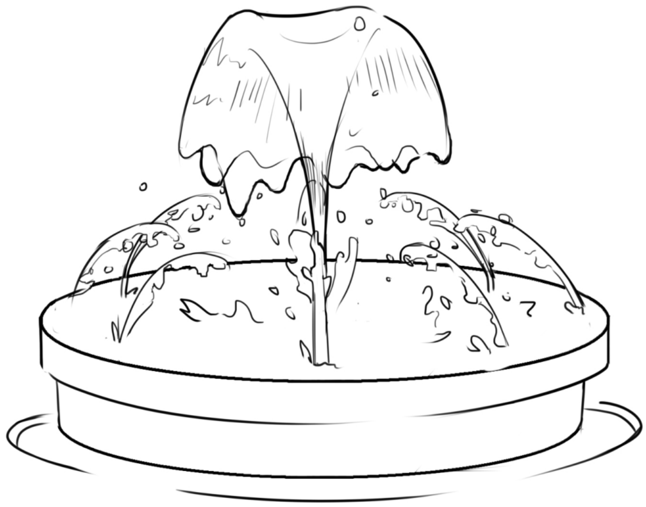
4-2 Compiling the Sequential Function Chart (SFC)

The SFC has its special operating method. Through the step-by-step description of compiling the SFC program in this chapter, designers can understand the methods of the SFC programming. Here, we use the application example of a fountain in park to explain the design procedure of the SFC in sequence.

The description of the park fountain function:
There is a control panel next to the park fountain and a START button on the panel for people to start up the fountain show, also a READY indicator to display whether it is under the standby state or not. When people press the start button during the standby status, a cycle of the fountain lighting and spray show will start. After a while, it is back to the standby status.

When the fountain show is activated, a series procedures will start to perform. First, the system turns to the busy status (the READY indicator OFF) and illuminates the fountain's lights. 5 seconds later, the surrounding water nozzles begin to spray. After 10 seconds, the central nozzle joins up. All the lights and water spraying will run for 20 seconds then stop, the fountain returns to the standby status and the READY indicator turns ON again.





The operating procedures for the fountain control are as follows:

(1) When the power is ON, the PLC is in the standby status and the READY indicator is ON.

(2) If someone presses the START button, the fountain's lights are illuminated and a 5 second timer is activated.

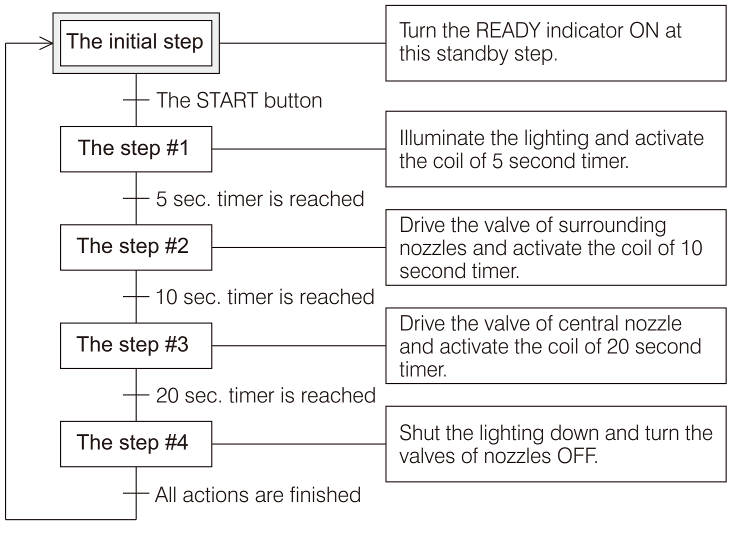
(3) When the 5 second timer is reached, the surrounding water nozzles begin to spray and a 10 second timer is activated.

(4) When the 10 second timer is reached, the central nozzle begins to spray too, and a 20 second timer is activated.

(5) After the 20 second timer is reached, all the lights and nozzles are turned OFF and the PLC comes back to the standby status.

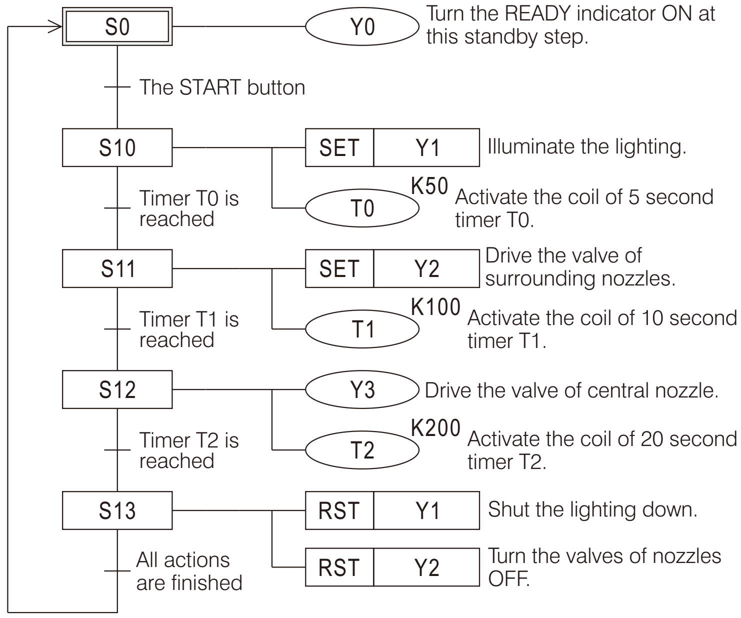
4-2-1 Create the Flowchart of the Sequential Procedures

According to the description of the fountain control actions, we can draw the flowchart as below.
First of all, we divide those continuous actions to be completed into few individual steps. Then, insert a control condition between every two steps that could transfer activating state of a step into next step. By this method, the following procedure flowchart is produced.



4-2-2 Convert the Flowchart to the SFC

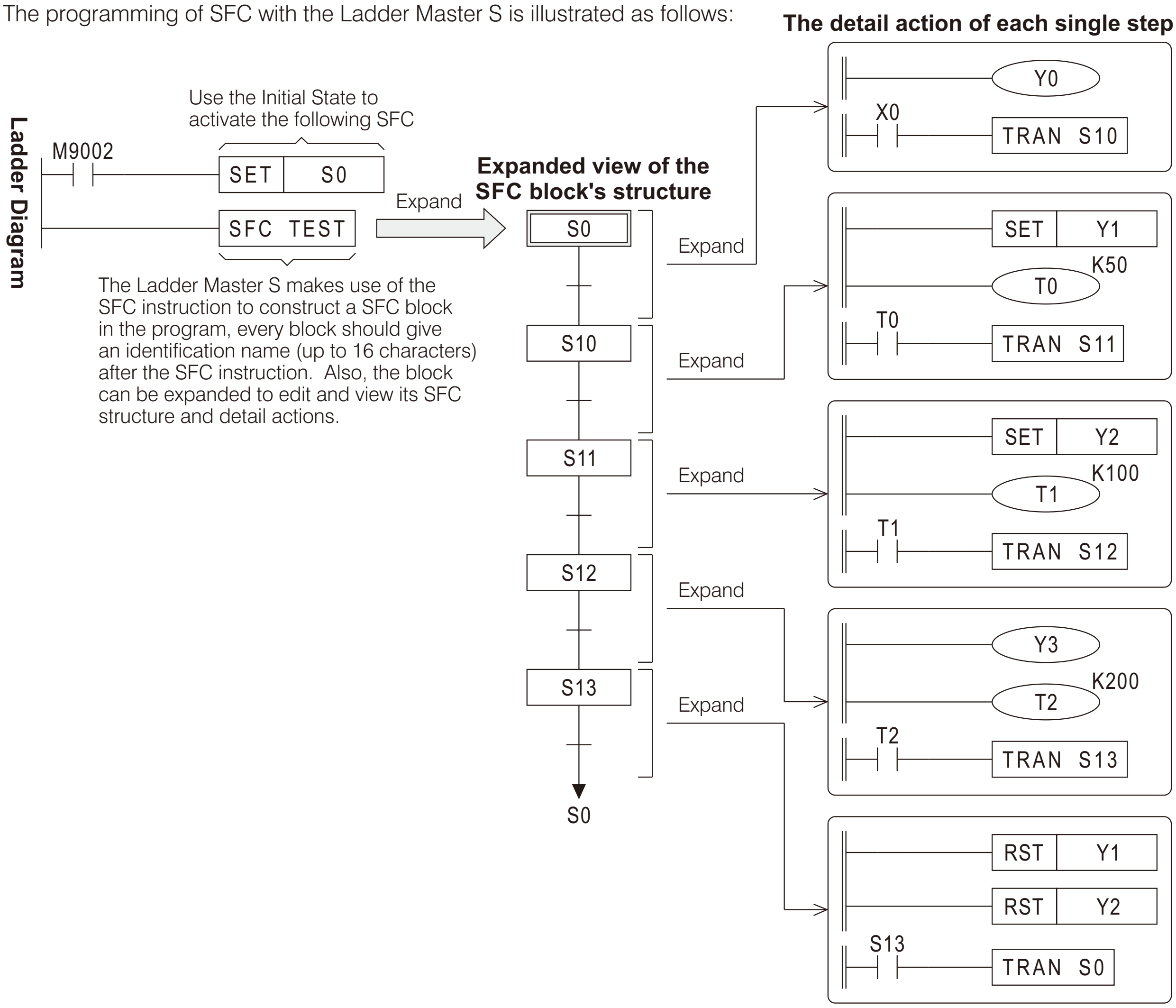
In an actual application, the external input signals and output loads must be wired to the PLC's input and output terminals, therefore it could receive the state of external signal and actually drive the load.
In this example, the START button connects to the input point X0, the output Y0 is for the READY indicator, the Y1 is for the lighting, the Y2 is for the valve of surrounding nozzles and the Y3 is for the valve of central nozzle.
The state of each Step Relay in the program indicates which SFC procedure is active, and uses the Transferring Condition under the step to move the effective status to next Step Relay.
At the system, the General and Latched Step Relays (S10~S899, S1000~S4095) can be used without to notice about the order of the component number, but the S0~S9 are specially assigned to the Initial State. Besides, should not reuse the same component number for the steps.
According to the note above and the I/O control requirements, could convert the flowchart to the right side SFC program.
Those components T0~T2 are the PLC built-in timers, the characteristic of this timer type is up counting by the unit of 0.1 seconds.



4-2-3 Use the Ladder Master S to Edit the SFC

The programming software Ladder Master S provides the function to construct the SFC. Thus, the SFC at the previous section can use the Ladder Master S to compose at a computer, then load it to a PLC to test run and debug.

The programming of SFC with the Ladder Master S is illustrated as follows:



4-3 Description the Application Types of SFC

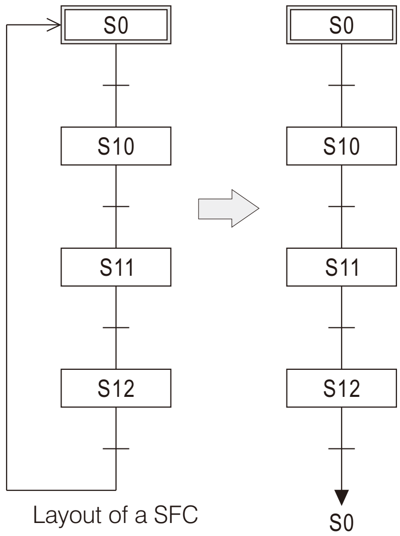
This section introduces various of processing flow modes about the SFC and the related precautions when to compile it.

4-3-1 Single Flow, Jump and Repeat Flow During the Flowchart Transfer

The Single Flow is the basic construction of a SFC and it is used in simple sequential controls.
In the actual layout of the SFC, the JUMP instruction (represented by ↓) is used to specify a jump or repeat about the process of steps flow .

If the JUMP instruction at the Simple Flow is used to move back to an earlier step, that is called a Repeat Flow.

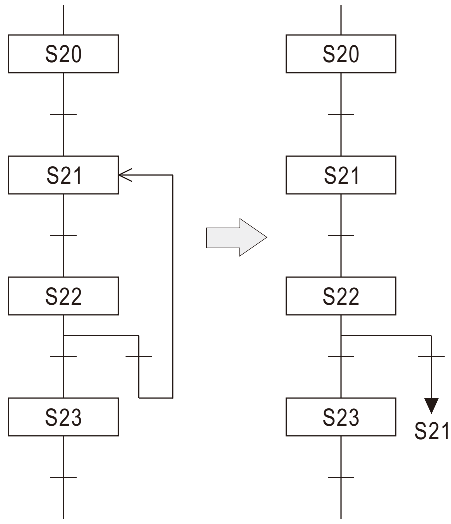
Simple Flow



Layout of a SFC

Pattern at the Ladder Master S

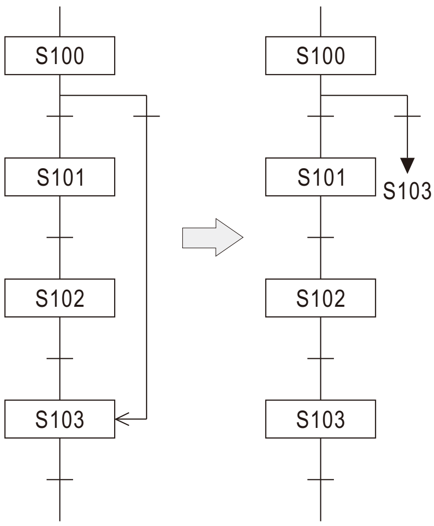
Repeat Flow



Layout of a SFC

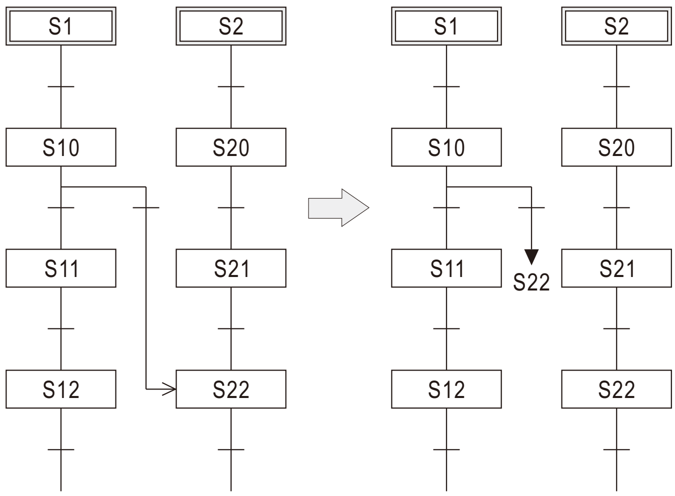
Pattern at the Ladder Master S

If the JUMP instruction at the Simple Flow is used to move forward to a later step or to another SFC, that is called a Jump Flow.



Layout of a SFC

Pattern at the Ladder Master S

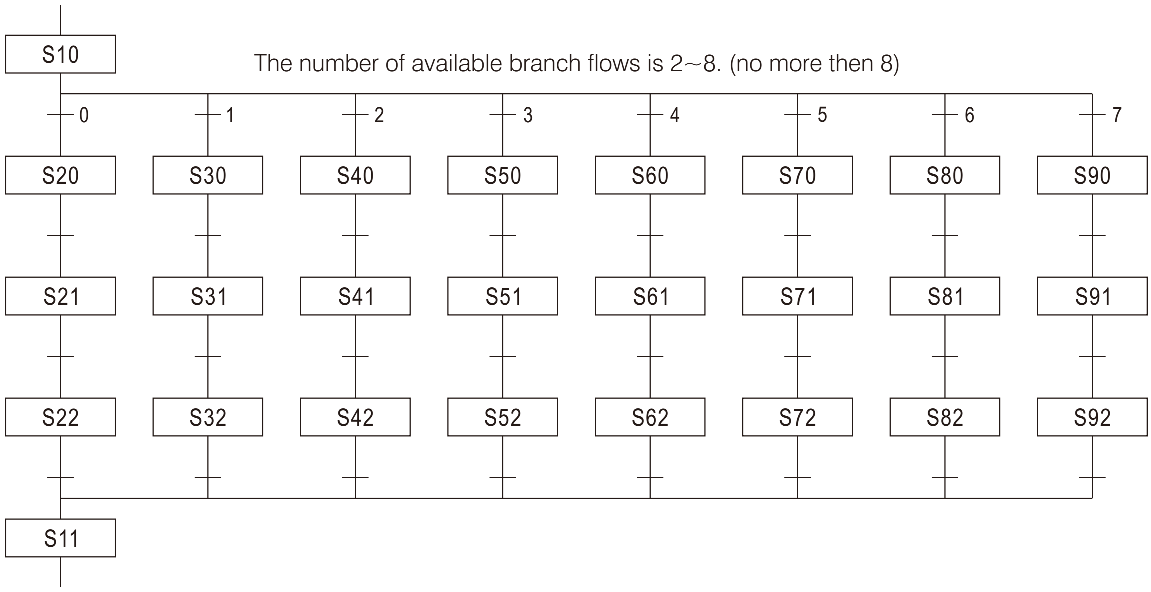


Layout of SFCs

Pattern at the Ladder Master S (separately create two SFCs)

4-3-2 Selective Branch and Merge

Selective Branch : Select one of the branch flows to transfer the effective state.
Selective Merge : Merge a number of branch flows into a single flow.



When the step relay S10 is effective, its following eight transfer conditions are simultaneously used for to select and move the effective state to either one of the S20, S30, S40, S50, S60, S70, S80 or S90.

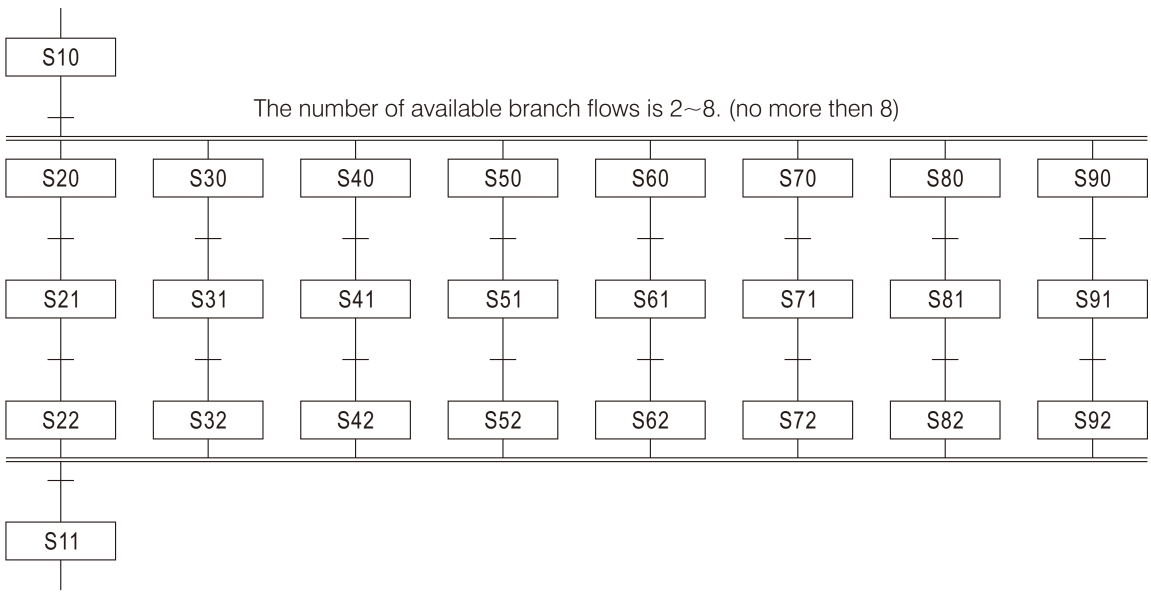
Assume the S10 is effective and the transfer condition “0” is just turning ON, the activation will be transferred from the S10 to the S20, then moved to the S21 after that to the S22. At this time, the transfer condition under the S22 turns ON, the activation will move to the S11. Suppose the S10 is effective and the transfer condition “1” is ON, the activation will be transferred from the S10 to the S30, and so on.

The entry transfer conditions of every Selective Branch must be particular and unique, that should avoid more than one transfer condition is effective at the same time. If not, will lead to more than one flows executing at the same time and result in errors.

4-3-3 Simultaneously Parallel Branch and Merge

Simultaneously Parallel Branch : When the foremost transfer condition is authorized, all the first steps of branch flows become effective.

Simultaneously Parallel Merge : When all the last steps of branch flows are effective and the merge transfer condition is authorized, the branches are combined to a single flow.

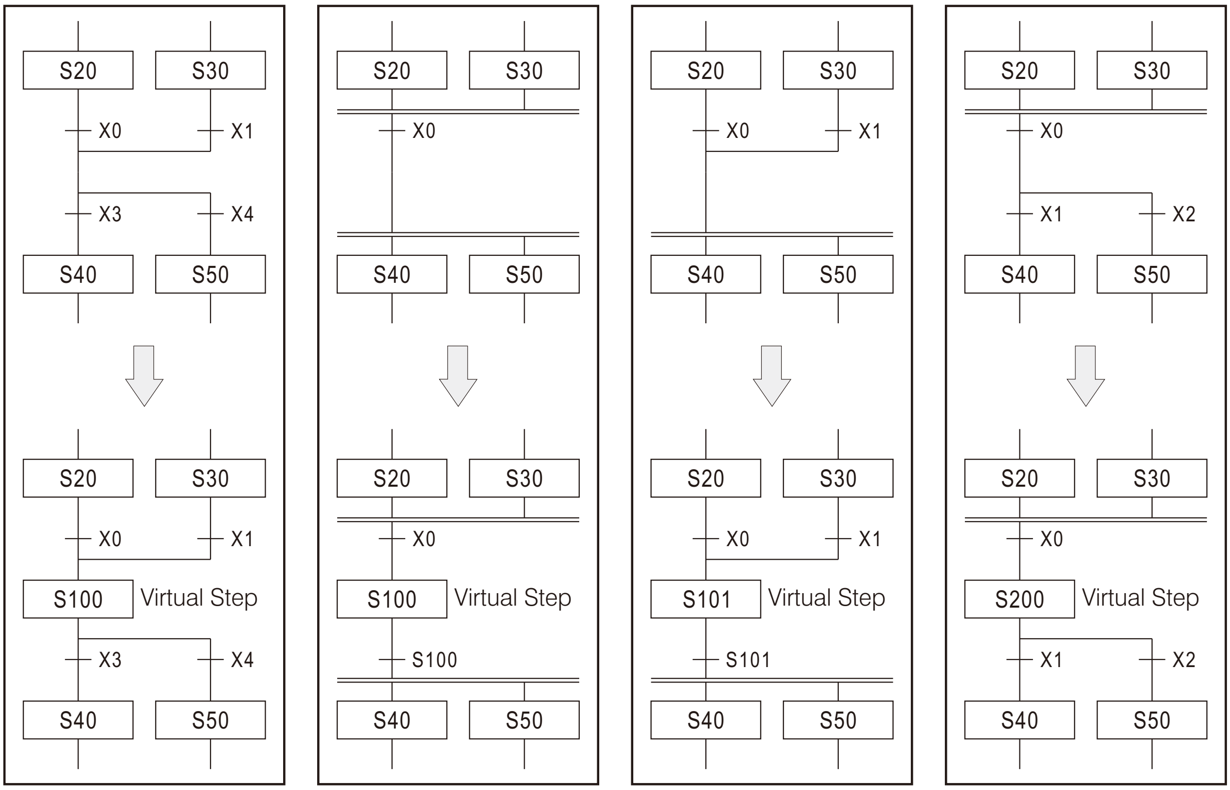


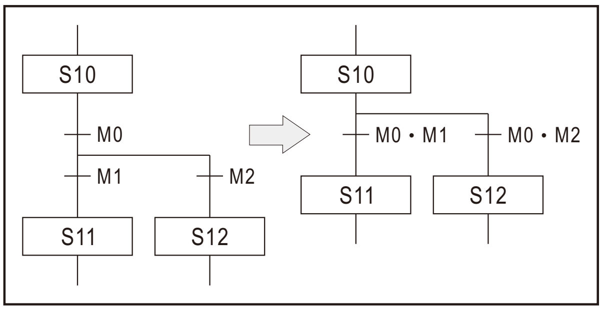
When the step relay S10 is effective and its transfer condition is NO, the following S20, S30, S40, S50, S60, S70, S80 and S90 become active simultaneously. Each flow can do its own process and transfer. After a while, the S22, S32, S42, S52, S62, S72, S82 and S92 are effective and waiting the transfer condition of merging. If all the requirements above are authorized, the effective state will move to the S11.

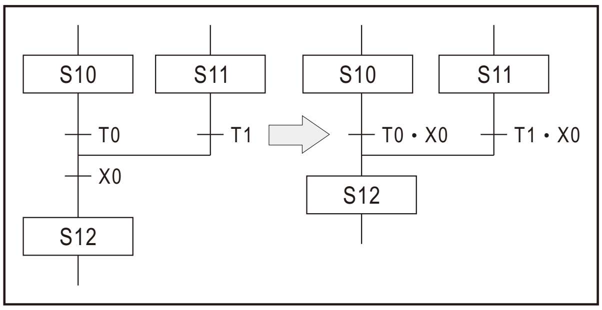
4-3-4 Special Notices About the SFC

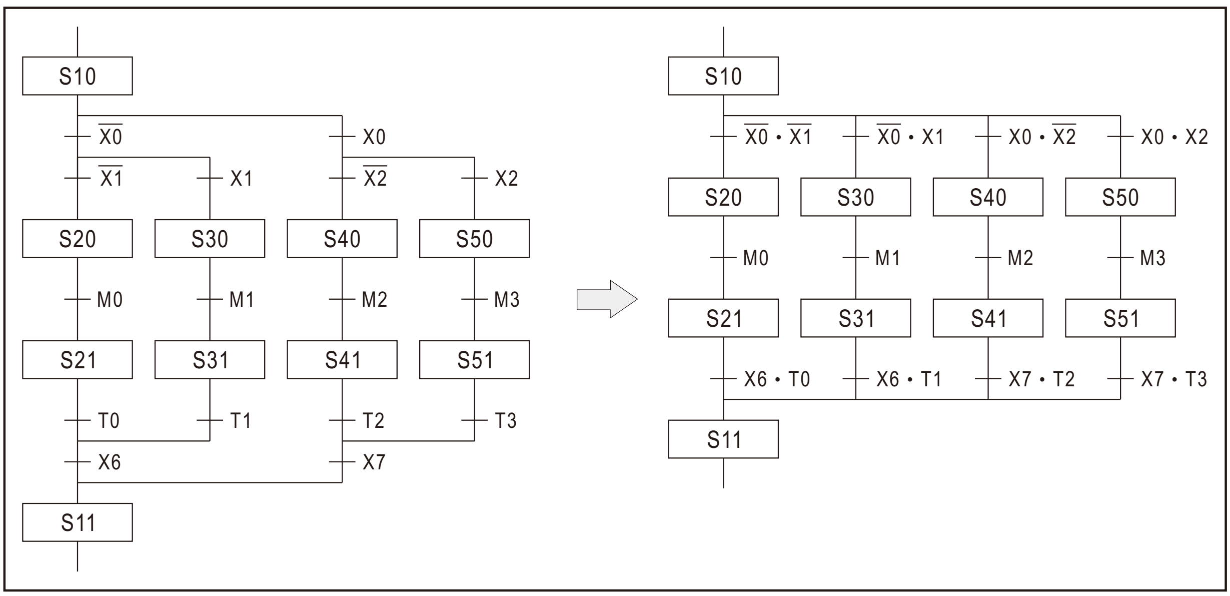
The SFC must use the Ladder Masters S (LMS) to compose and join that into the project, then could be loaded and executed at the PLC. There are some rules about the state transfer. The merge operation must direct to a destination step. Furthermore, any step must bind with one and only one transfer condition. Thus, it is necessary to modify an unqualified SFC to meet the requirement.

In a SFC flow can be added with an empty step. Since the step is for connection only, it is called a Virtual Step.





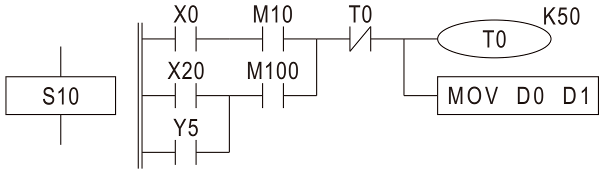




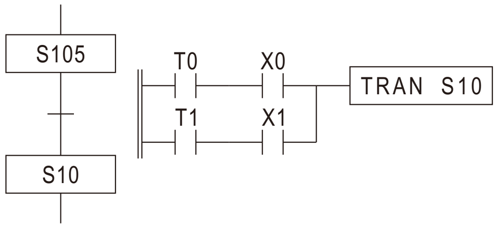
4-4 Precautions About the Composition of the SFC

To create an user program for the VS PLC, the Ladder Master S provides the SFC to combine with the original ladder diagram. Be sure to understand its specifications and restrictions, in order to use it smoothly and complete the control program correctly.

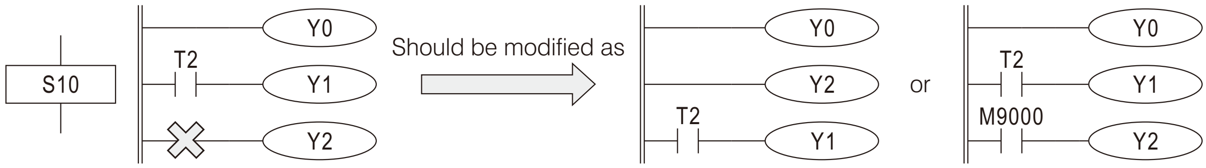
① The control process within a step can use a variety of serial / parallel links, outputs and application instructions.



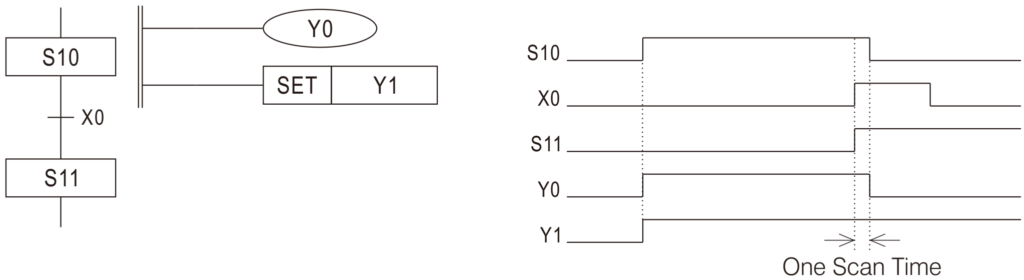
② The transfer condition between two steps can be driven by serial / parallel links.



③ The control outputs in a step may directly drive by the inner bus line. However, once a contact is inserted, the subsequent outputs and application instructions cannot directly drive from the bus line.

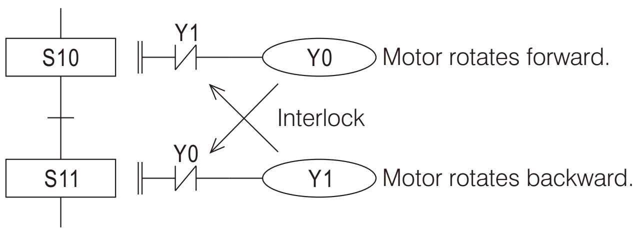


④ When using the OUT instruction to drive a coil in the step, the coil will turn OFF after the effective state transferred. When using the SET instruction to drive a coil ON in the step, the coil will retain ON after the effective state transferred.



⑤ When the effective state is transferring between two sequential steps, both of the states are ON at this scan time. Please refer to the above figure.

⑥ Since both the states of two sequential steps will ON at the scan time of transferring. It is necessary to add the output interlock circuits.

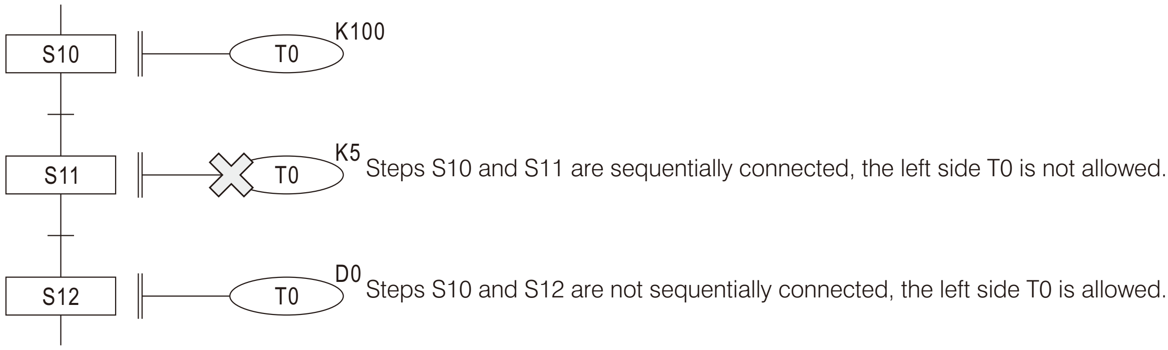


⑦ A SFC cannot be written in a subroutine.

⑧ Although the use of the jump (CJ) instruction is not prohibit inside of a SFC step, it makes the program process more complicated and should be avoided.

⑨ When there is a SFC in the PLC program, do not use the SET or OUT instruction in the interrupt subroutine to drive the step relay.

⑩ Two non sequential steps can share a timer with the same ID number. Also, it can be set with different Set Values respectively.



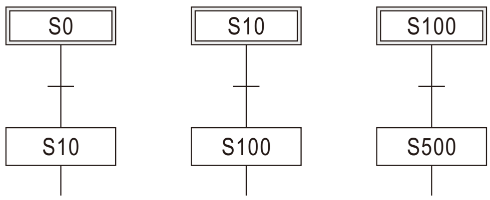
⑪ The MC and MCR instructions cannot be used in the SFC.

⑫ The SET, RST and OUT instructions cannot be used to do the state transfer in the SFC.

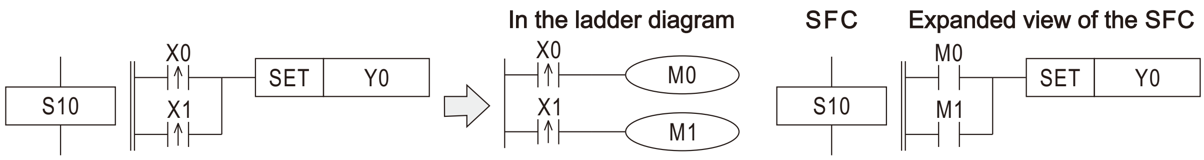
⑬ The same step relay ID number cannot be used repeatedly in the SFC.

⑭ When designing a SFC, the ID numbers of used step relays do not need to follow through a specific sequence.

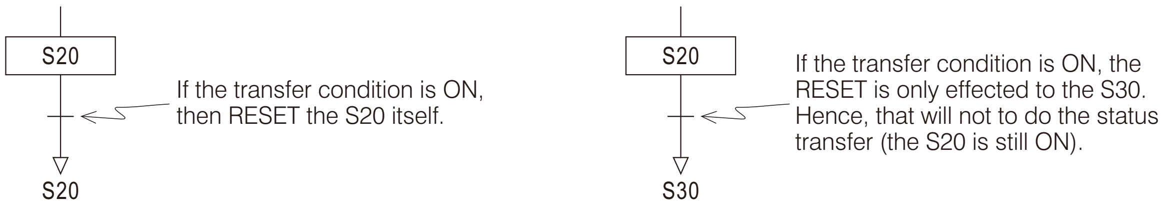
⑮ In the SFC, the S0~S9 are the main initial step relays. However, a general step relay can also be used as an initial step relay, at the beginning of a SFC flow.



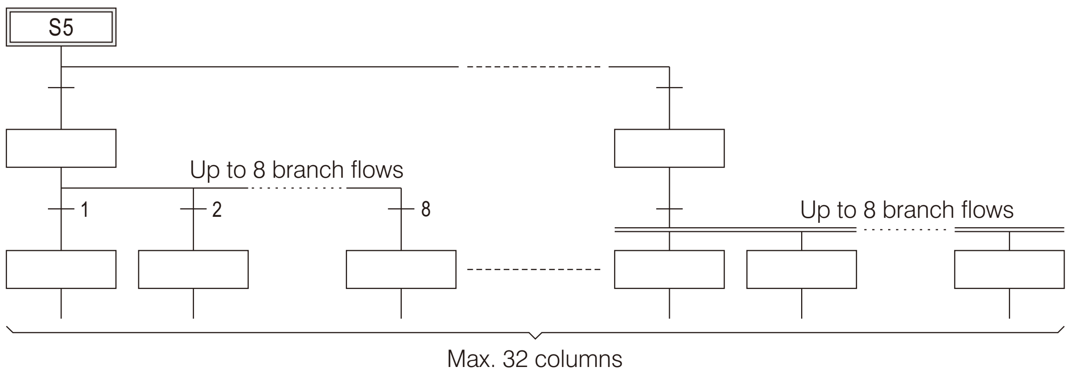
⑯ Avoid using the pulse of rising/falling edge in a SFC. If necessary, can transfer the edge pulse to an auxiliary relay M in the ladder diagram first.



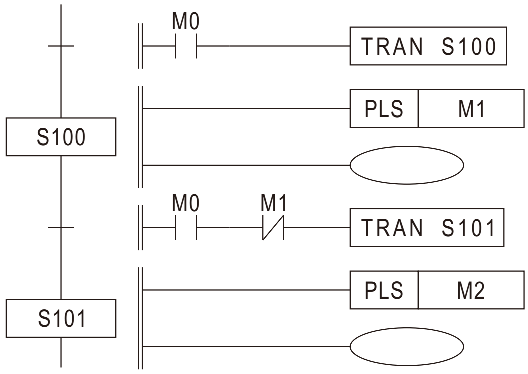
⑰ In the SFC, the sign “ ▽ ” represents the RESET instruction.



⑱ For a selective or simultaneously parallel branch, the maximum limit is 8 branch flows.
If using the Ladder Master S to compose a SFC, one SFC diagram can have at most 32 columns.



⑲ When a signal is repeatedly used to transfer state at different steps, this signal must be a pulse signal. Also, since the effective state transferring between two sequential steps will cause both of the states ON for a scan time, therefore the prohibitive M1 signal should be added as shown below. That could avoid transferring from S100 to S101 immediately.



⑳ In the example of the following figure, when the state of S100 is changed from OFF to ON, it is expected that the Y1 motor rotates to drive the conveyor to move the “Bottle 2” to the position of bottle inspector X1.
When the S100 changes from OFF to ON and the X1 has been turned ON already by the “Bottle 1”, at the Fig. 1 that will cause the state of S100 transfers to S101 immediately, thus the Y1 turned OFF at once. Therefore, it is necessary to improve the SFC become the Fig. 2 in order to successfully achieve the purpose. This kind of application is common in the automatic control, the user must understand the principle of its movements.

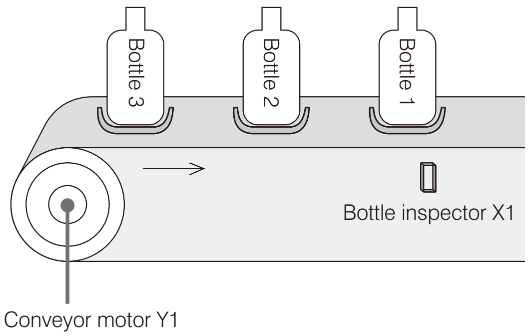


Fig.1

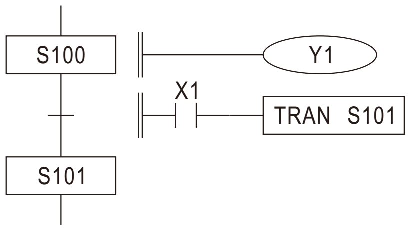
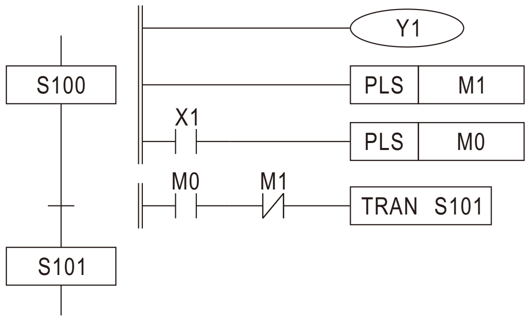


Fig.2



4-5 STL / SFC Relevant Special Components

In the tables below, the symbol “■” represents that the component is not allowed to use an instruction in the program to drive the relay or write data to the register.

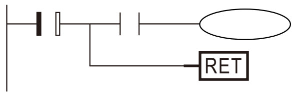
Relay ID No.	Description
M9040	To prevent the step transfer. When M9040=“ON”, the STL state transfer function is disabled.
■M9046	STL step is working. When M9047=“ON” and any relay of S0~S899=“ON” than M9046=“ON”.
M9047	STL monitoring is enable. D9040~D9047 will be active only when M9047=“ON”.

Register ID No.	Description	
■D9040	1st active STL step	When M9047=“ON”, the active STL step ID numbers 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	

4-6 The Relationship Between the SFC and STL

For to programming a step by step movement program, the VS series PLCs provide SFC and STL to choose from. The relationship between these two methods is described below.

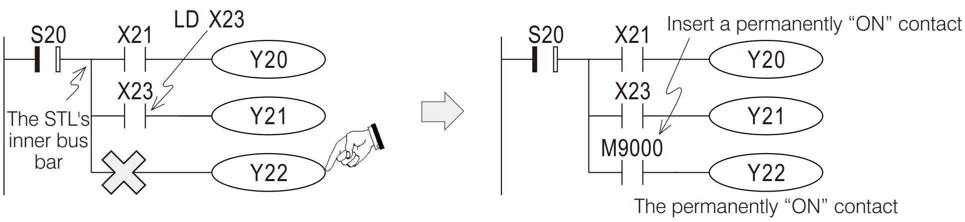
4-6-1 The Step Ladder Instruction (STL)

Mnemonic	Staircase chart	Devices	Function
STL		S	Step ladder starts
RET		—	Return to standard ladder, end of the step ladder

A series of steps is composed by the organized STL instructions and devices S. When an STL instruction appears in the program and the PLC scans to it, that means to change the original sequential procedure into the Step Flow control by the Step Ladder. While the RET instruction is scanned, that indicates the end of this Step Ladder, subsequently the vertical bus bar is reset to an ordinary initial logical operation in the Ladder Diagram. However, after the framework of the Step Flow control is completed, it should be converted into a Step Ladder, and the following important points should be noted during the conversion:

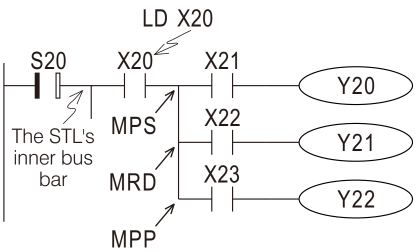
(1) Output Driving Method

As in the left diagram referred below. If inside the Step Ladder has an LD, LDI, LDP or LDF instruction, an output coil can not directly connected from the inner bus bar of the STL after one of those instructions had been used.
Please change the left diagram into the right one.



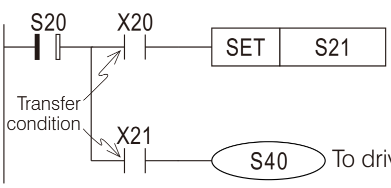
(2) Location of Instructions MPS, MRD and MPP

The MPS, MRD and MPP instructions can not be directly used for the STL's inner bus bar, unless an LD, LDI, LDP or LDF instruction has been used previously.



(3) Transferring Method of STL

As in the diagram referred below, these two instructions SET S21 and OUT S40 are to drive and transfer to another step. When the active state is transferred to another step, the previous step itself will be reset to "OFF" automatically.
The difference is that the SET instruction is used to drive an immediately following STL step, but the OUT instruction is used for to distant jump and drive a step which is not immediately following or at the separate STL program flow.



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

To drive a step which is not immediately following or at the separate STL program flow.

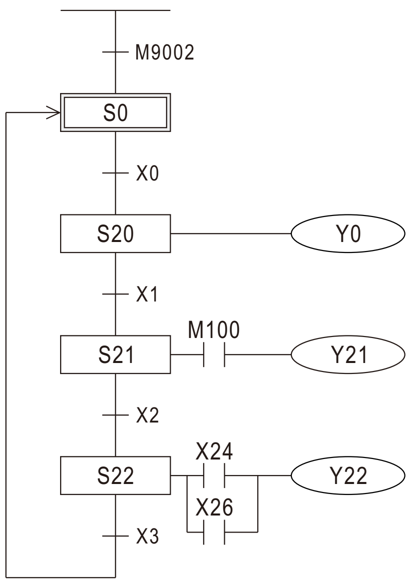
(4) Function of Instruction RET

Since the RET instruction represents the end of a Step Flow the RET instruction will appear eventually after a series of steps. A program may be written many Step Flows each Step Flow should put an instruction RET at the end. The instruction RET can be used as many times as required.

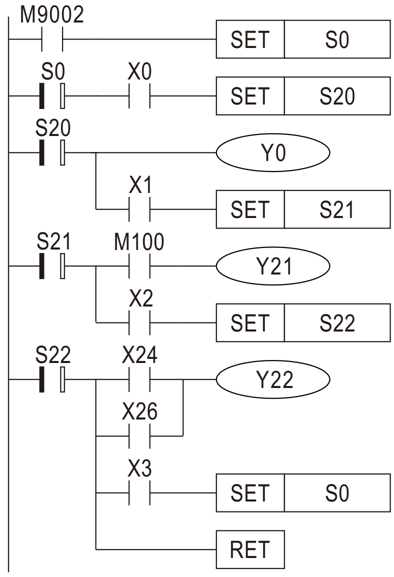
4-6-2 Compare the Descriptive Methods Between the SFC and STL

- Simple Flow at the SFC and STL

(a)SFC



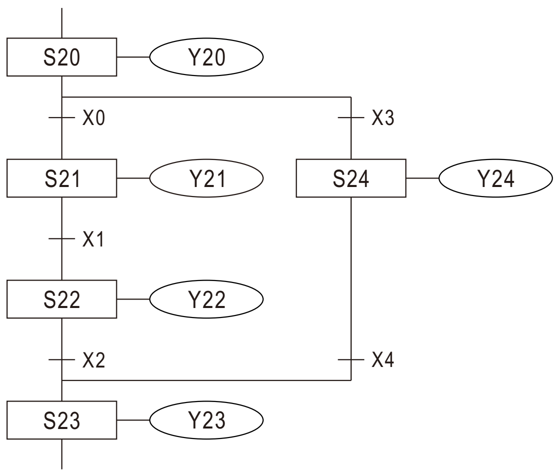
(b)STL



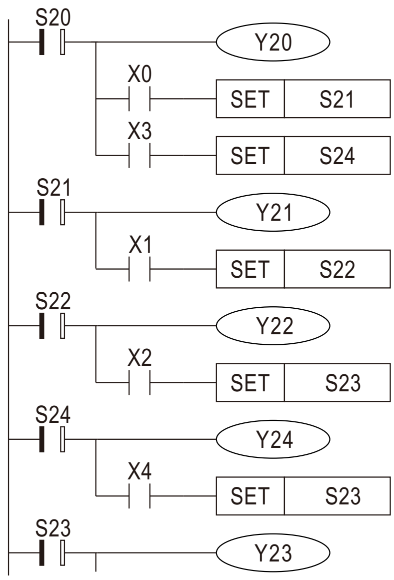
In diagram (a) SFC, each step has three functions and parts: to drive the output points for loaders to appoint transfer destination devices and to assign the transition conditions. Such the left SFC by the format of STL is displayed as in diagram (b), in which we adopt as the symbol for use of STL instructions. And those STL instructions are provided with the state transfer and auto reset functions.

- Selective Branch / Merge at the SFC and STL

(a)SFC

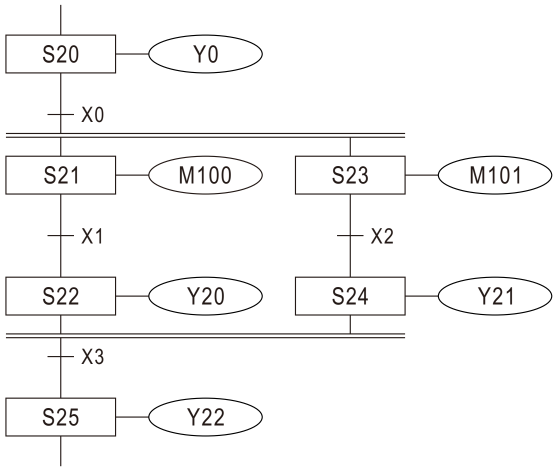


(b)STL

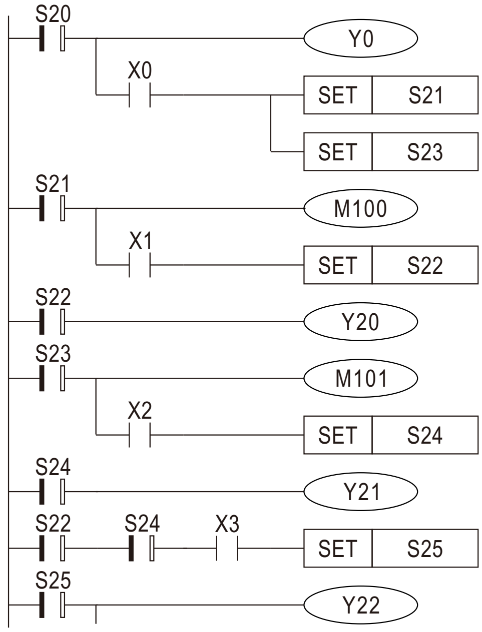


● Simultaneously Parallel Branch / Merge at the SFC and STL

(a)SFC

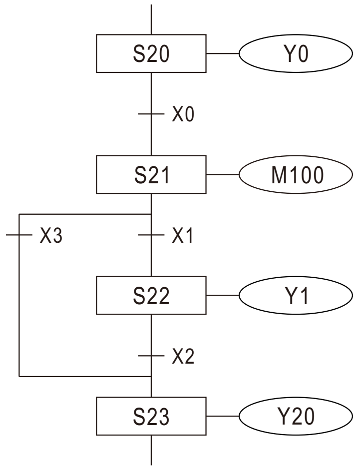


(b)STL

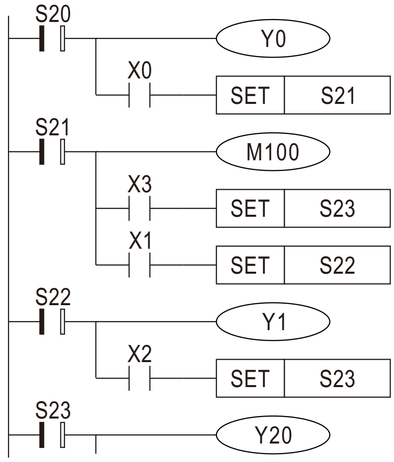


● Jump at the SFC and STL

(a)SFC

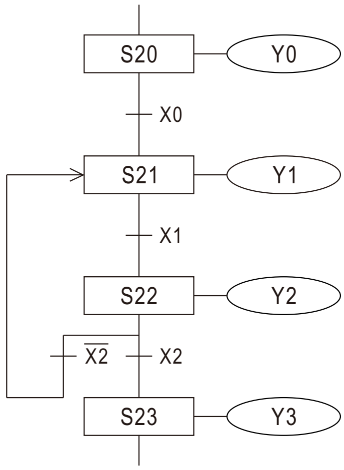


(b)STL

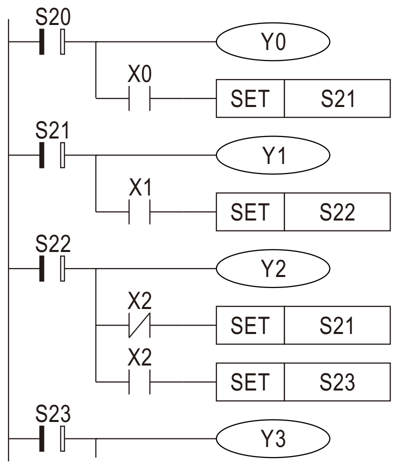


● Repeat at the SFC and STL

(a)SFC



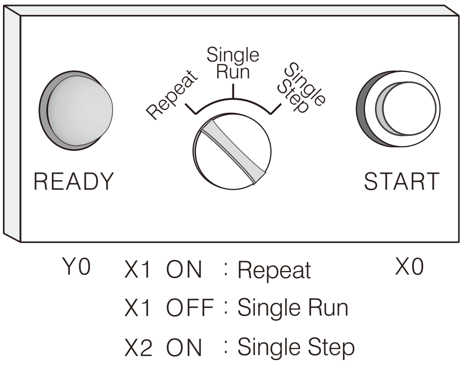
(b)STL



4-7 Examples of SFC Applications

4-7-1 Construct the Repeat / Single Run / Single Step Control Modes for a Park Fountain

This example is modified from the example of the Park Fountain in the section 4-2. It is to demonstrate the controls of Repeat / Single Run / Single Step.

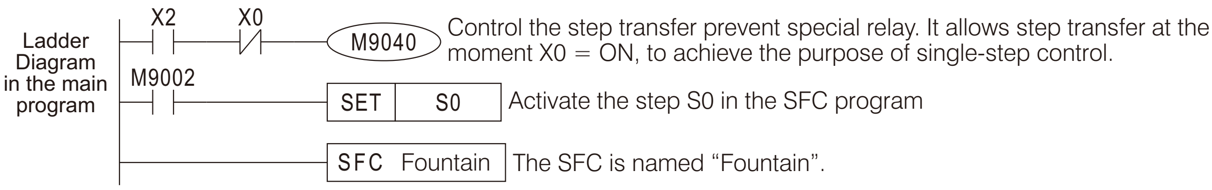


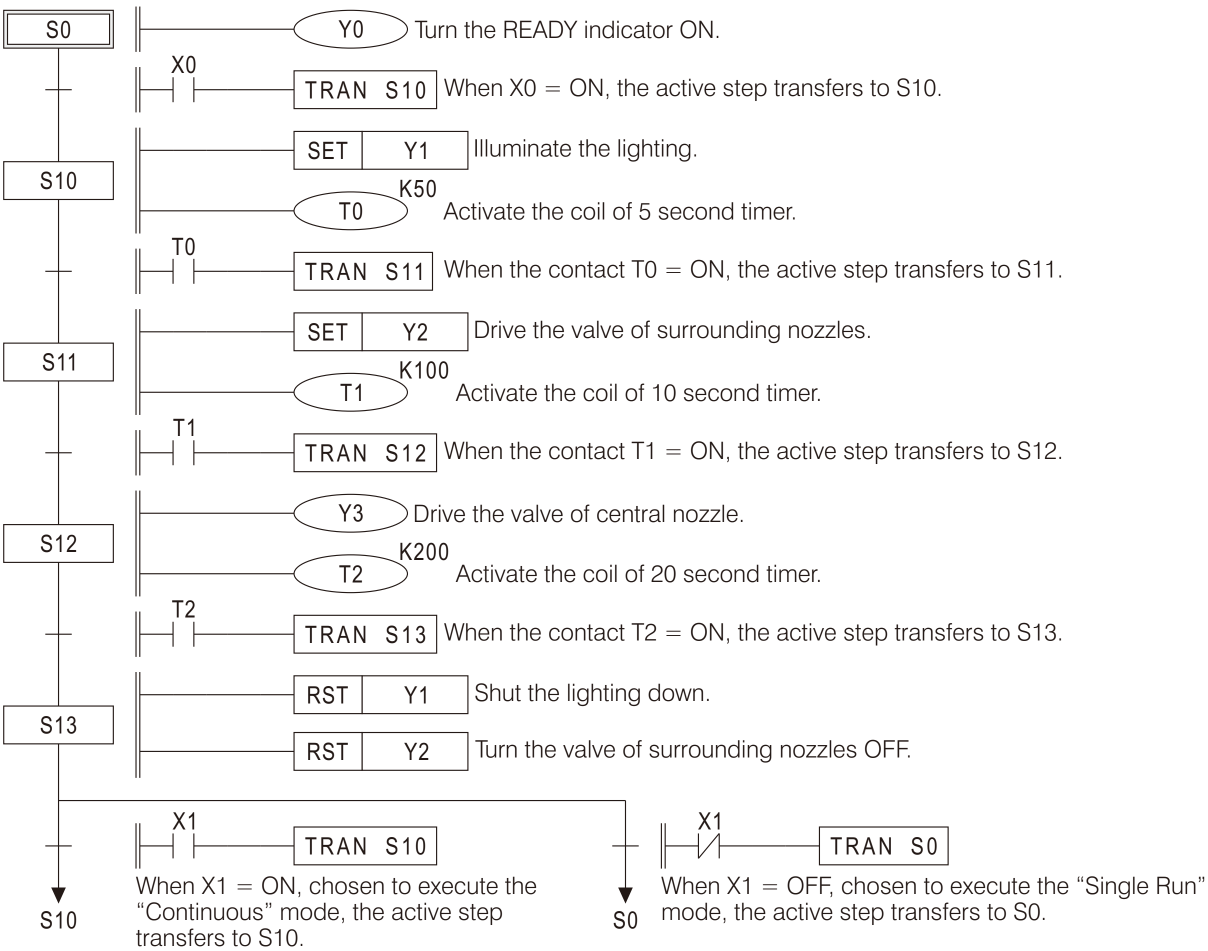
Y0: READY indicator
Y1: The lighting for the fountain
Y2: The valve of surrounding nozzles
Y3: The valve of central nozzle

When the "Repeat" mode (X1 = ON, X2 = OFF) is selected, press the START button to execute the operation repeatedly. When the "Single Run" mode (X1 = OFF, X2 = OFF) is selected, press the START button to execute a cycle of operation. When the "Single Step" mode (X1 = OFF, X2 = ON) is selected, press the START button once can make the action to forward one step.

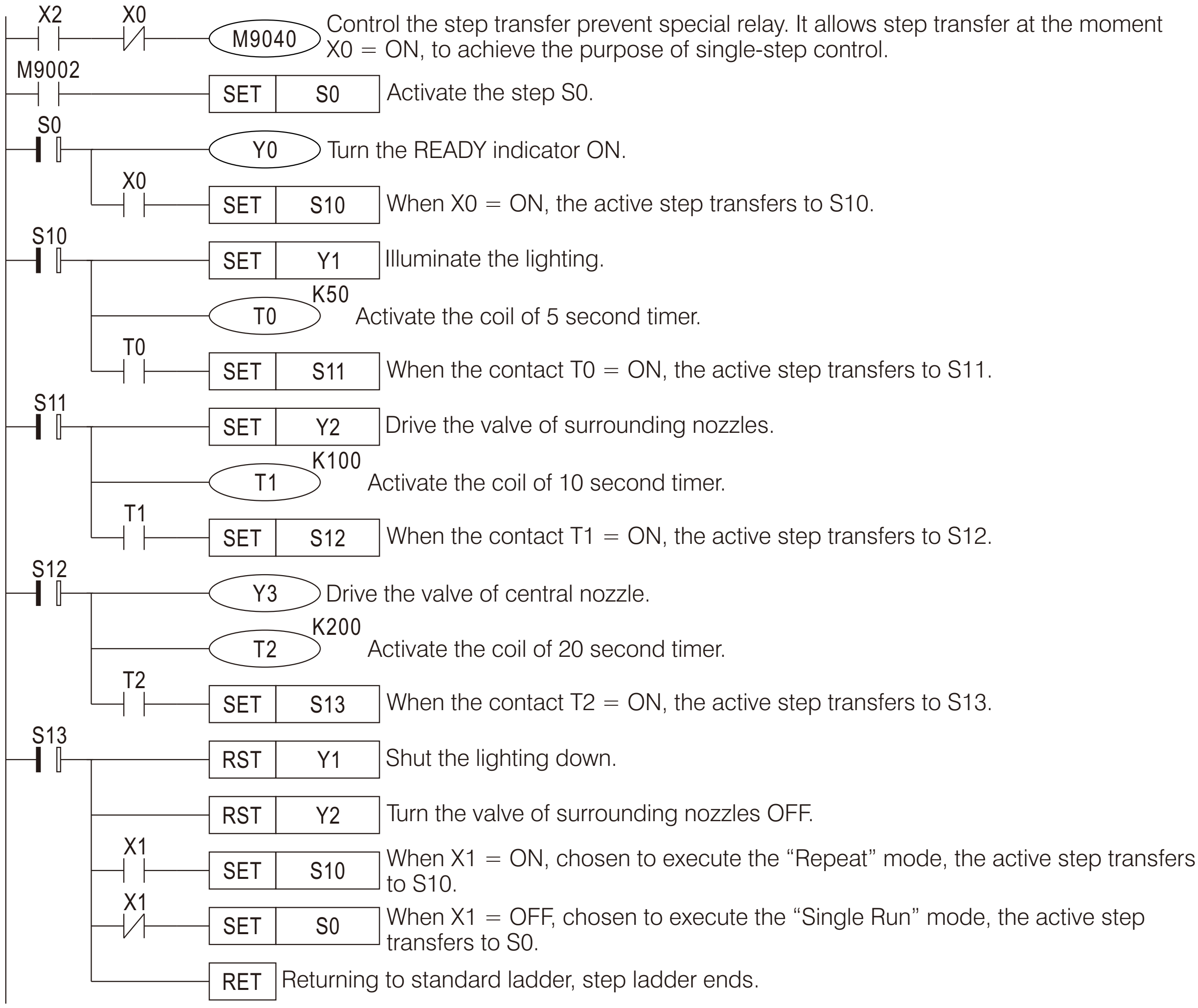
This program uses the special relay M9040 (flag to prevent the step transfer) to meet the purpose of single-step control. Users can make good use of this method after understanding it. The two drawings below display using the (a) SFC or (b) STL in writing the program, in order to comparison reference.

(a) Programming by the SFC

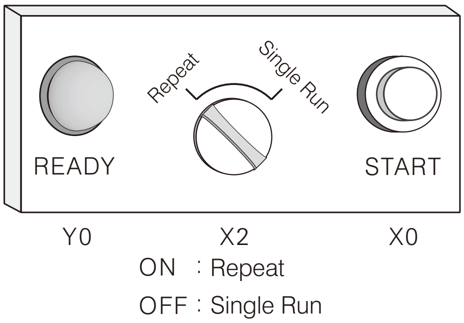


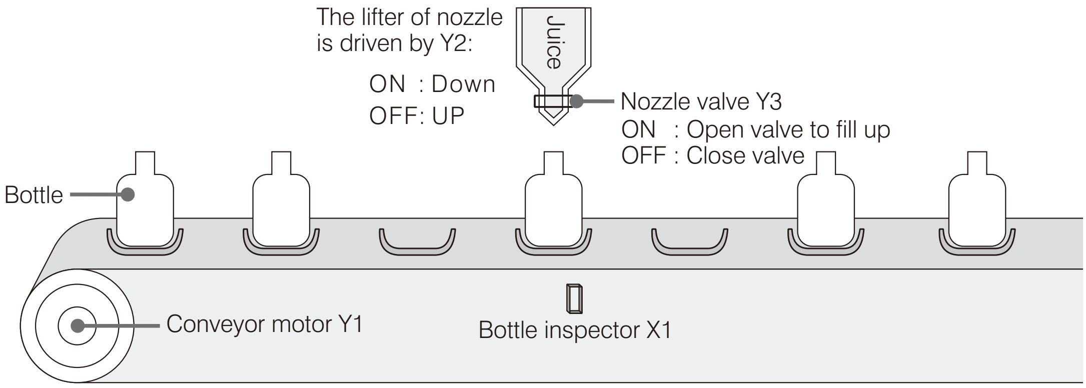


(b) Programming by the STL



4-7-2 Filling Bottles





Function Description of Bottle Filling:

There is a control panel next to the bottle filling machine. The READY indicator on the front of panel that indicates at the standby status. Use the START button to activate the device and the mode selector switch to choose from a single run or repeated operation.

(1) When the machine is ready to start operation, the READY indicator is ON.

(2) Press the START button and the conveyor motor starts to rotate. When the bottle inspector detects that the bottle has reached the right position, the motor stops rotating.

(3) When the nozzle lifter is ON, the nozzle going down. It is determined that in 3 seconds the filling nozzle will reach the position.

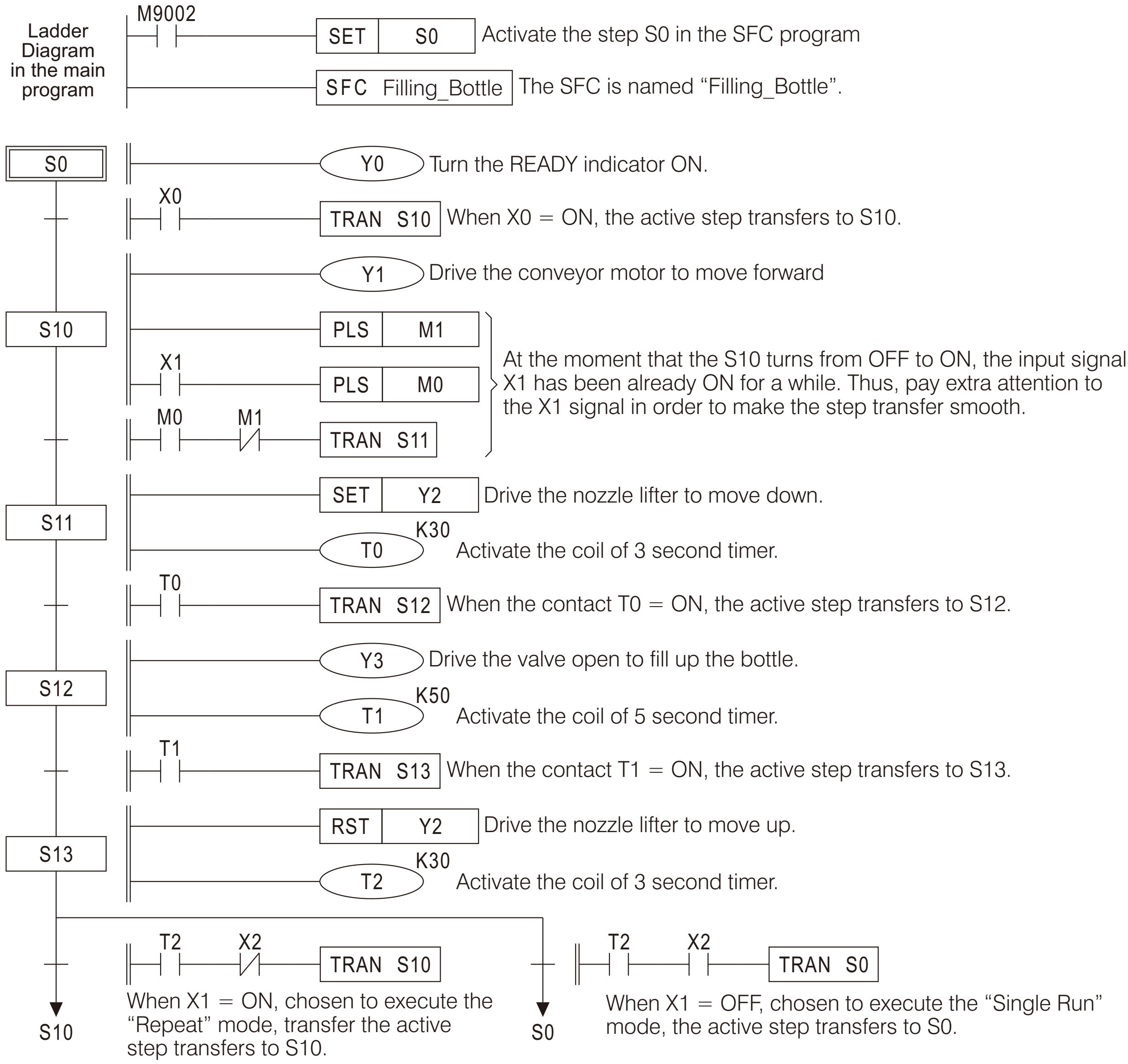
(4) The injection valve opens and starts filling. In 5 seconds, the filling will be completed and the valve will be closed.

(5) When the valve of the filling nozzle is OFF, the filling nozzle rises. It is determined that in 3 seconds the filling nozzle will rise to reach the position.

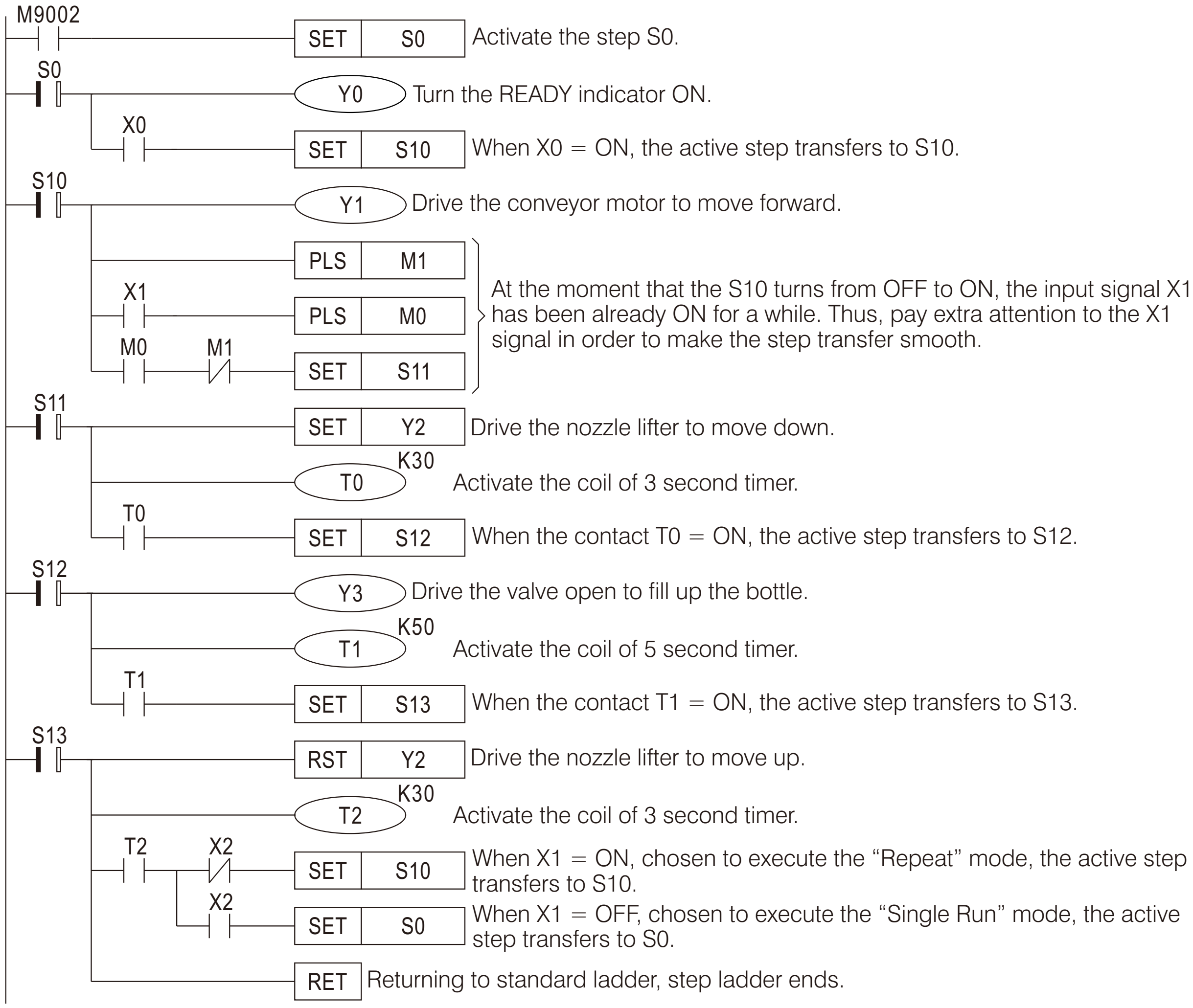
(6) The filling process is completed. At this point, it returns to the state of standby if at the “Single Run” mode; it starts the conveyor motor again and processes the next bottle fill up if at the “Repeat” mode.

The two drawings below display using the (a) SFC or (b) STL in writing the program, in order to comparison reference.

(a) Programming by the SFC



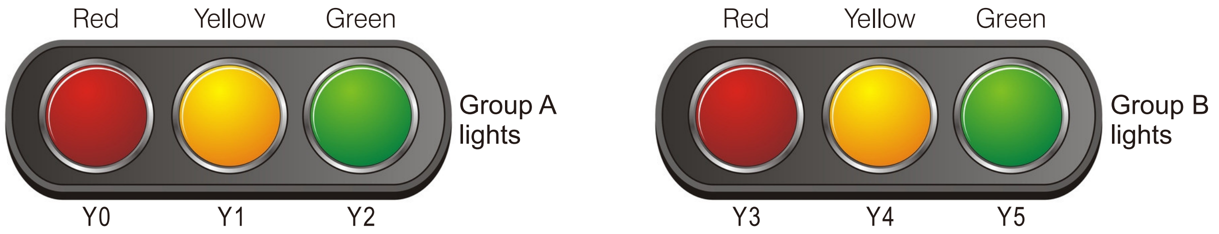
(b) Programming by the STL



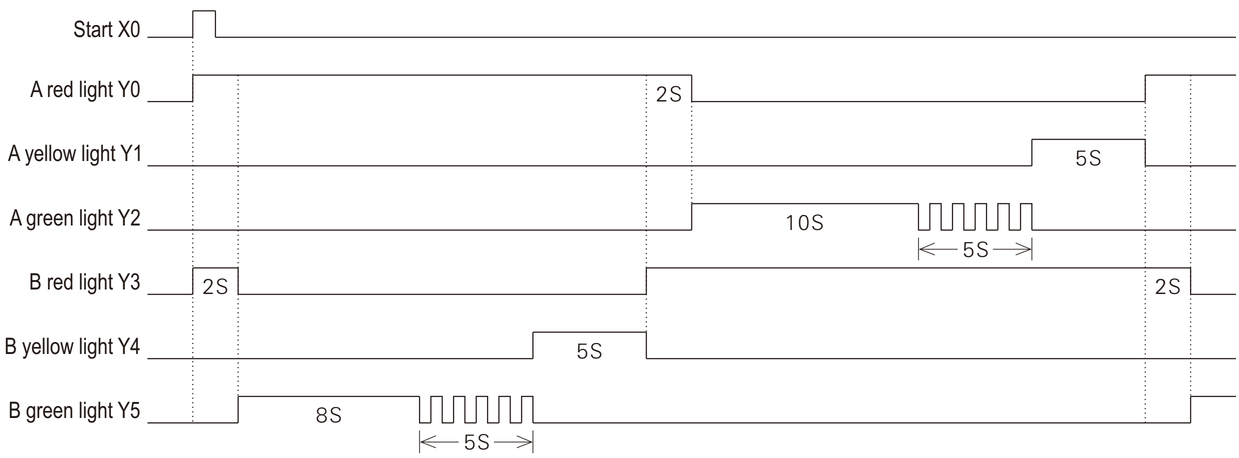
4-7-3 Traffic Lights

In this section, we choose the familiar traffic lights, as an example for the description of step-by-step movement.

There are two sets of traffic lights at the crossroads, hereby referred to as Group A lights and Group B lights as shown below. When the start button X0 is pressed, the sequential control of traffic light is activated. First, the A red and B red lights are lit. After 2 seconds, B red light goes OFF. Followed by B green light is lit for 8 seconds and flashes for 5 seconds. After that B yellow light is ON for 5 seconds, then shift to the B red ON. In 2 seconds later, Group A lights take over the operation, following the same sequences of movements as Group B lights.



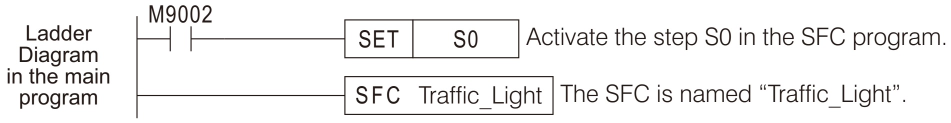
Traffic lights active I/O duration time sheet:

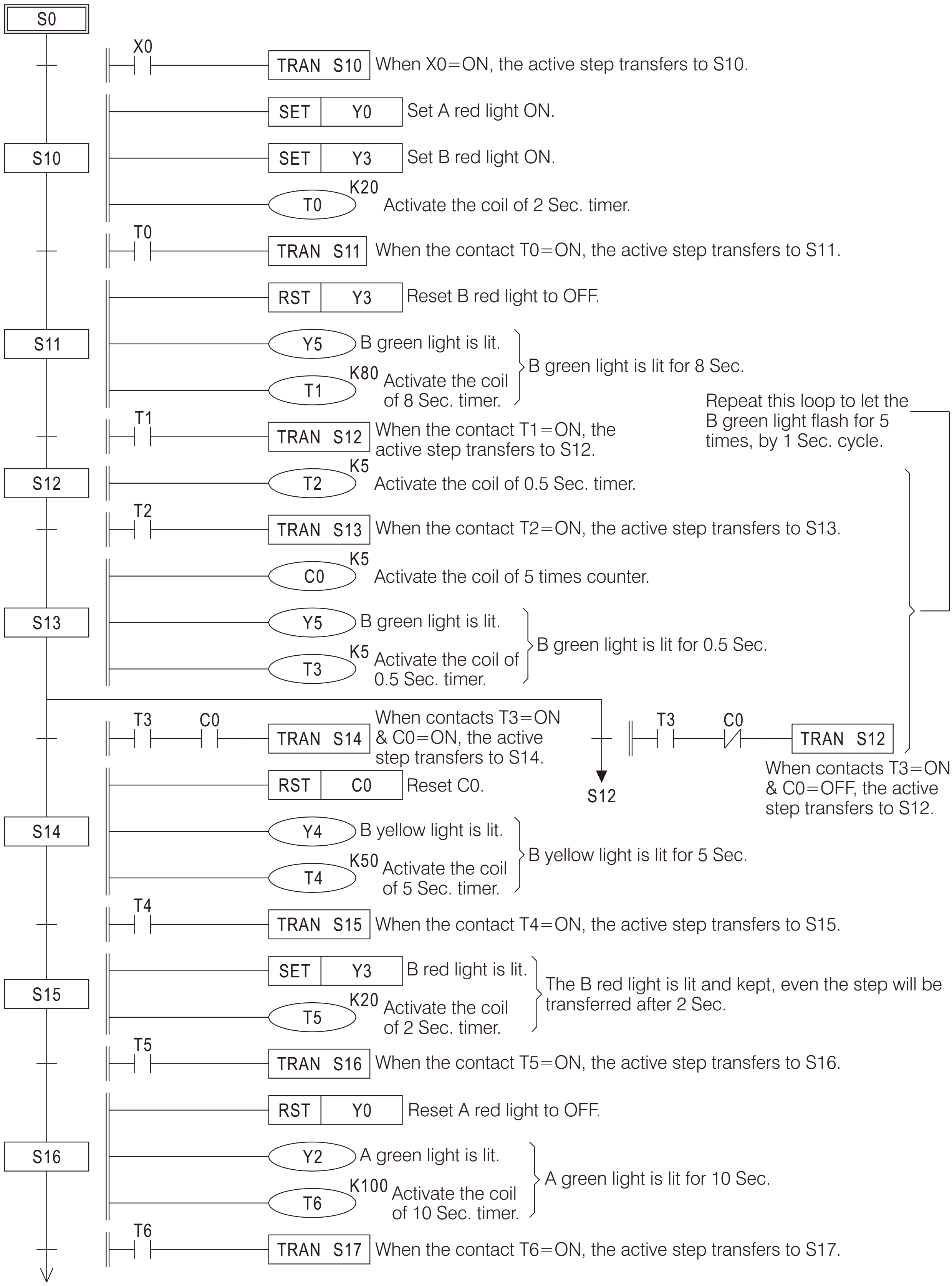


The following demonstrates are compiling by some SFCs and STLs with different programming methods.

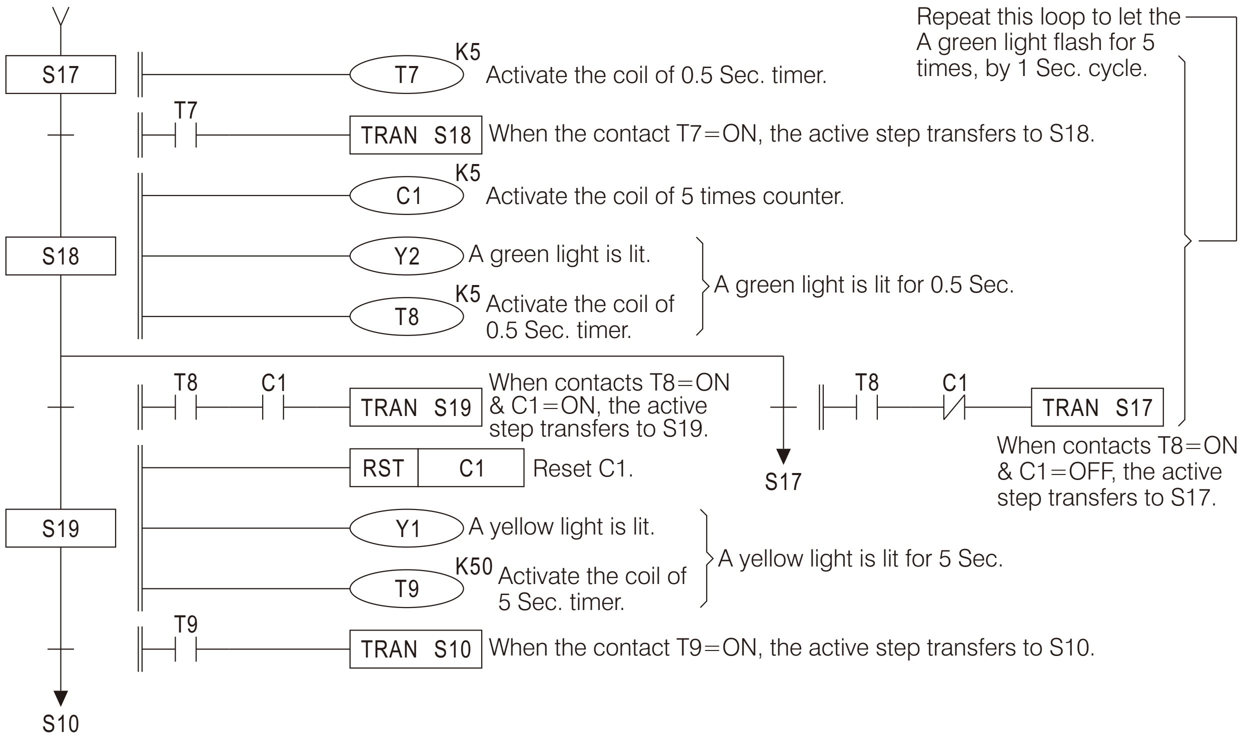
● Below displays simple flow programming, individually using (a) SFC and (b) STL in writing the program, in order to comparison reference.

(a) Programming by the SFC

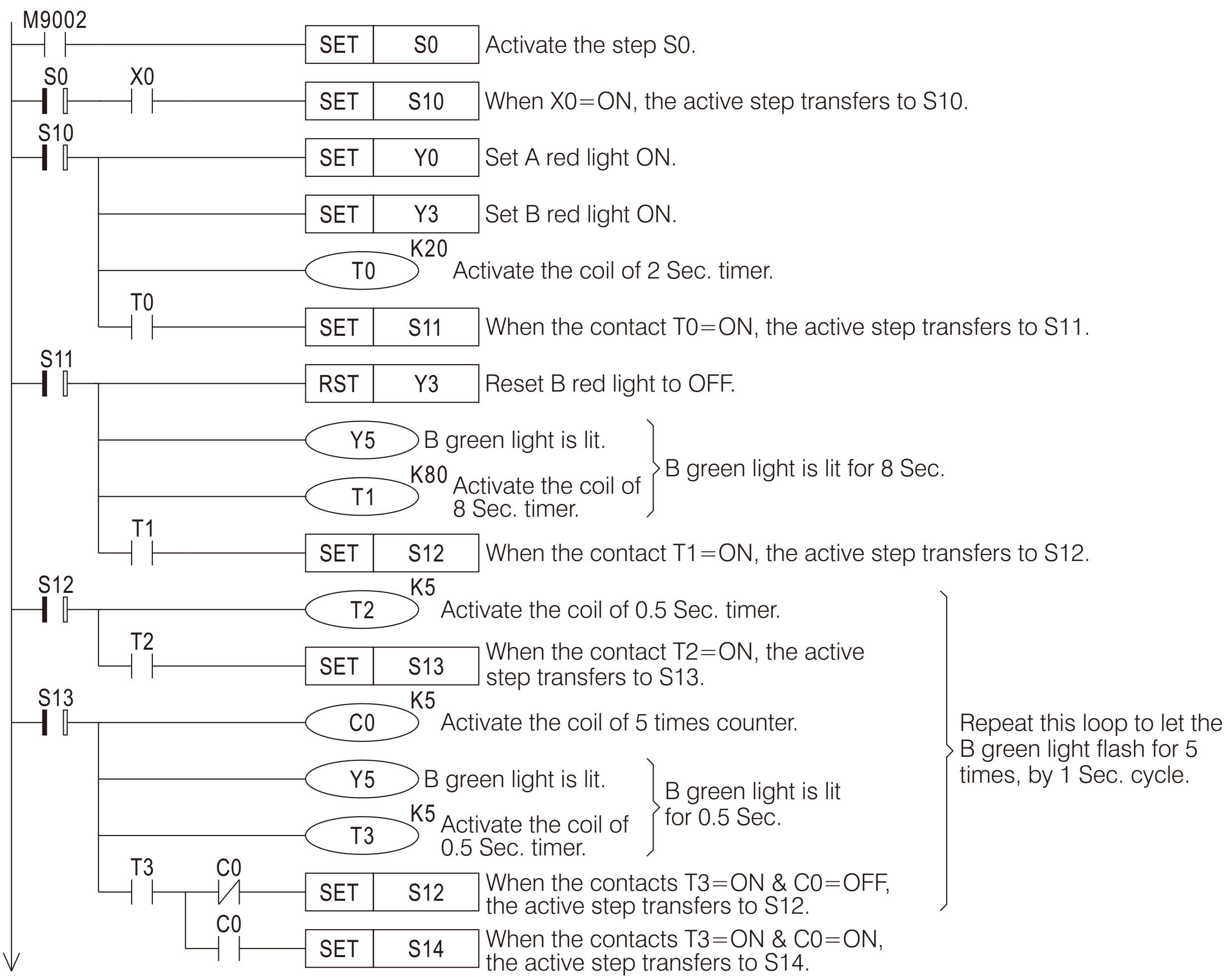




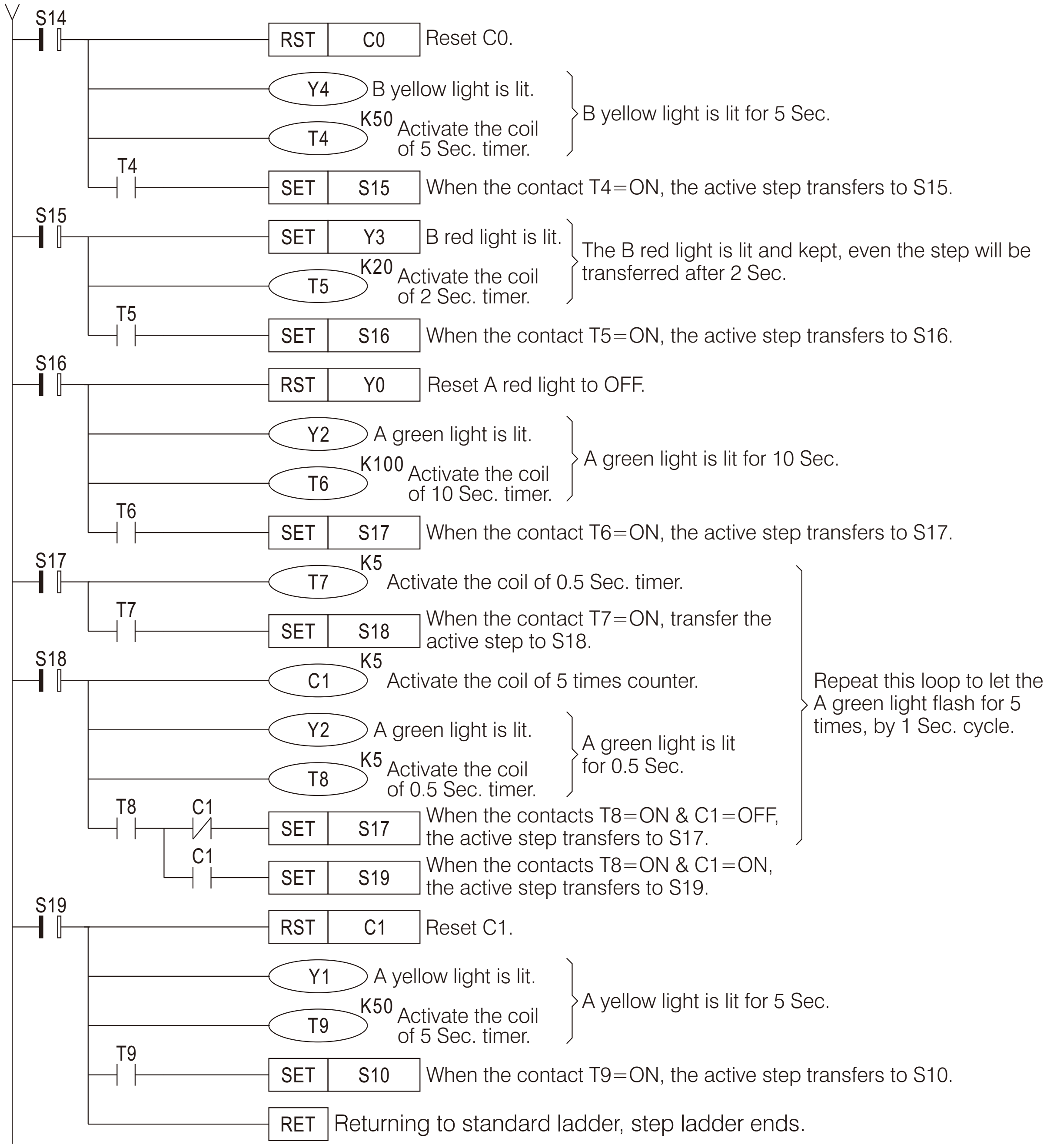
Continue on next page



(b) Programming by the STL

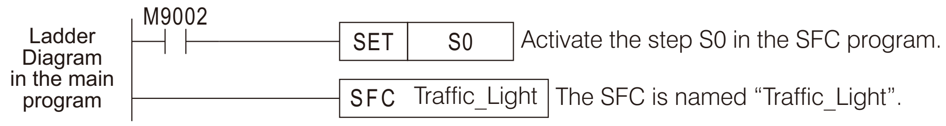


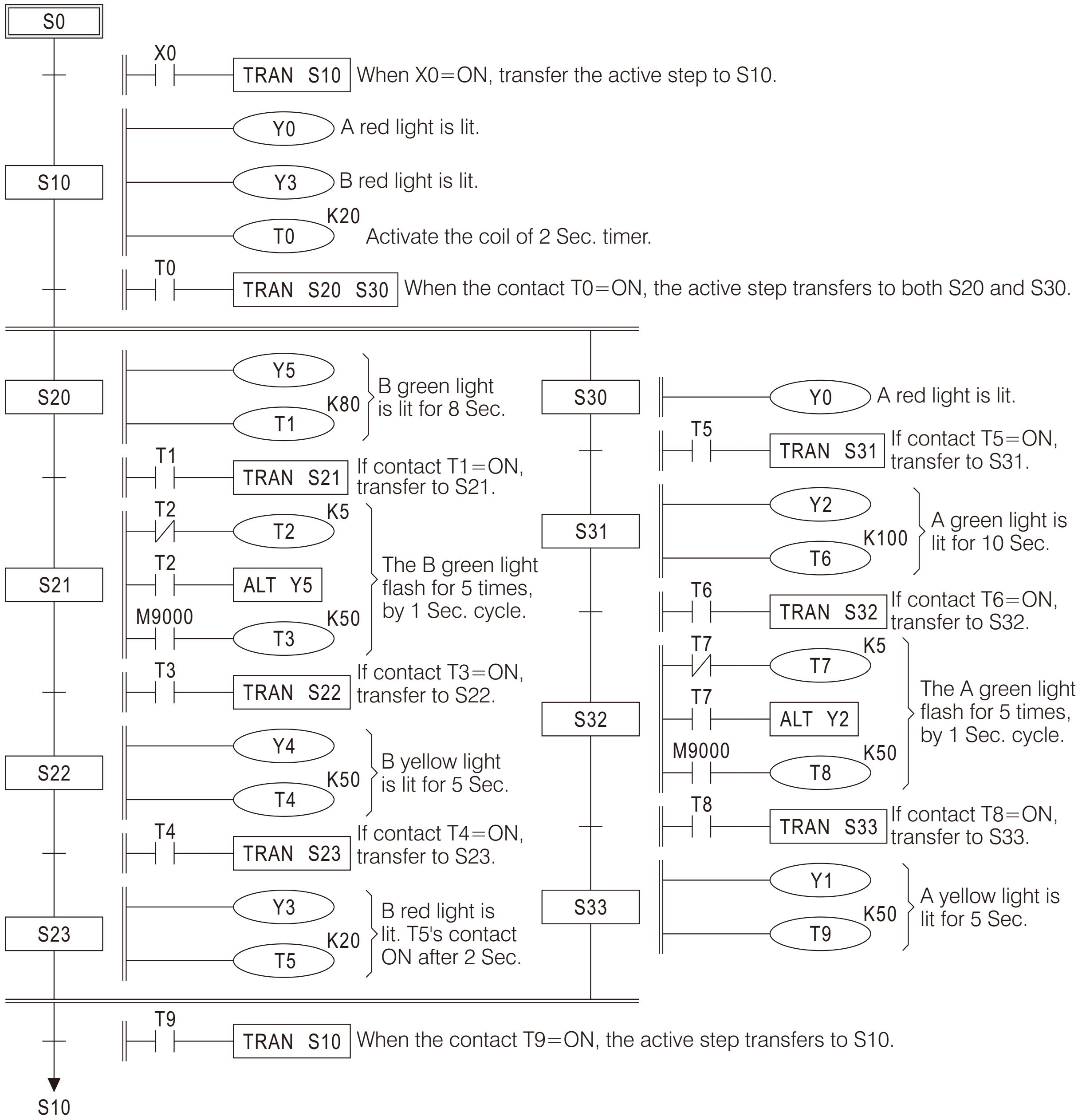
Continue on next page


S14
RST C0
Reset C0.
Y4
B yellow light is lit.
K50
T4
Activate the coil of 5 Sec. timer.
B yellow light is lit for 5 Sec.
T4
SET S15
When the contact T4=ON, the active step transfers to S15.
S15
SET Y3
B red light is lit.
K20
T5
Activate the coil of 2 Sec. timer.
The B red light is lit and kept, even the step will be transferred after 2 Sec.
T5
SET S16
When the contact T5=ON, the active step transfers to S16.
S16
RST Y0
Reset A red light to OFF.
Y2
A green light is lit.
K100
T6
Activate the coil of 10 Sec. timer.
A green light is lit for 10 Sec.
T6
SET S17
When the contact T6=ON, the active step transfers to S17.
S17
K5
T7
Activate the coil of 0.5 Sec. timer.
T7
SET S18
When the contact T7=ON, transfer the active step to S18.
S18
K5
C1
Activate the coil of 5 times counter.
Y2
A green light is lit.
K5
T8
Activate the coil of 0.5 Sec. timer.
A green light is lit for 0.5 Sec.
T8
C1
SET S17
When the contacts T8=ON & C1=OFF, the active step transfers to S17.
C1
SET S19
When the contacts T8=ON & C1=ON, the active step transfers to S19.
Repeat this loop to let the A green light flash for 5 times, by 1 Sec. cycle.
S19
RST C1
Reset C1.
Y1
A yellow light is lit.
K50
T9
Activate the coil of 5 Sec. timer.
A yellow light is lit for 5 Sec.
T9
SET S10
When the contact T9=ON, the active step transfers to S10.
RET
Returning to standard ladder, step ladder ends.

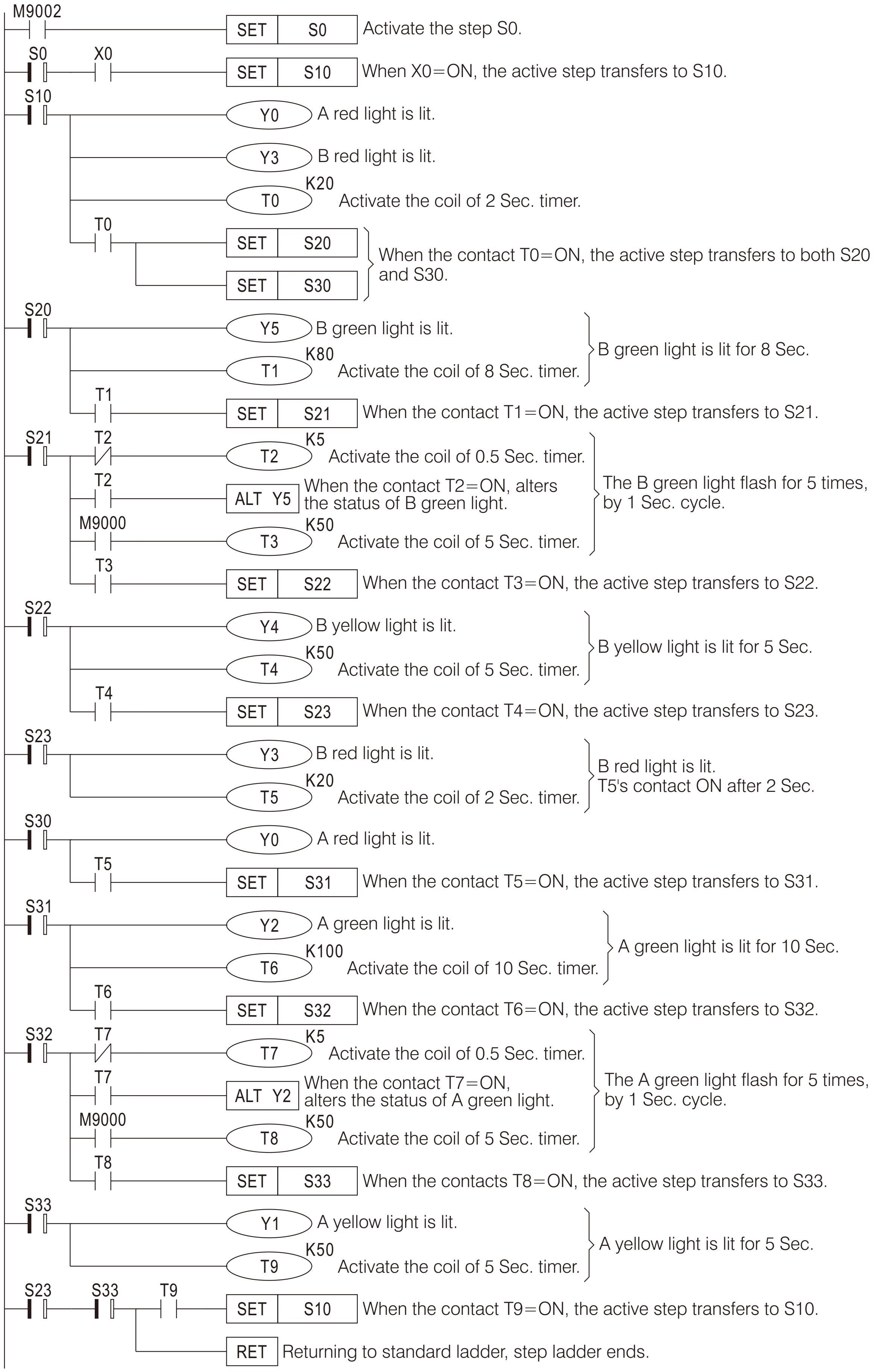
● Below displays Simultaneously Parallel Branch / Merge programming, individually using (a) SFC and (b) STL in writing the program, in order to comparison reference.

(a) Programming by the SFC



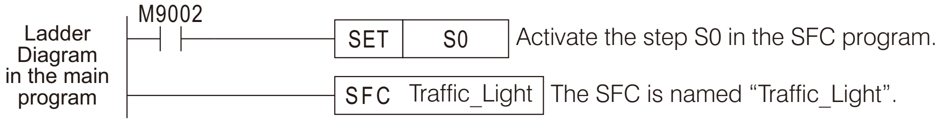


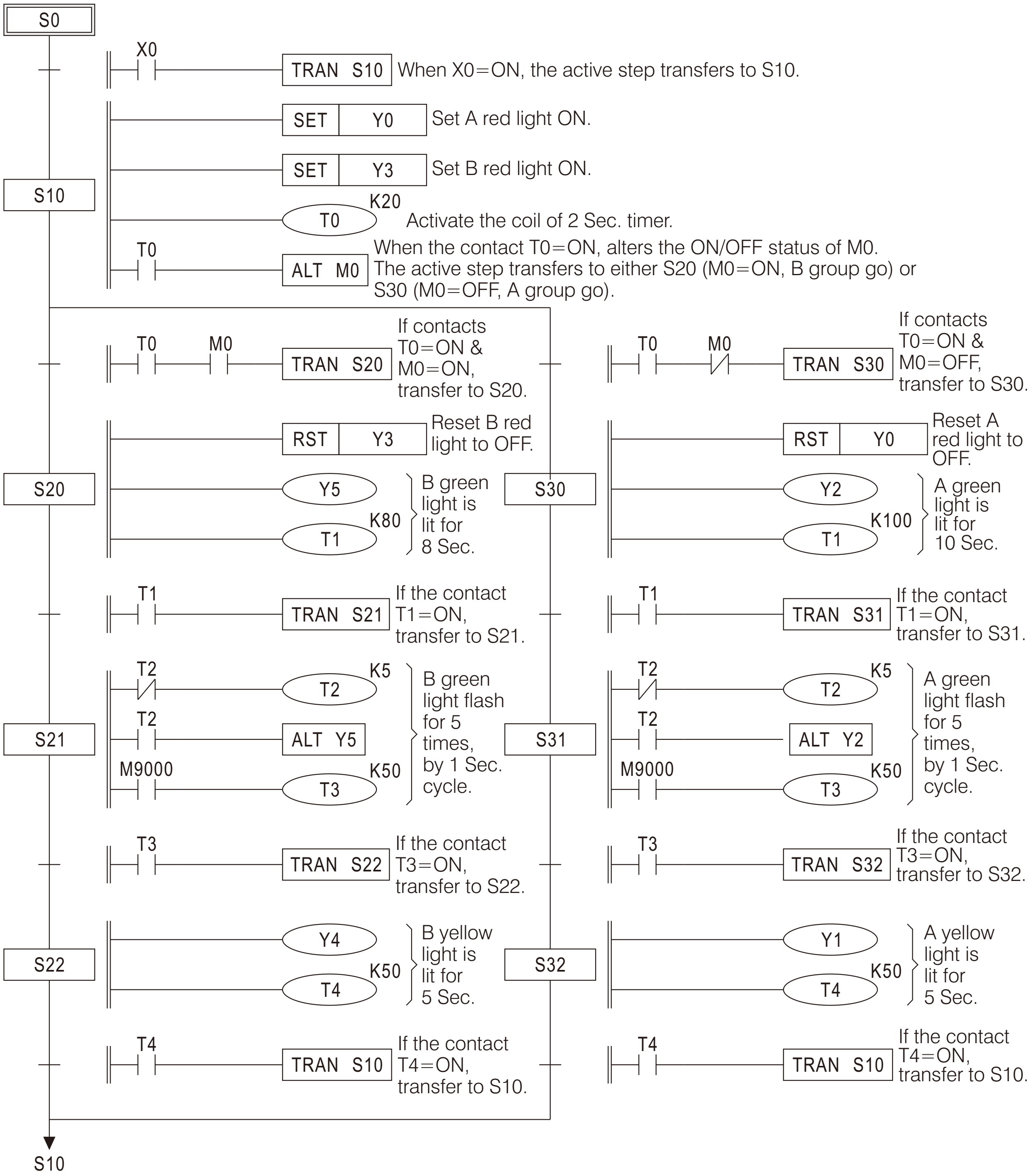
(b) Programming by the STL



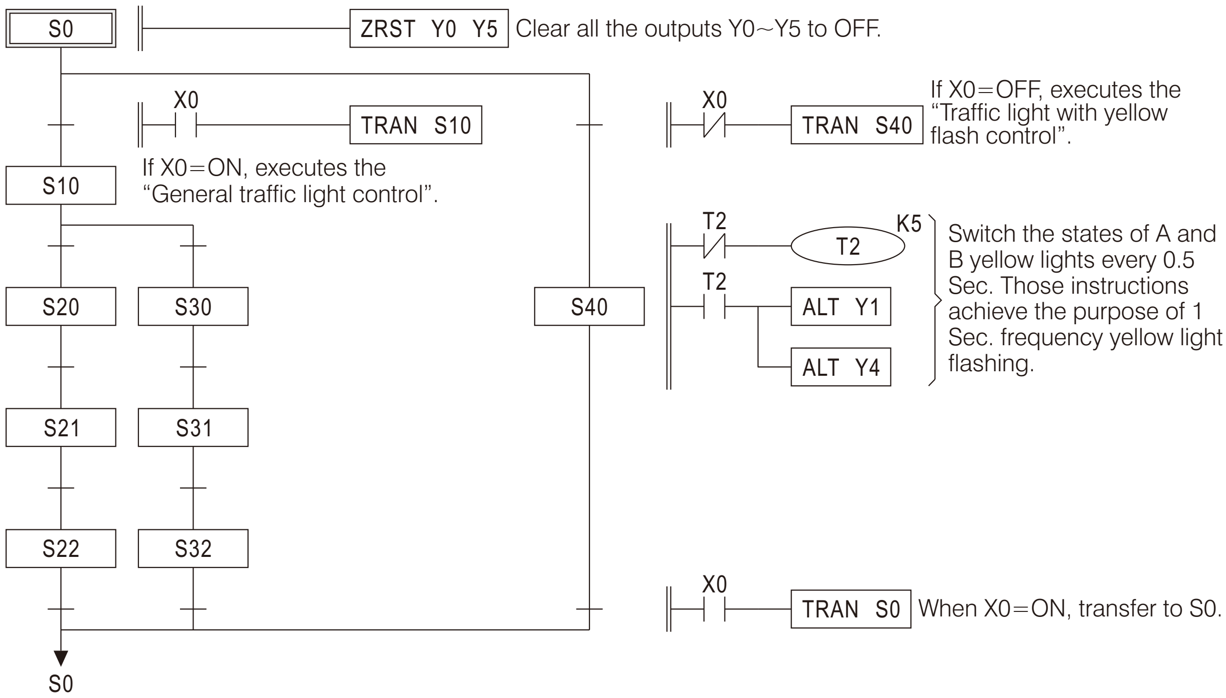
● Below displays Selective Parallel Branch / Merge programming, individually using (a) SFC and (b) STL in writing the program, in order to comparison reference.

(a) Programming by the SFC

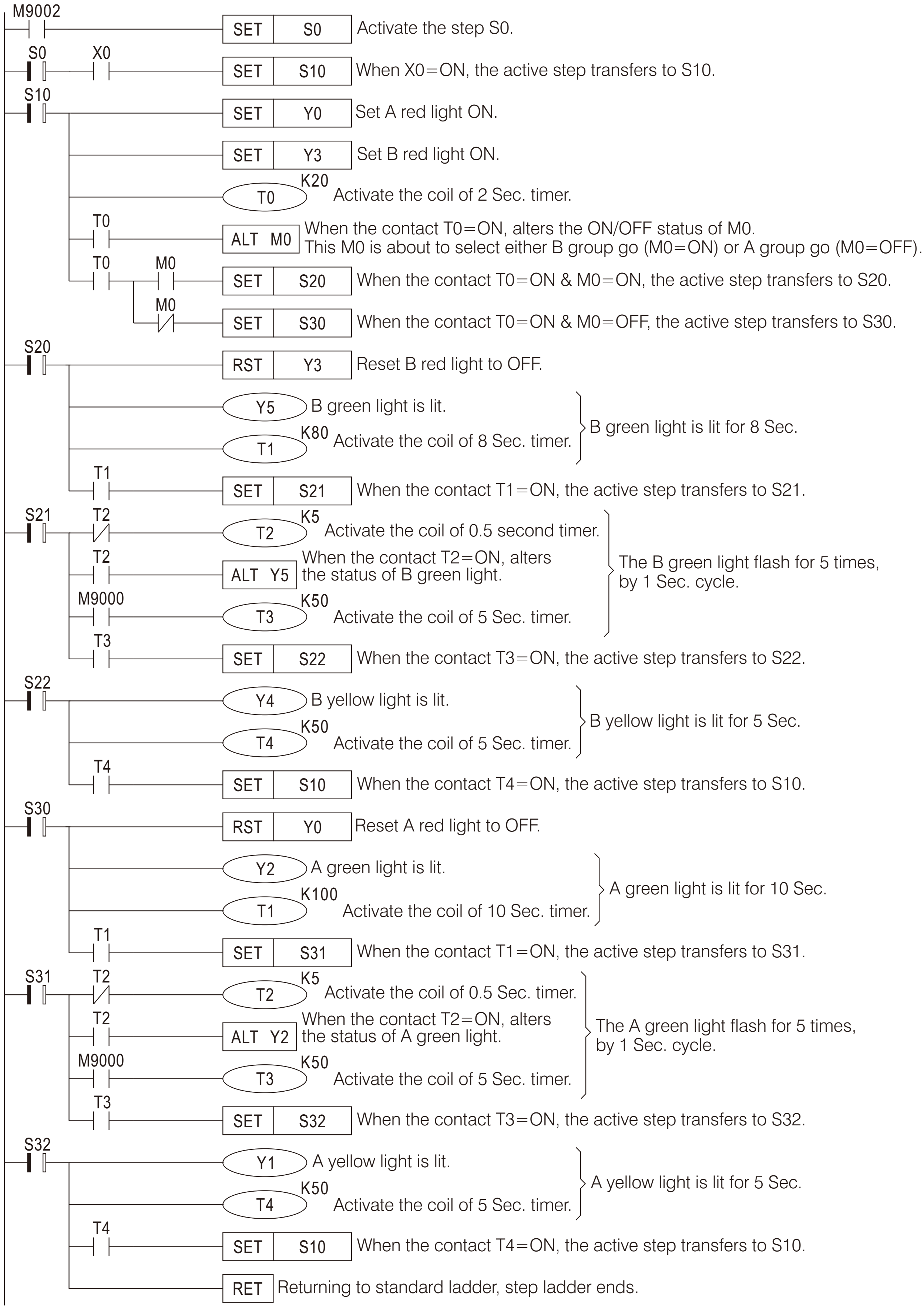




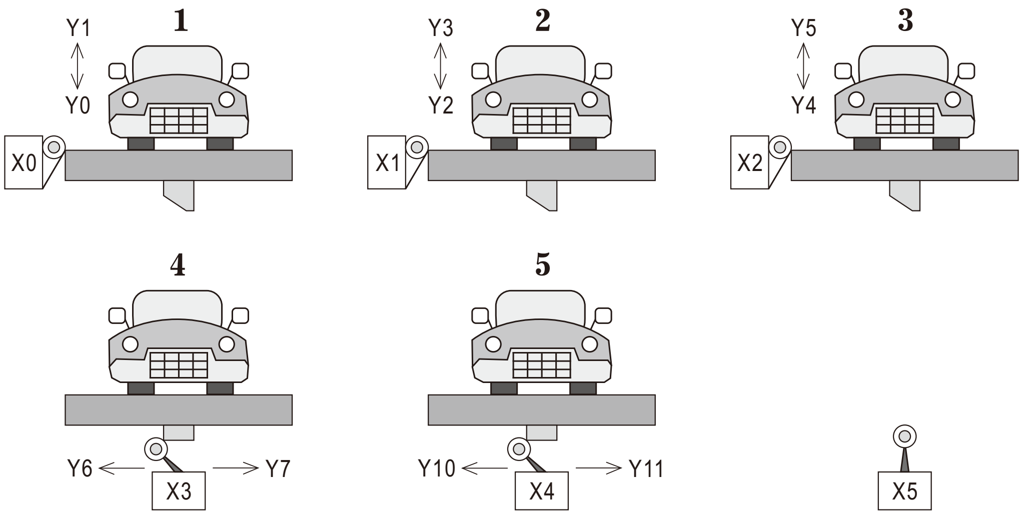
The SFC example above via slightly modify that could execute the “General traffic light control” or “Traffic light with yellow flash control”. The select signal from a switch is to the X0 input.



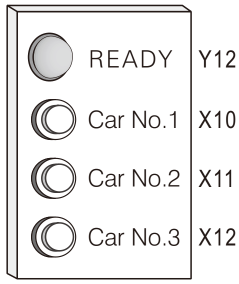
(b) Programming by the STL



4-7-4 Mechanical Double-Decked Parking Space



The Calling Panel



The car parking in the lower level is available to drive out. but the upper level is for storage only. When the READY indicator is ON and press a No. button on the calling panel, the decks will start to move. The decks on the upper level (No.1, 2, 3 parking spaces) can only move up or down, while the decks on the lower level (No.4, 5 parking spaces) can only move horizontally. When the upper parking space moves to the lower level, the car picking-up action is completed, and the READY indicator will be ON again.

As in the picture above, the No. 2 button is pressed in to pick the car up. Firstly the No. 5 deck should move to the empty space at its right. When it reaches the position, the No. 2 decks begins to decline to the lower level, and then the car picking-up action is completed.

The operation in this example is composed of the following three steps to complete the car picking-up action:

Step 1: Except for the target car, move all cars which belong to the upper parking spaces back to the upper level (if any parked currently in the lower level). The S10, S20 or S30 executes the movement in this step.

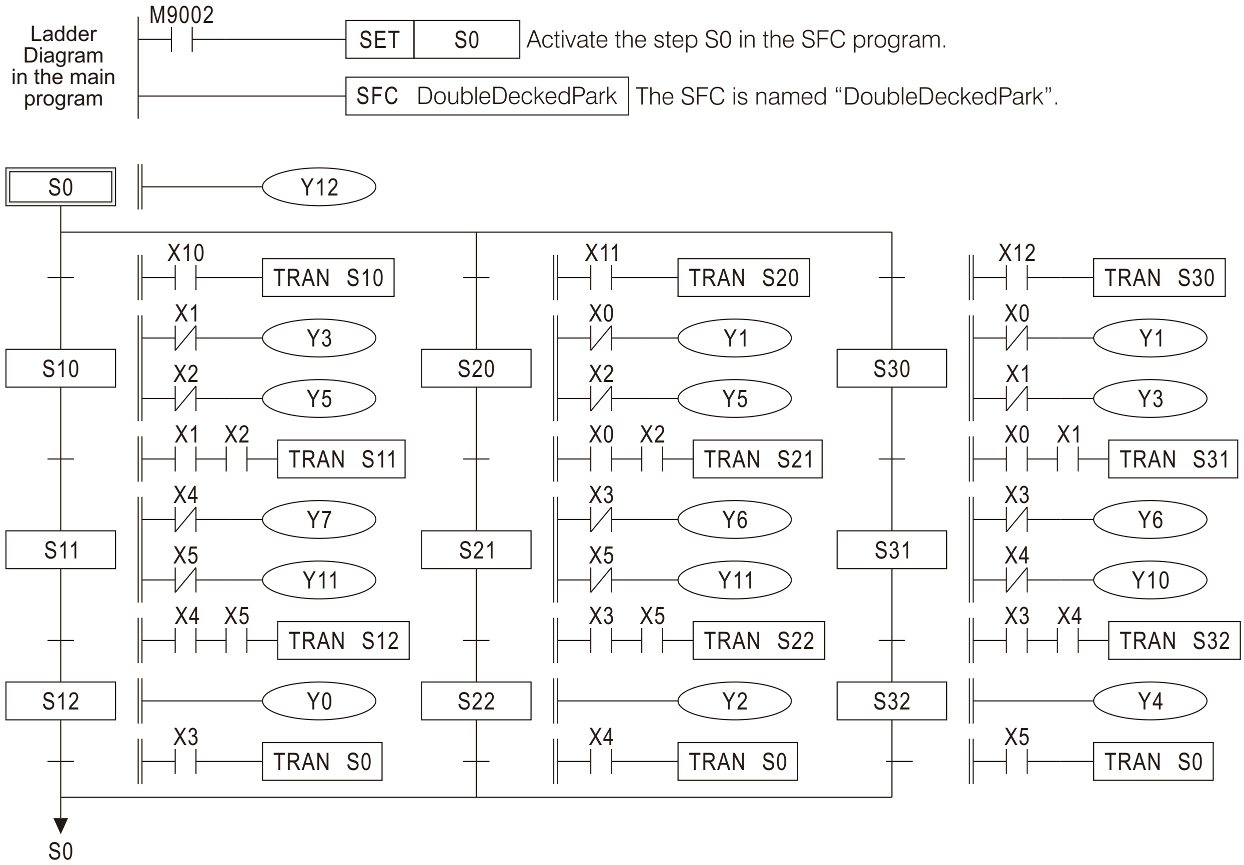
Step 2: The deck below the target car should be removed in order to make a space for it. The S11, S21 or S31 executes the movement in this step.

Step 3: The target car will descend to the lower level and the car picking-up action is complete. The S12, S22 or S32 executes the movement in this step.

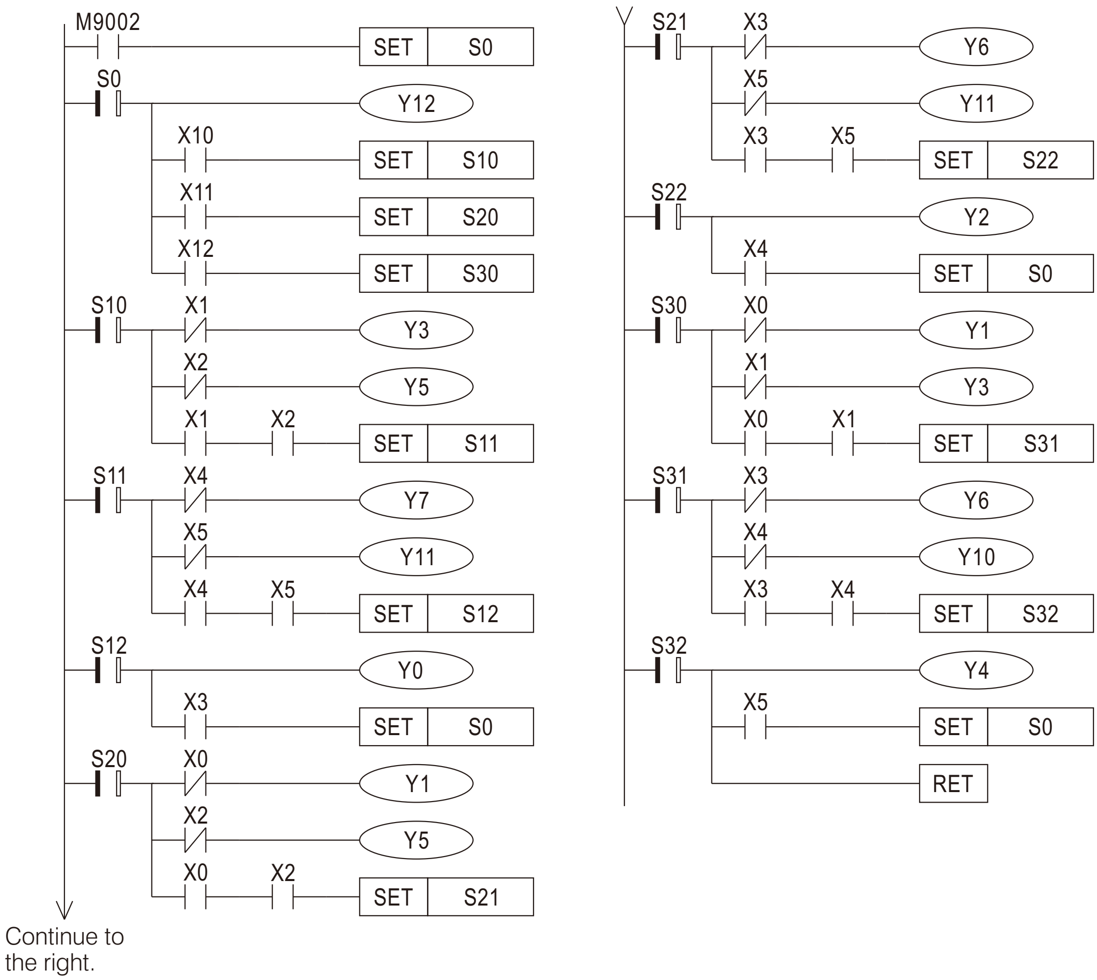
The operation is by pressing the button to select a process flow and its following steps complete that sequentially. This operation fully featured the SFC's attributes of being easy to understand.

Below individually using (a) SFC and (b) STL in writing the program, in order to comparison reference.

(a) Programming by the SFC



(b) Programming by the STL

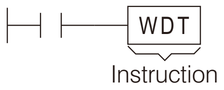


5 General Rules of Application Instructions

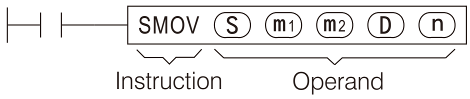
5-1 The Formats of Application Instructions

Instruction and Operand

- Each application instruction has its unique instruction mnemonic, e.g. ADD, CMP.... , etc.
- Some application instructions are purely made up of themselves:



- Most of the application instructions are constituted by instruction themselves and several “Operands”:



As shown above S, m_1, m_2, D, n are Operands. There are many types of Operand in application instructions, their symbolic meanings are:

S : Source Operand (device). It usually refers to the Operand with unchanged contents after executed.
S_1, S_2 ... represent multiple source Operands for an instruction.

D : Destination Operand (device). It usually refers to the Operand in which instruction execution outcomes are stored.
D_1, D_2 ... 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 or R to execute indirect specification. m_1, m_2, n_1, n_2... represent multiple m, n.

- For programming software such as the Ladder Master S, a space is needed between the instruction and each operand to separate them.

Devices for Operand

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

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1							●	●	●	●	●	●	●	○	●	●		
S_2							●	●	●	●	●	●	●	○	●	●		
D		●	●	●	○	○												

- The “●” in the table above indicates the operand which can be modified by an Index Register V, Z; while the “○” means not modifiable.
- In the application instructions, if a D.b, R.b or V, Z is specified as the device of the operand, using V or Z for modification is prohibited.
- After organized, continuous bit devices are displayed as K*n*X, K*n*Y, K*n*M, K*n*S to store data. That “*n*” behind K is to represent how many continuous “Nibbles” will be used.
- Those T and C in the table above refer the Present Value registers of Timer (T) and Counter (C).
- All of T0~T511, C0~C199, D and R are 16-bit registers. When the instruction specifies the process of 32-bit data, two consecutive 16-bit registers will be occupied. For example, if a 32-bit instruction specifies to use the D100 as its Operand, then a 32-bit register (which is composed of D101 and D100) will be occupied. While the D101 will assign to the upper 16 bits and the D100 is the lower 16 bits. The same rule is also applied to the T, C and R.
- The 32-bit Counter (C200~C255) can be used as the Operand of a 32-bit instruction only.
- The UnG in the table refers to the BFM in the special module. For example, U1G3 indicates the operand of this instruction is specified to the BFM #3 in the special module No. 1. Since the programming software Ladder Master S at the monitoring mode could not to read the status at the special module's BFM directly, the operand in the instruction which is appointed to the UnG will not have the correct value to display.
- The letter E in the table refers to a real number constant.
- The “$” in the table refers to a string that is composed of ASCII code.

16-bit or 32-bit Instructions

- Because of different Operand value sizes, some of the application instruction can be organized into 16-bit instruction or 32-bit instruction.

```
            (S) (D)
|--| |----[MOV D0 D10]
```
A 16-bit instruction, the content of D0 is transferred to D10.

```
             (S) (D)
|--| |----[DMOV D0 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.
- The 32-bit Counter (C200~C255) can be used as the Operand of a 32-bit instruction only.
- At the instruction list, if a sign is marked on the left side of the instruction mnemonic, the explanation is below:

① There is no other extra sign on the left of instruction, this indicates it can be executed by the way of 16-bit only. For example, —[BMOVP (S) (D) (n)]

② There is a reverse out D on the left, this indicates that it can be either 16 or 32-bit instruction. For example: —[DMOVP (S) (D)]

③ There is a framed D on the left, this indicates that the instruction is only 32-bit instruction. For example: —[DHSCS (S1) (S2) (D)]

Pulse Execution Instructions

- Based on requires, some application instruction can be organized into continuous execution instruction or pulse execution instruction.

```
   X20
|--| |----[MOV D0 D1]
```
Sequential execution instruction: When X20 = "ON", the instruction will be executed once in each scan cycle.

```
   X20
|--| |----[MOVP D0 D1]
```
Pulse execution instruction: The instruction is only executed once when X20 = "OFF"→"ON".

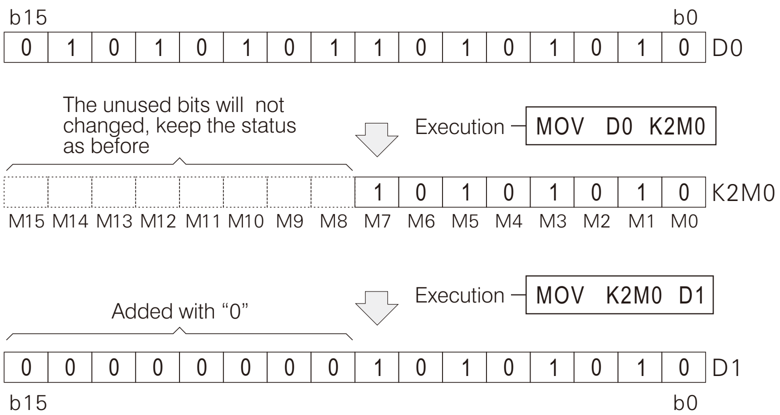
- A pulse instruction displayed with a "P" (to be added directly AFTER the instruction mnemonic), e.g. MOV represents a sequential execution instruction, while MOVP represents a pulse execution instruction.
- Suitable using pulse execution instructions to replace sequential execution instructions in a program, can cut down unnecessary execution time.
- When X20 = "OFF", both MOV and MOVP instuctions are not executed.
- At the instruction list, if a sign is marked on the right side of the instruction mnemonic, the explanation is below:

① There is no other extra sign on the right of instruction, this indicates that the instruction is sequential execution instruction only. For example, —[INCD (S1) (S2) (D) (n)]

② There is a reverse out P on the right, this indicates that it can be either a sequential execution instruction or pulse execution instruction. For example: —[DMOVP (S) (D)]

5-2 Data Process of Application 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 D and R 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 an application instruction 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 (a Nibble). 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 or *n* Nibbles). Thence, the bit devices to be used as a word component that number of organized bits is the multiple of 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 recommend to specify the K*n*X or K*n*Y is by the ID number ended with "0", such as the X0, X20, Y20, Y30..., while to specify the ID number of the K*n*M or K*n*S is the multiple of "8", such as the M0, M8, M16... The recommendations can improve system efficiency.
- When the Operand of an application instruction is transformed to several sequential devices, the sequential ID number at different types are referred as below:

  ① Word device (16 bits)
  D0; D1; D2; D3 ....
  R0; R1; R2; R3 ....
  T0; T1; T2 ....
  C0; C1; C2 ....

  ② Double-word device (32 bits)
  D0 (D1, D0); D2 (D3, D2); D4 (D5,D4) ....
  R0 (R1, R0); R2 (R3, R2); R4 (R5, R4) ....
  T0 (T1, T0); T2 (T3, T2); T4 (T5, T4) ....
  C200; C201; C202 ....

  ③ Word device which is composed of bit devices
  K1X20; K1X24; K1X30; K1X34 ....
  K2Y20; K2Y30; K2Y40; K2Y50 ....
  K3M0; K3M12; K3M24; K3M36 ....
  K4S0; K4S16; K4S32; K4S48 ....

5-3 Precautions of Using Application Instruction

Flags

- The execution result is related to the Application Instruction and that will cause some change to the corresponding flags:

  M9020 : Addition / Subtraction Zero Flag

  M9021 : Borrow Flag

  M9022 : Carry Flag

  M9023 : Multiplication / Division Zero Flag

  M9025 : Division Overflow Flag

  M9029 : Instruction Execution Completed Flag
- When processing an Application Instruction, the status (ON/OFF) of corresponding flags will be changed according to the result; but when the Application Instruction is not executed, the status of corresponding flags will be kept. Since there are so many instructions in the program will change and cover the status of same flags, to plan a program must notice about this.

Floating Point Instructions

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

FLT(FNC49)	DECMP(FNC110)	DEZCP(FNC111)	DEMOV(FNC112)
DESTR(FNC116)	DEVAL(FNC117)	DEBCD(FNC118)	DEBIN(FNC119)
DEADD(FNC120)	DESUB(FNC121)	DEMUL(FNC122)	DEDIV(FNC123)
DEXP(FNC124)	DLOGE(FNC125)	DLOG10(FNC126)	DESQR(FNC127)
DENEG(FNC128)	INT(FNC129)	DSIN(FNC130)	DCOS(FNC131)
DTAN(FNC132)	DASIN(FNC133)	DACOS(FNC134)	DATAN (FNC135)
DRAD(FNC136)	DDEG(FNC137)		

- Every floating point number will occupy two registers.
- The format of floating point number store in registers, please refer to the section 2-13 “Numerical System”.
- If the source operand of a floating point operation instruction is a constant number K or H, the instruction will automatically transform this constant number to a BIN floating point number during the processing.
- When a floating point function is used, please pay attention to the format of its operands.

6.Application Instructions

The VS series PLC has many application instructions, each instruction has its specific function. The PLC will easily achieve a complicated control system also diminish programming codes and programming development time effectively by cleverly using these instructions. We hope readers will have an in-depth understanding of the application instructions and make the best use of them.

6-1 Application Instruction Table

FNC No.	Mnemonic			Brief Function Introduction	Series				Page
					VS1	VS2	VSM	VS3	
Program Flow Instructions									
00		CJ	P	Conditional Jump	○	○	○	○	136
01		CALL	P	Call Subroutine	○	○	○	○	137
02		SRET		Subroutine Return	○	○	○	○	137
03		IRET		Interrupt Return	○	○	○	○	138
04		EI		Enable Interrupt	○	○	○	○	138
05		DI		Disable Interrupt	○	○	○	○	138
06		FEND		First End	○	○	○	○	139
07		WDT	P	Watch Dog Timer Refresh	○	○	○	○	140
08		FOR		Start of a FOR-NEXT Loop	○	○	○	○	141
09		NEXT		End of a FOR-NEXT Loop	○	○	○	○	141
Comparison Instructions									
10	D	CMP	P	Compare	○	○	○	○	144
11	D	ZCP	P	Zone Compare	○	○	○	○	145
224	D	LD=		Initial In-line Compare, (S1) = (S2)	○	○	○	○	346
225	D	LD>		Initial In-line Compare, (S1) > (S2)	○	○	○	○	346
226	D	LD<		Initial In-line Compare, (S1) < (S2)	○	○	○	○	346
228	D	LD<>		Initial In-line Compare, (S1) ≠ (S2)	○	○	○	○	346
229	D	LD<=		Initial In-line Compare, (S1) ≤ (S2)	○	○	○	○	346
230	D	LD>=		Initial In-line Compare, (S1) ≥ (S2)	○	○	○	○	346
232	D	AND=		Serial In-line Compare, (S1) = (S2)	○	○	○	○	346
233	D	AND>		Serial In-line Compare, (S1) > (S2)	○	○	○	○	346
234	D	AND<		Serial In-line Compare, (S1) < (S2)	○	○	○	○	346
236	D	AND<>		Serial In-line Compare, (S1) ≠ (S2)	○	○	○	○	346
237	D	AND<=		Serial In-line Compare, (S1) ≤ (S2)	○	○	○	○	346
238	D	AND>=		Serial In-line Compare, (S1) ≥ (S2)	○	○	○	○	346
240	D	OR=		Parallel In-line Compare, (S1) = (S2)	○	○	○	○	346
241	D	OR>		Parallel In-line Compare, (S1) > (S2)	○	○	○	○	346
242	D	OR<		Parallel In-line Compare, (S1) < (S2)	○	○	○	○	346
244	D	OR<>		Parallel In-line Compare, (S1) ≠ (S2)	○	○	○	○	346
245	D	OR<=		Parallel In-line Compare, (S1) ≤ (S2)	○	○	○	○	346
246	D	OR>=		Parallel In-line Compare, (S1) ≥ (S2)	○	○	○	○	346
Move Instructions									
12	D	MOV	P	Move	○	○	○	○	146
13		SMOV	P	Shift Move	○	○	○	○	147
14	D	CML	P	Complement	○	○	○	○	148
15		BMOV	P	n → n Block Move	○	○	○	○	149
16	D	FMOV	P	1 → n Fill Move	○	○	○	○	150
17	D	XCH	P	Exchange	○	○	○	○	151
Code Exchange Instructions									
18	D	BCD	P	Convert BIN to BCD	○	○	○	○	152
19	D	BIN	P	Convert BCD to BIN	○	○	○	○	152
170	D	GRY	P	Convert BIN to Gray Code	○	○	○	○	309

FNC No.	Mnemonic				Brief Function Introduction	Series				Page
						VS1	VS2	VSM	VS3	
Code Exchange Instructions										
171	D	GBIN	P		Convert Gray Code to BIN	○	○	○	○	310
260	D	DABIN	P		Convert Decimal ASCII String to BIN Number				○	353
261	D	BINDA	P		Convert BIN Number to Decimal ASCII String				○	354
Arithmetic Insructions										
20	D	ADD	P		Addition (S1) + (S2) → (D)	○	○	○	○	154
21	D	SUB	P		Subtraction (S1) – (S2) → (D)	○	○	○	○	155
22	D	MUL	P		Multiplication (S1) × (S2) → (D + 1,D)	○	○	○	○	156
23	D	DIV	P		Division (S1) ÷ (S2) → (D),(D + 1)	○	○	○	○	157
24	D	INC	P		Increment (D) + 1 → (D)	○	○	○	○	158
25	D	DEC	P		Decrement (D) – 1 → (D)	○	○	○	○	158
29	D	NEG	P		Negation ($\overline{D}$) + 1 → (D)	○	○	○	○	160
45	D	MEAN	P		Mean	○	○	○	○	175
48	D	SQR	P		Square Root	○	○	○	○	178
Logical Operation Instructions										
26	D	WAND	P		Logic Word AND	○	○	○	○	159
27	D	WOR	P		Logic Word OR	○	○	○	○	159
28	D	WXOR	P		Logic Word Exclusive OR	○	○	○	○	159
44	D	BON	P		Check Specified Bit Status	○	○	○	○	174
Rotary Instructions										
30	D	ROR	P		Rotation Right	○	○	○	○	162
31	D	ROL	P		Rotation Left	○	○	○	○	162
32	D	RCR	P		Rotation Right with Carry	○	○	○	○	163
33	D	RCL	P		Rotation Left with Carry	○	○	○	○	163
Shift Instructions										
34		SFTR	P		Bit Shift Right	○	○	○	○	164
35		SFTL	P		Bit Shift Left	○	○	○	○	164
36		WSFR	P		Word Shift Right	○	○	○	○	165
37		WSFL	P		Word Shift Left	○	○	○	○	166
213		SFR	P		Shift n Bit Right in 16-bit Word Data with Carry				○	343
214		SFL	P		Shift n Bit Left in 16-bit Word Data with Carry				○	343
Data Table Handling Instructions										
38		SFWR	P		Shift Register Write (FIFO Write)	○	○	○	○	167
39		SFRD	P		Shift Register Read (FIFO Read)	○	○	○	○	168
210		FDEL	P		Delete Data from Specific Location of Table				○	340
211		FINS	P		Insert Data into Specific Location of Table				○	341
212		POP	P		Shift the Last Register Read (FIFO Last Read)				○	342
Data Processing Instructions										
40		ZRST	P		Zone Reset	○	○	○	○	170
41		DECO	P		Decode	○	○	○	○	171
42		ENCO	P		Encode	○	○	○	○	172
43	D	SUM	P		The Sum of Active Bits	○	○	○	○	173
61	D	SER	P		Search a Data Stack	○	○	○	○	198
69		SORT			Sort Tabulated Data	○	○	○	○	207
140	D	WSUM	P		Sum of Word Data				○	286
141		WTOB	P		Split Word to Byte				○	287
142		BTOW	P		Combine Byte to Word				○	288
143		UNI	P		Combine 4-bit Nibble to Word				○	289
144		DIS	P		Separate Word to 4-bit Nibble				○	290

FNC No.	Mnemonic			Brief Function Introduction	Series				Page
					VS1	VS2	VSM	VS3	
Data Processing Instructions									
147	D	SWAP	P	Swap High / Low Byte	○	○	○	○	291
148	D	SORT2		Sort Tabulated Data 2	○	○	○	○	292
Floating Point Arithmetic Instructions									
49	D	FLT	P	BIN Integer → BIN Floating Point Format	○	○	○	○	179
110	D	ECMP	P	Compare Two BIN Floating Point Numbers	○	○	○	○	260
111	D	EZCP	P	Compare a BIN Float No. to BIN Float Zone	○	○	○	○	261
112	D	EMOV	P	Move Floating Point Data	○	○	○	○	262
116	D	ESTR	P	Convert BIN Floating Point to Character String				○	263
117	D	EVAL	P	Convert Character String to BIN Floating Point				○	265
118	D	EBCD	P	Convert BIN to DEC Floating Point Format	○	○	○	○	266
119	D	EBIN	P	Convert DEC to BIN Floating Point Format	○	○	○	○	266
120	D	EADD	P	BIN Floating Point Addition	○	○	○	○	267
121	D	ESUB	P	BIN Floating Point Subtraction	○	○	○	○	268
122	D	EMUL	P	BIN Floating Point Multiplication	○	○	○	○	269
123	D	EDIV	P	BIN Floating Point Division	○	○	○	○	270
124	D	EXP	P	BIN Floating Point Number Exponent	○	○	○	○	271
125	D	LOGE	P	BIN Floating Point Nature Logarithm	○	○	○	○	272
126	D	LOG10	P	BIN Floating Point Common Logarithm	○	○	○	○	273
127	D	ESQR	P	BIN Floating Point Square Root	○	○	○	○	274
128	D	ENEG	P	BIN Floating Point Negation	○	○	○	○	275
129	D	INT	P	BIN Floating Point → BIN Integer Format	○	○	○	○	276
130	D	SIN	P	Calculate Sine	○	○	○	○	277
131	D	COS	P	Calculate Cosine	○	○	○	○	278
132	D	TAN	P	Calculate Tangent	○	○	○	○	279
133	D	ASIN	P	Calculate Arc Sine	○	○	○	○	280
134	D	ACOS	P	Calculate Arc Cosine	○	○	○	○	281
135	D	ATAN	P	Calculate Arc Tangent	○	○	○	○	282
136	D	RAD	P	Convert Angle From Degrees to Radians	○	○	○	○	283
137	D	DEG	P	Convert Angle From Radians to Degrees	○	○	○	○	284
High Speed Processing Instructions									
50		REF	P	I/O Refresh	○	○	○	○	182
51		REFF	P	I/O Refresh and Filter Adjust	○	○	○	○	183
52		MTR		Input Matrix	○	○	○	○	184
53	D	HSCS		Software High Speed Counter Set	○	○	○	○	185
54	D	HSCR		Software High Speed Counter Reset	○	○	○	○	187
55	D	HSZ		Software High Speed Counter Zone Compare	○	○	○	○	188
56	D	SPD		Speed Detection	○	○	○	○	190
57	D	PLSY		Pulse Y Output	○	○	○	○	191
58		PWM		Pulse Width Modulation	○	○	○	○	193
59	D	PLSR		Pulse Ramp	○	○	○	○	194
189	D	HHCMV	P	Hardware High Speed Counter Data Move	○	○	○	○	319
280	D	HSCT		Software High Speed Counter Table Compare	○	○	○	○	355
Handy Instructions									
62	D	ABSD		Absolute Drum Sequencer	○	○	○	○	199
63		INCD		Incremental Drum Sequencer	○	○	○	○	201
64		TTMR		Teaching Timer	○	○	○	○	202
65		STMR		Special Timer	○	○	○	○	203
66		ALT	P	Alternate State	○	○	○	○	204

FNC No.		Mnemonic		Brief Function Introduction	Series				Page
					VS1	VS2	VSM	VS3	
Handy Instructions									
67		RAMP		Ramp Variable Value	○	○	○	○	205
88		PID		PID Control Loop	○	○	○	○	233
90		DBRD	P	Read Data From Data Bank	○	○	○	○	248
91		DBWR	P	Write Data Into Data Bank	○	○	○	○	249
92		TPID		Temperature PID Control	○	○	○	○	250
93		DTRD	P	Read Data From Data Table	○	○	○	○	257
102		ZPUSH	P	Batch Store of All Index Register				○	258
103		ZPOP	P	Batch Recover of All Index Register				○	258
256	D	LIMIT	P	Limit Control				○	348
257	D	BAND	P	Dead Band Control				○	349
258	D	ZONE	P	Zone Shift Control				○	350
259	D	SCL	P	Scaling	○	○	○	○	351
269	D	SCL2	P	Scaling 2	○	○	○	○	351
External Setting and Display Instructions									
70	D	TKY		Ten Key Input	○	○	○	○	210
71	D	HKY		Hexadecimal Key Input	○	○	○	○	211
72		DSW		Digital Switch (Thumbwheel) Input	○	○	○	○	213
73		SEGD	P	Seven Segment Decoder	○	○	○	○	214
74		SEGL		Seven Segment with Latch	○	○	○	○	215
76		ASC		Convert Letters to ASCII Code	○	○	○	○	216
77		PR		Print ASCII Code	○	○	○	○	217
78	D	FROM	P	Read From Special Module		○	○	○	218
79	D	TO	P	Write To Special Module		○	○	○	219
Serial Port Communication Instructions									
80		RS		Receive/Send Communication	○	○	○	○	222
81	D	PRUN	P	Parallel Run (Octal Mode)	○	○	○	○	227
82		ASCI	P	Convert HEX to ASCII	○	○	○	○	228
83		HEX	P	Convert ASCII to HEX	○	○	○	○	229
84		CCD	P	Check Code	○	○	○	○	230
87		CPUL		CPU Link Communication	○	○	○	○	231
89		LINK		Easy Link Communication	○	○	○	○	244
149		MBUS		MODBUS Communication	○	○	○	○	294
Real Time Clock Related Instructions									
160		TCMP	P	Time Compare	○	○	○	○	298
161		TZCP	P	Time Zone Compare	○	○	○	○	299
162		TADD	P	Time Addition	○	○	○	○	300
163		TSUB	P	Time Subtraction	○	○	○	○	301
164	D	HTOS	P	Convert Hour to Second	○	○	○	○	302
165	D	STOH	P	Convert Second to Hour	○	○	○	○	303
166		TRD	P	Read RTC Data	○	○	○	○	304
167		TWR	P	Set RTC Data	○	○	○	○	305
Timer Instructions									
169	D	HOUR		Hour Meter	○	○	○	○	308
176		TFT		Timer (10 ms.)	○	○	○	○	311
177		TFH		Timer (100 ms.)	○	○	○	○	312
178		TFK		Timer (1 s.)	○	○	○	○	313
Block Data Handling Instructions									
192	D	BK+	P	Block Data Subtraction				○	322

FNC No.	Mnemonic			Brief Function Introduction	Series				Page
					VS1	VS2	VSM	VS3	
Block Data Handling Instructions									
193	D	BK－	P	Block Data Subtraction				○	323
194	D	BKCMP＝	P	Block Data Compare (S1) = (S2)				○	324
195	D	BKCMP＞	P	Block Data Compare (S1) > (S2)				○	324
196	D	BKCMP＜	P	Block Data Compare (S1) < (S2)				○	324
197	D	BKCMP＜＞	P	Block Data Compare (S1) ≠ (S2)				○	324
198	D	BKCMP＜＝	P	Block Data Compare (S1) ≤ (S2)				○	324
199	D	BKCMP＞＝	P	Block Data Compare (S1) ≥ (S2)				○	324
Character String Handling Instructions									
200	D	STR	P	BIN to Character String Conversion				○	326
201	D	VAL	P	Character String to BIN Conversion				○	328
202		$+	P	Join Up Two Character Strings				○	330
203		LEN	P	Character String Length Detection				○	331
204		RIGHT	P	Read Character from the Right of String				○	332
205		LEFT	P	Read Character from the Left of String				○	333
206		MIDR	P	Read Character from Specific Place of String				○	334
207		MIDW	P	Write Character to Specific Place of String				○	335
208		INSTR	P	Search Character String from another String				○	337
209		$ MOV	P	Transfer Character String				○	338
Positioning Control Instructions									
300	D	ZRN		Zero Return	○	○	○	○	434
301	D	JOGF		Jog Forward	○	○	○	○	436
302	D	JOGR		Jog Reverse	○	○	○	○	436
303	D	DRVR		Drive to Relative Position	○	○	○	○	438
304	D	DRVA		Drive to Absolute Position	○	○	○	○	440
305	D	DV2R		Drive to Relative Position by 2 Stages	○	○	○	○	442
306	D	DV2A		Drive to Absolute Position by 2 Stages	○	○	○	○	444
307	D	DVIT		Interrupt Constant Quantity Positioning	○	○	○	○	446
308	D	DV2I		2 Stages Interrupt Constant Quantity Position	○	○	○	○	448
309	D	DVSR		IngInterrupt to Stop or Drive to Relative Position	○	○	○	○	450
310	D	DVSA		Interrupt to Stop or Drive to Absolute Position	○	○	○	○	452
311	D	PLSV		Variable Speed Pulse Output	○	○	○	○	454
312		DTBL		Data Table Positioning	○	○	○	○	456
313	D	ABS		Absolute Current Value Read	○	○	○	○	462
314		MPG		Handwheel Positioning	○	○	○	○	463
315	D	LIR		Relatively Linear Interpolation	○	○	○	○	466
316	D	LIA		Absolutely Linear Interpolation	○	○	○	○	469
Other Instructions									
46		ANS		Timed Annunciator Set	○	○	○	○	176
47		ANR	P	Annunciator Reset	○	○	○	○	176
184		RND	P	Generate Random Number	○	○	○	○	316
186		DUTY		Timing Pulse Generation				○	317
188		CRC	P	Cyclic Redundancy Check - 16	○	○	○	○	318

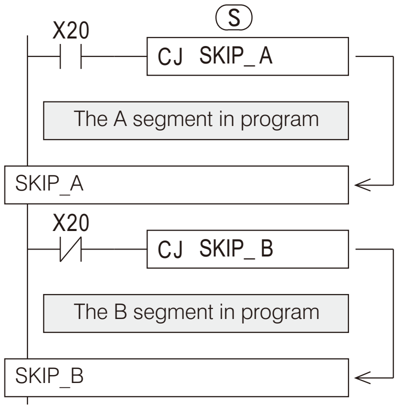
MEMO

6-2 Program Flow Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
00	CJP (S)	Conditional Jump	○	○	○	○
01	CALLP (S)	Call Subroutine	○	○	○	○
02	SRET	Subroutine Return	○	○	○	○
03	IRET	Interrupt Return	○	○	○	○
04	EI	Enable Interrupt	○	○	○	○
05	DI	Disable Interrupt	○	○	○	○
06	FEND	First End	○	○	○	○
07	WDTP	Watch Dog Timer Refresh	○	○	○	○
08	FOR (n)	Start of a FOR-NEXT Loop	○	○	○	○
09	NEXT	End of a FOR-NEXT Loop	○	○	○	○

FNC 00	CJP (S)	Conditional Jump	1	2	M	3
			○	○	○	○

Operand	Devices	
	P	By a numeric Branch Pointer P or an user-defined 16 English characters Mark Point
S	●	○
• Numeric Branch Pointer P: P0~P1023		• The total amount of the Mark and Branch Pointers is 1024.



S : the destination Pointer of the Conditional Jump instruction

- When the conditional contact for the CJ instruction is "OFF" (CJ is not active), the segment in the program between the CJ instruction and its pointer will execute normally.
  When the conditional contact for the CJ instruction is "ON" (CJ is active), the system will execute the Jump action to the destination point of CJ, then executes the following program after the pointer.
- When X20 = "OFF", the "CJ SKIP_B" instruction will execute the Jump action, then the B segment will not be executed.
- When X20 = "ON", the "CJ SKIP_A" instruction will execute the Jump action then the A segment 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 action, every device of the skipped segment will as follows:

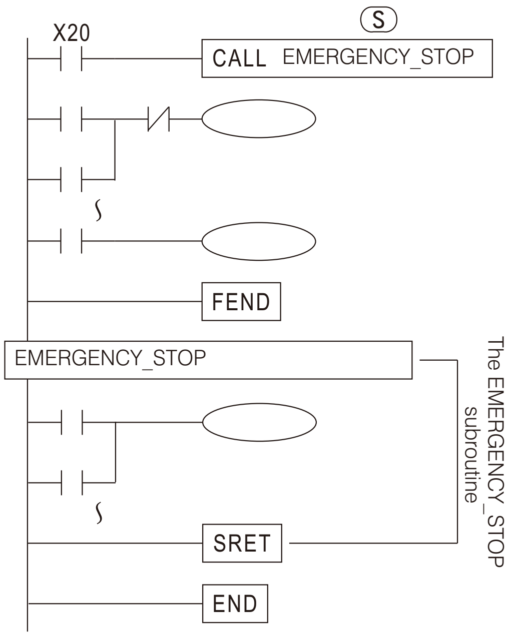
The action of every device in the bypassed segment during the Jump is executing

- The status of Y, M and S will remain as same as before the Jump.
- The 10 ms. or 100 ms. timer will pause counting time.
- The 1 ms. timer will continue to count time, but its output contact will not function normally until the Jump finishes.
- T192~T199 will continue to count time and its output contact will also effective.
- High Speed Counter will continue to count and its output contact will also effective.
- A normal counter will stop counting.
- If the Reset instruction for a Retentive timer or counter is driven before Jump, the device will still be reset during the Jump.
- Applied instruction will not be executed.

- Using the CJ instruction can skip unnecessary program segment directly, so the program Scan Time can be saved.
- The CJ instruction can be used to solve the problem of double coil outputs.
- A numeric pointer or the name of a pointer can only appear once in a program; If the Pointer is specified more than once, errors will be incurred.
- As the Pointer P63 or P255 is equal to the END address in the program, to execute the "CJ P63" or "CJ P255" is equal to jump to the END of program.

FNC	Instruction	Description	1	2	M	3
FNC 01	CALLP S	Call Subroutine	○	○	○	○
FNC 02	SRET	Subroutine Return	○	○	○	○

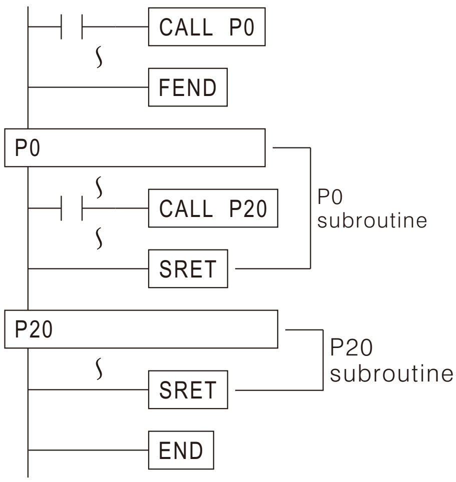
Operand	Devices	
	P	By a numeric Branch Pointer P or an user-defined 16 English characters Mark Point
S	●	○
• Numeric Branch Pointer P: P0~P1023	• The total amount of the Mark and Branch Pointers is 1024.	



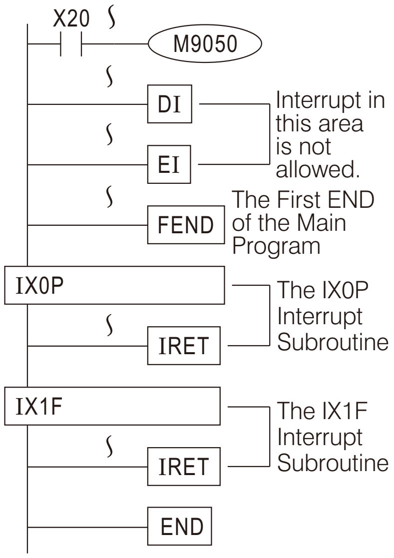
S : the Pointer of the subroutine

- When the conditional contact X20 for the CALL instruction is “ON”, the system will execute the CALL action to jump to the pointer of “EMERGENCY_STOP” and then executes the subroutine. Until the SRET instruction in the subroutine executes, the system jumps back the line 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 is not allowed to assign for both.
- A same subroutine can be called in a program as many times as required.
- In a subroutine, a second CALL instruction is available for calling other subroutine, while subroutines can be nested for 5 levels at most.
- The Timers used in the subroutine must be selected from T192~T199.

- 2-Level Nest Subroutine Call (5 levels at the most)



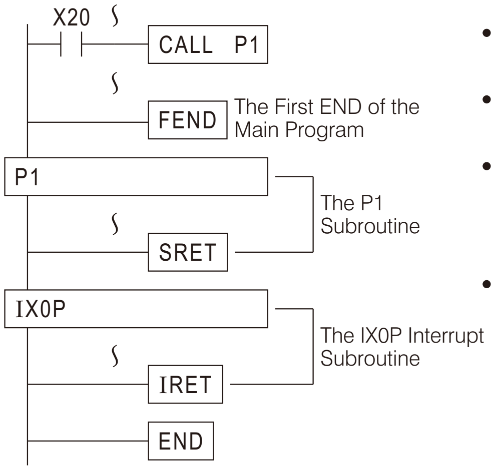
			1	2	M	3
FNC 03	IRET	Interrupt Return	○	○	○	○
FNC 04	EI	Enable Interrupt	○	○	○	○
FNC 05	DI	Disable Interrupt	○	○	○	○



- Generally a program is under Enable Interrupt status. Except for the program flow is during the area between the DI and EI instructions, where interruption is prevented.
- Assume that programs are under Enable Interrupt status:
  When X0 = "OFF"→"ON", the IX0P 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", the IX1F 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 Relay M9050 is active and then "X0" is driven to disable Interrupt, the interrupt from the external input X0 is blocked.
- Please write the Interrupt Pointer "IX□□", "IHC□□", "IHHC□" or "IT□□□" after the FEND instruction.
- When the program flow executing an interrupt subroutine, all other interrupts are not allowed.

- The Timers used in general subroutines and interrupt subroutines must be selected from the T192~T199.
- When the program flow is worked between the DI and EI instructions, instructions, an interrupt demand cannot be executed immediately. The demand will be memorized and executed when the interrupt function is allowed again.
- For the pulse width of the external interrupt signal, please refer to the input reaction time varies with the series of PLC.
- If the interrupt subroutine's I/O needs processed instantly, please use the FNC50 REF immediate I/O refresh instruction.

FNC 06	FEND	First End	1	2	M	3
			○	○	○	○



- An FEND instruction indicates the first end of the main program block.
- An FEND instruction placed before the CALL instruction or after the SRET instruction will be treated as an error.
- The Interrupt Pointer “I” (including all the “IX□□”, “IHC□□”, “IHHC□”, “IT□□□”) and the Mark / Branch Point “P” (which is specified by the 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 the END instruction.

FNC 07	WDT P	Watch Dog Timer Refresh	1	2	M	3
			○	○	○	○

A PLC is provided with the WDT (Watch Dog Timer), which is used to monitor operation condition of the PLC system. By way of the WDT to monitor the process, when the PLC's CPU runs abnormally, that will command the PLC to stop operation and turn all external output “OFF”. This system achieves the protection purpose.

The explanation of the WDT (Watch Dog Timer):

The WDT is a hardware timer with 200 ms. set value. When PLC = “STOP”→“RUN”, the value of WDT will reload from the content value of Special Register D9000 and while the initial value of D9000 is “200”.

This timer is counting downward by a timing unit of 1 ms. If the current value reaches “0”, the 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.

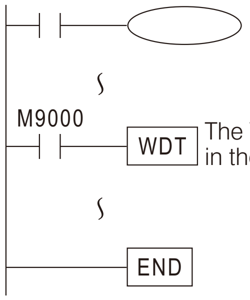
When the system operates normally, PLC will revert its WDT timer before it executes the beginning of program (STEP 0).

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

For the second cause, here provide two solutions:

(1) Insert WDT instruction into the program, because WDT instruction will revert the current value of WDT.

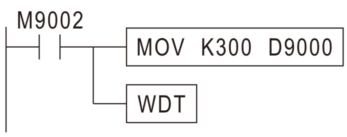


The WDT instruction reverts the current value of the WDT timer to the content value of D9000 in the middle of the program.

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

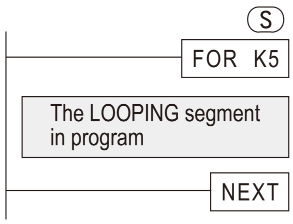
Set the WDT to be a timer of 300 ms

To adopt this approach, it should be noted that at the first Scan Time of the PLC (from “STOP” to “RUN”), the value of WDT timer is still 200 ms. The program below can be used for the solution where necessary.



FNC 08	FOR (S)	Start of a FOR-NEXT Loop	1 ○	2 ○	M ○	3 ○
FNC 09	NEXT	End of a FOR-NEXT Loop	1 ○	2 ○	M ○	3 ○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
• S = 1~32,767 (beyond the range, the set value S is regarded as 1)																		

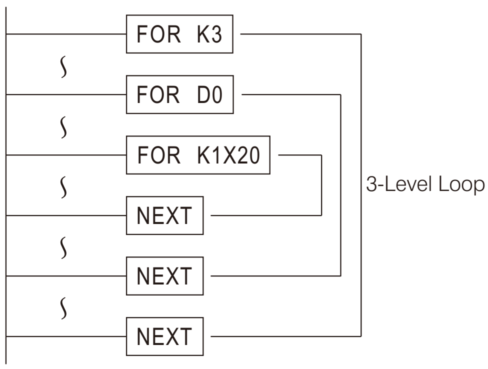


S : the number of times to be repeated in the FOR-NEXT loop.

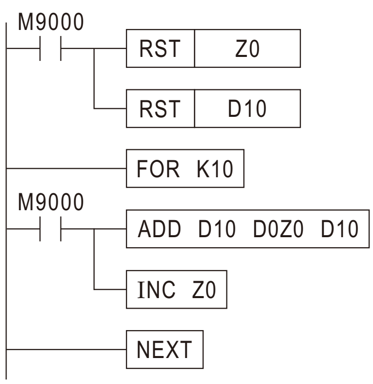
- The program segment in the FOR-NEXT loop will be executed (S) times.
- As in the left diagram, the LOOPING segment in program will be executed 5 sequentially at every scan time.

- In a FOR-NEXT loop, the CJ instruction can be used to jump out of the loop.
- At most 5 levels can be used for a nest FOR~NEXT loop. Be sure to note that the loops will cause the currently Scan Time extended. The Scan Time should be taken not to exceed WDT's default value, otherwise an error will occur.
- Errors will occur under the following circumstances:
  ① The NEXT instruction is placed in front of the FOR instruction.
  ② The NEXT instruction is placed behind the FEND or END instruction.
  ③ The FOR instruction and the NEXT instruction are not programmed as a pair.

- Multiple-level Loop Program



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



MEMO

6-3 Comparison, Move and Code Exchange Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
10	D CMP P S1 S2 D	Compare	○	○	○	○
11	D ZCP P S1 S2 S D	Zone Compare	○	○	○	○
12	D MOV P S D	Move	○	○	○	○
13	SMOV P S m1 m2 D n	Shift Move	○	○	○	○
14	D CML P S D	Complement	○	○	○	○
15	BMOV P S D n	n→n Block Move	○	○	○	○
16	D FMOV P S D n	1→n Fill Move	○	○	○	○
17	D XCH P D1 D2	Exchange	○	○	○	○
18	D BCD P S D	Convert BIN to BCD	○	○	○	○
19	D BIN P S D	Convert BCD to BIN	○	○	○	○

FNC 10	D CMP P (S1) (S2) (D)	Compare	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D		●	●	●	○	○												
• D occupies 3 consecutive devices																		

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

S_1 : the compare value #1

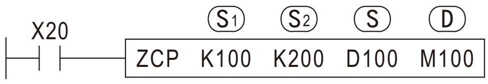
S_2 : the compare value #2

D : the compare result; occupying 3 consecutive points

- Compare the content value of (S1) (Compare Value #1) with the value of (S2) (Compare Value #2), and save the result in (D) (Compare Result).
- The CMP instruction will be enabled when X20 = “ON”.
  If (S1) > (S2) (K100 > D100), then M100 = “ON”
  If (S1) = (S2) (K100 = D100), then M101 = “ON”
  If (S1) < (S2) (K100 < D100), then M102 = “ON”.
- When X20 = “OFF”, the instruction is inactive, the “ON”/“OFF” statuses of M100, M101 and M102 will remain as same as the statuses before it is disabled.
- Please use serial or parallel links of M100~M102 to generate the result as “≥”, “≤” or “≠”.

FNC 11	D ZCP P (S1) (S2) (S) (D)	Zone Compare	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1							●	●	●	●	●	●	●	○	●	●		
S_2							●	●	●	●	●	●	●	○	●	●		
S							●	●	●	●	●	●	●	○	●	●		
D		●	●	●	○	○												
• D occupies 3 consecutive devices	• $S_1 \le S_2$																	



S_1 : the lower limit of zone compare

S_2 : the upper limit of zone compare

S : the compare value

D : the compare result; occupying 3 consecutive points

- Compare the content value of (S) (Compare Value) with the values of (S1) (lower limit of zone compare) and (S2) (upper limit of zone compare), then save the result in (D) (compare result).

- The ZCP instruction will be enabled when X20 = "ON".
  If (S) < (S1) (D100 < K100, the compare value is less than the lower limit), then M100 = "ON";
  If (S1) ≤ (S) ≤ (S2) (K100 ≤ D100 ≤ K200, the content value of D100 is between the upper and lower limit), then M101 = "ON";
  If (S) > (S2) (D100 > K200, the compare value is bigger than the upper limit), then M102 = "ON".

- When X20 = "OFF", the instruction is inactive, the "ON"/"OFF" statuses of M100, M101 and M102 will remain as same as the statuses before it is disabled.

- When (S1) > (S2) , the content value of (S1) will become both upper/lower limits to be compared with the (S) .

FNC 12	D MOV P (S) (D)	Move	1 2 M 3

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

S : the source device to be transferred
D : the destination device

- To copy the designated value from (S) to (D).
- The content value of D100 will be copied to D200 when X20 = "ON".
- When X20 turns from "ON" to "OFF", the instruction is inactive besides the content value of D200 will remain as same as the value before it is disabled.

Bit Data Transfer

To perform the program of left diagram, which can be achieved by the right side.

32-bit Data Transfer

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

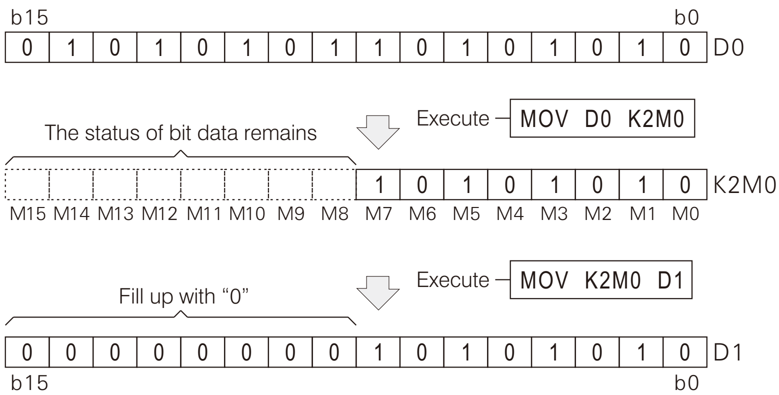
When X20 = "ON", move the content value (D1, D0) to (D101, D100).

If the transfer source or destination device is a 32-bit counter, the instruction should be headed with a "D".

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

Data Transfer Between Word and Bits

When using the MOV instruction to transfer data between words and bits, those bits should be organized by the unit of "Nibble". For example, (that K2 in the program means 2 Nibbles)



FNC 13	SMOVP (S) (m1) (m2) (D) (n)	Shift Move	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●			
m1																○		
m2																○		
D								●	●	●	●	●	●	○	●			
n																○		

• m_1 = 1~4 • m_2 = 1~m_1 • n = m_2~4

X20 —| |— SMOV D0 K3 K2 D1 K2
(S) (m1) (m2) (D) (n)

S : the source device of transfer

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

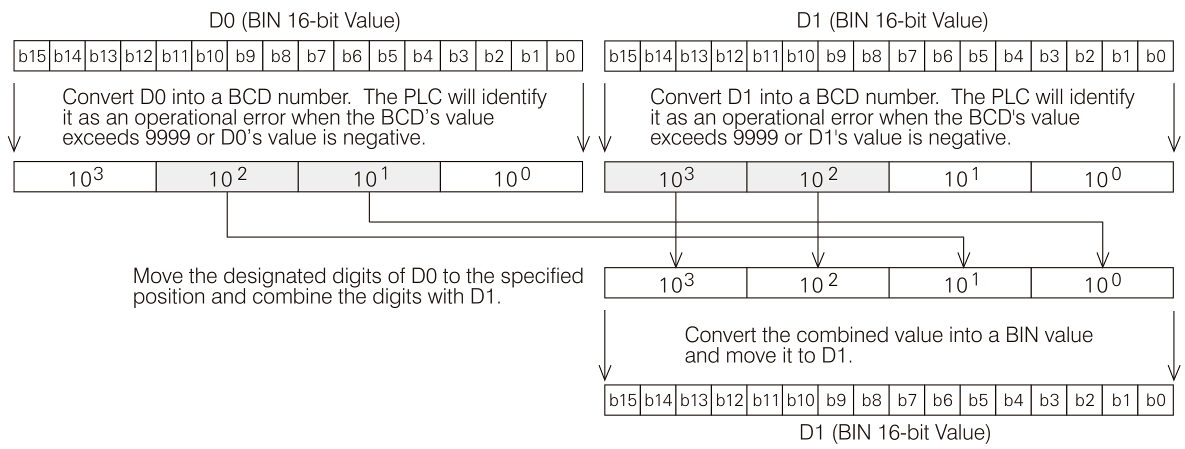
m_2 : the number of source digits to be moved

D : the destination device

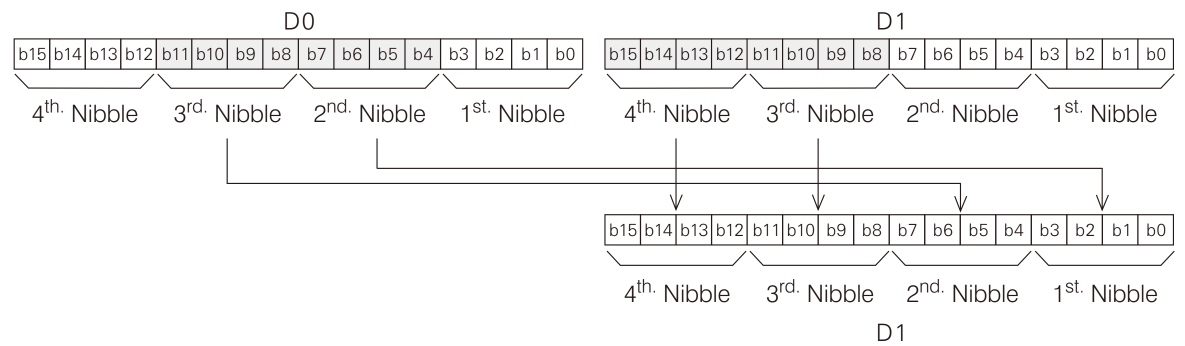
n : the destination position for the first digit

- This instruction can be used to reorganize data.
- The instruction can select different operation modes, it is based on the status of Special Relay M9168.

When the M9168 = "OFF"



When the M9168 = "ON"



FNC 14	DCMLP (S) (D)	Complement	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

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

S : the source device of transfer
D : the destination device

- Invert all the content bits from the designated (S) (status of each bit, “0” inverts to “1” or “1” inverts to “0”) and copy the result to (D) .
- When X20 = “ON”, all of contents of D0 are inverted and copied to D1.
- When X20 turns from “ON” to “OFF”, the instruction is inactive besides the content value of D1 will remain as same as the value before it is disabled.

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

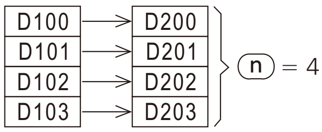
↓ X20 = ON

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

FNC 15	BMOVP (S) (D) (n)	n→n Block Move	1	2	M	3
			○	○	○	○

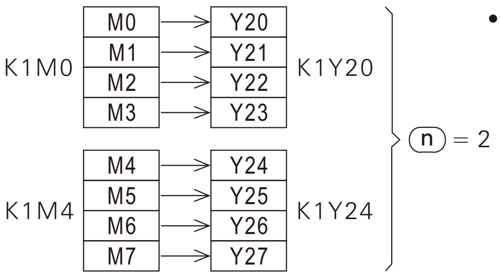
Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
n													○			○		
• The S and D occupies n consecutive devices respectively										• n = 1~512								

S : the head ID of source device
D : the head ID of destination device
n : the length of the block to be moved



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

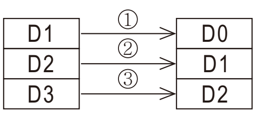
- When a block transfer of organized bit devices is executed, the number of nibbles of (S) and (D) should correspond.



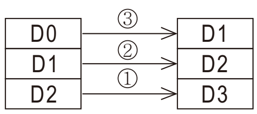
- When X20 = "ON", two nibbles in the K1M0 and K1M4 will be copied to the K1Y20 and K1Y24 (that equals to move the block M0~M7 to Y20~Y27).

- 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)



- The Special Relay M9024 is assigned as the move direction control flag for the BMOV instruction. The "ON"/"OFF" status of M9024 can determine the data move direction of the BMOV instruction.

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

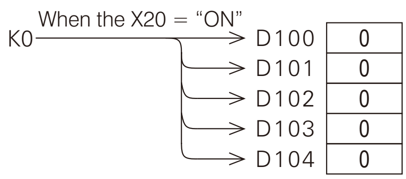
FNC 16	DFMOVP (S) (D) (n)	1 → n Fill Move	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●		●			
n																○		

• For 16-bit instruction, D occupies n devices • For 32-bit instruction, D occupies n×2 devices • n = 1~512

S : the source device of transfer
D : the head ID of destination device
n : the length of the block to be moved

- Move the content value of (S) to (n) registers which headed with (D) .
- When X20 = “ON”, the value of K0 will be copied to 5 continuous registers that headed with D100 (D100~D104).
- If a part of destination devices which is over the available range of the device (usually caused by the content value in the (n) is large), then only the available destination devices will be copied to.



FNC 17	DXCHP D1 D2	Exchange	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D1								●	●	●	●	●	●	○	●			
D2								●	●	●	●	●	●	○	●			

X20 — XCHP D100 (D1) D200 (D2)

D1 : the data #1 to be exchanged

D2 : the data #2 to be exchanged

- Exchange (swap) the content values of the appointed devices D1 and D2 .
- When X20 = “OFF”→“ON”, the 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 (D1) D100 (D2)

- When X21 = “OFF”→“ON”, the content values of (D1, D0) and (D101, D100) 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

FNC	Instruction	Description	1	2	M	3
FNC 18	DBCDP (S) (D)	Convert BIN to BCD	○	○	○	○
FNC 19	DBINP (S) (D)	Convert BCD to BIN	○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●			
D								●	●	●	●	●	●	○	●			

X20 —| |— [BCD D0 K4Y20] (S) (D)

S : the source (BIN) of the conversion

D : the converted result destination (BCD)

- When X20 = "ON", the BIN value at 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 the value in (S) exceeds the available range (0~9,999).
- For a 32-bit instruction, PLC will identify an error when the value in (S) exceeds the available range (0~99,999,999).

X21 —| |— [BIN K4X20 D1] (S) (D)

S : the source (BCD) of the conversion

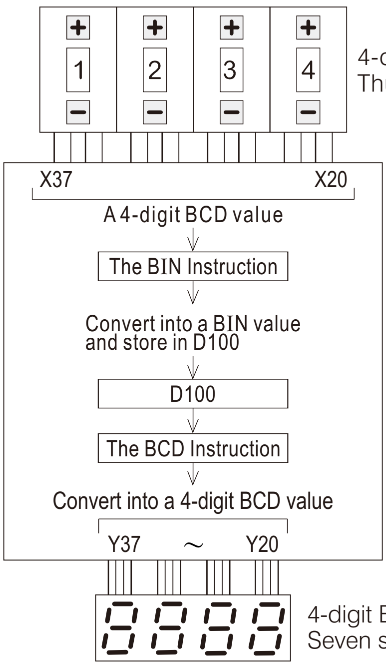
D : the converted result destination (BIN)

- When X21 = "ON", the BCD value from 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 code format, the PLC will regard that as an operational error.

Application of the BCD and BIN Instructions

M9000 —| |— [BIN K4X20 D100] Convert the BCD value from K4X20 into an equivalent BIN value and move it to D100.

[BCD D100 K4Y20] Convert the BIN value at D100 into an equivalent BCD value and move it to K4X20.



4-digit BCD-based Thumbwheel Switch

4-digit BCD-based Seven segment display

- Usually the data in a PLC is stored in a BIN format, and applied instructions (arithmetic and logical operations, etc.) are also executed based on BIN values.
- When a PLC is reading a BCD-based thumbwheel switche data, 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 Operation Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
20	DADDP S1 S2 D	Addition $(S1)+(S2) \rightarrow (D)$	○	○	○	○
21	DSUBP S1 S2 D	Subtraction $(S1)-(S2) \rightarrow (D)$	○	○	○	○
22	DMULP S1 S2 D	Multiplication $(S1)\times(S2) \rightarrow (D+1,D)$	○	○	○	○
23	DDIVP S1 S2 D	Division $(S1)\div(S2) \rightarrow (D),(D+1)$	○	○	○	○
24	DINCP D	Increment $(D)+1 \rightarrow (D)$	○	○	○	○
25	DDECP D	Decrement $(D)-1 \rightarrow (D)$	○	○	○	○
26	DWANDP S1 S2 D	Logic Word AND $(S1)\wedge(S2) \rightarrow (D)$	○	○	○	○
27	DWORP S1 S2 D	Logic Word OR $(S1)\vee(S2) \rightarrow (D)$	○	○	○	○
28	DWXORP S1 S2 D	Logic Word Exclusive OR $(S1)\veebar(S2) \rightarrow (D)$	○	○	○	○
29	DNEGP D	Negation $(\bar{D})+1 \rightarrow (D)$	○	○	○	○

FNC 20	DADDP (S1) (S2) (D)	Addition (S1)+(S2) → (D)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

S1 : the summand
S2 : the addend
D : the sum

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

	10	D0
+	5	D1
	15	D2

- 16-bit Operation

  When the result of an operation stores in the (D) and it is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".

  When the result of an operation exceeds 32,767, the Carry Flag M9022 turns "ON".

  When the result of an operation is less than −32,768, the Borrow Flag M9021 turns "ON".

- When X20 = "ON", the instruction will add the data in (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 stores in the (D) and it is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".

  When the result of an operation exceeds 2,147,483,647, the Carry Flag M9022 turns "ON".

  When the result of an operation is less than −2,147,483,648, the Borrow Flag M9021 turns "ON".

FNC 21	DSUBP (S1) (S2) (D)	Subtraction (S1) – (S2) → (D)	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

S1 : the minuend
S2 : the subtrahend
D : the difference

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

	10	D0
−	5	D1
	5	D2

- 16-bit Operation

  When the result of an operation stores in the (D) and it is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".

  When the result of an operation exceeds 32,767, the Carry Flag M9022 turns "ON".

  When the result of an operation is less than −32,768, the Borrow Flag M9021 turns"ON".

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

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

- 32-bit Operation

  When the result of an operation stores in the (D) and it is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".

  When the result of an operation exceeds 2,147,483,647, the Carry Flag M9022 turns "ON".

  When the result of an operation is less than −2,147,483,648, the Borrow Flag M9021 turns "ON".

FNC 22	⊣ ⊢ DMULP (S1) (S2) (D)	Multiplication (S1)×(S2) → (D+1,D)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

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

S1 : the multiplicand

S2 : the multiplier

D : the product (of a multiplication)

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

	10	D0
×	5	D1
	50	(D3, D2)

- Two 16-bit data sources multiplied together will produce a 32-bit product.
- The Most Significant Bit (MSB) of a 32-bit product indicates a positive or negative ("0" represents it is positive; "1" negative).
- When the product of an operation stores in the (D) and it is equal to "0", the Multiplication / Division Zero Flag M9023 turns "ON".

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

- When X20 = "OFF"→"ON", the instruction will 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)

- Two 32-bit data sources multiplied together will create a 64-bit product.
- The Most Significant Bit (MSB) of a 64-bit product indicates a positive or negative ("0" represents it is positive; "1" negative).
- When the product of an operation stores in the (D) and it is equal to "0", the Multiplication / Division Zero Flag M9023 turns "ON".

FNC 23	DDIVP (S1) (S2) (D)	Division (S1)÷(S2) → (D),(D+1)	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

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

S1 : the dividend
S2 : the divisor
D : the 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 of the 16-bit instruction, a 16-bit quotient and a 16-bit remainder are produced. The Most Significant Bit of a component will indicate a positive or negative ("0" represents it is positive; "1" negative).

- When the quotient of an operation stores in the (D) and it is equal to "0", the Multiplication / Division Zero Flag M9023 turns "ON".
  When the quotient of an operation exceeds 32,767, the Division Overflow Flag M9025 turns "ON".

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

- When X20 = "OFF"→"ON", the dividend (D1, D0) will be divided by the divisor (D3, D2), and the quotient will be stored at the destination device (D5, D4) while the remainder will be stored in (D7, D6).

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

- In case of the 32-bit instruction is used, a 32-bit quotient and a 32-bit remainder are produced. The Most Significant Bit of a component will indicate a positive or negative ("0" represents it is positive; "1" negative).

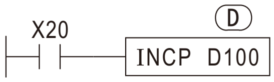
- When the quotient of an operation stores in the (D) and it is equal to "0", the Multiplication / Division Zero Flag M9023 turns "ON"·
  When the quotient of an operation exceeds 2,147,483,647, the Division Overflow Flag M9025 turns "ON".

Note

- If the divisor (S2) is equal to "0", the PLC will regard that as an operational error.
- The quotient will automatically be positive when it is the result of operating a positive dividend and a positive divisor, or a negative dividend and a negative divisor; 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.

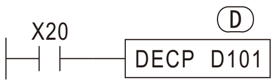
FNC 24	DINCP (D)	Increment (D)+1 → (D)	1	2	M	3
			○	○	○	○
FNC 25	DDECP (D)	Decrement (D) – 1 → (D)	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D								●	●	●	●	●	●	○	●			



D : the destination device

- When X20 = "OFF"→"ON", the content value of destination (D100) will be increased by one.
- If the instruction is not a pulse (P) instruction, the destination (D100) will increase its value 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 : the destination device

- When X20 = "OFF"→"ON", the content value of destination (D100) will be decreased by one.
- If the instruction is not a pulse (P) instruction, the destination (D100) will decrease its value 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.

FNC	Instruction	Function	1	2	M	3
FNC 26	DWANDP (S1) (S2) (D)	Logic Word AND $(S1) \wedge (S2) \rightarrow (D)$	○	○	○	○
FNC 27	DWORP (S1) (S2) (D)	Logic Word OR $(S1) \vee (S2) \rightarrow (D)$	○	○	○	○
FNC 28	DWXORP (S1) (S2) (D)	Logic Word Exclusive OR $(S1) \veebar (S2) \rightarrow (D)$	○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

X20 ── WAND D0 D1 D2 (S1) (S2) (D)

S1 : the source device #1
S2 : the source device #2
D : the operation result

- When X20 = "ON", two 16-bit content data in the (D0) and (D1) execute the logic AND operation and restore the result in (D2).
- The logic AND operation result for each bit is: $0 \wedge 0 = 0$, $0 \wedge 1 = 0$, $1 \wedge 0 = 0$ or $1 \wedge 1 = 1$; any "0" at the parallel bits 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

↓ X20 = ON

0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	D2

X21 ── WOR D3 D4 D5 (S1) (S2) (D)

S1 : the source device #1
S2 : the source device #2
D : the operation result

- When X21 = "ON", two 16-bit content data in the (D3) and (D4) execute the logic OR operation and restore the result in (D5).
- The logic OR operation result for each bit is: $0 \vee 0 = 0$, $0 \vee 1 = 1$, $1 \quad 0 = 1$ or $1 \vee 1 = 1$; any "1" at the parallel bits 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

↓ X21 = ON

0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	D5

X22 ── WXOR D6 D7 D8 (S1) (S2) (D)

S1 : the source device #1
S2 : the source device #2
D : the operation result

- When X22 = "ON", two 16-bit content data in the (D6) and (D7) execute the logic XOR operation and restore the result in (D8).
- The logic XOR operation result for each bit is: $0 \veebar 0 = 0$, $0 \veebar 1 = 1$, $\veebar 0 = 1$ or $1 \veebar 1 = 0$; the same statuses at the parallel bits will cause a result of "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

↓ X22 = ON

0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	D8

FNC 29	DNEGP (D)	Negation $(\overline{D})+1 \rightarrow (D)$	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D								●	●	●	●	●	●	○	●			

```
 X20            (D)
─┤ ├───── NEGP  D0
```

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 selects 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	Before Execution	−100	D0
	↓ X20 = OFF→ON			↓ X20 = OFF→ON	
After Execution	−10	D0	After Execution	100	D0

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

```
 M9000
─┤ ├───── BON  D100  M0  K15
 M0
─┤ ├───── NEGP  D100
```

If the 15th. bit (MSB) of D100 is equal to "1", it indicates that the content value of D100 is negative, where M0 = "ON"; otherwise, M0 = "OFF".

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

- The absolute value of D0 can be generated with the following program.

```
 D0.F
─┤ ├───── NEGP  D0
```

If the 15th. bit (MSB) of D0 is equal to "1", it indicates that the content value of D0 is negative. Therefore, the absolute value of D0 can be generated through opposite.

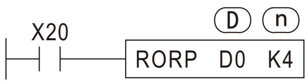
6-5 Rotary and Shift Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS 1	2	M	3
30	DRORP D n	Rotation Right	○	○	○	○
31	DROLP D n	Rotation Left	○	○	○	○
32	DRCRP D n	Rotation Right with Carry	○	○	○	○
33	DRCLP D n	Rotation Left with Carry	○	○	○	○
34	SFTRP S D n1 n2	Bit Shift Right	○	○	○	○
35	SFTLP S D n1 n2	Bit Shift Left	○	○	○	○
36	WSFRP S D n1 n2	Word Shift Right	○	○	○	○
37	WSFLP S D n1 n2	Word Shift Left	○	○	○	○
38	SFWRP S D n	Shift Register Write (FIFO Write)	○	○	○	○
39	SFRDP S D n	Shift Register Read (FIFO Read)	○	○	○	○

FNC 30	DRORP (D) (n)	Rotation Right	1	2	M	3
			○	○	○	○
FNC 31	DROLP (D) (n)	Rotation Left	1	2	M	3
			○	○	○	○

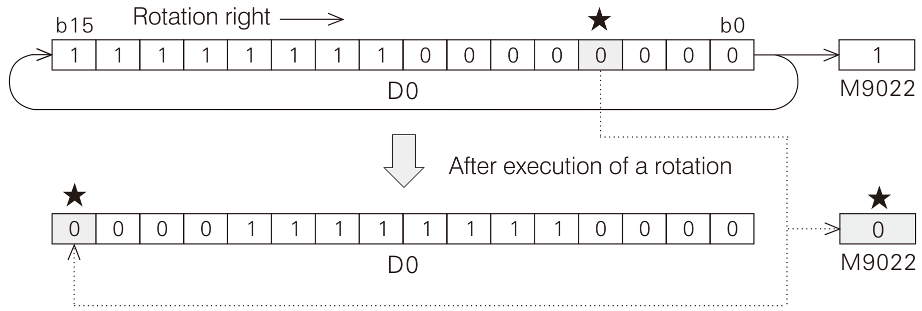
Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D								●	●	●	●	●	●	○	●			
n													○			○		

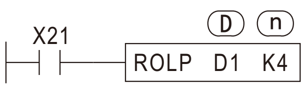
- The 16-bit instruction, n = 1~16 • The 32-bit instruction, n = 1~32
- When D is designated as K*n*Y, K*n*M and K*n*S, the 16-bit instruction can only designate K4Y, K4M and K4S, while the 32-bit instruction can only designate K8Y, K8M and K8S



D : the selected device to be rotated
n : the number of the bits to be rotated

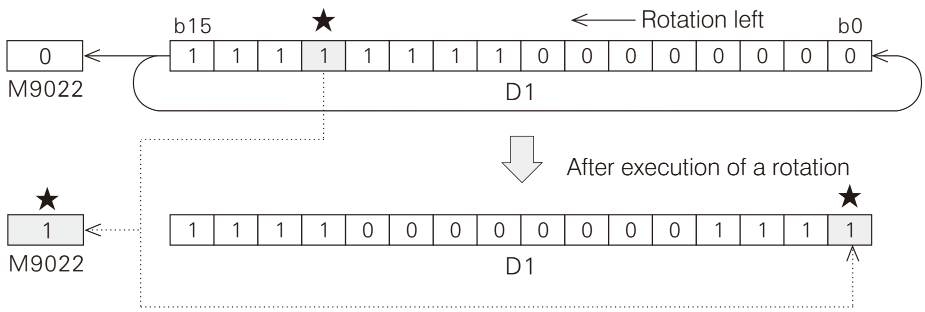
- The bit format of the designated device (D) executes to rotate to the right by the number of (n) bits.
- When X20 = "OFF"→"ON", the 16-bit data of (D0) will be rotated by 4 bits to the right, and the status of the highest bit that rotated out will be copied to the Carry Flag M9022.





D : the selected device to be rotated
n : the number of the bits to be rotated

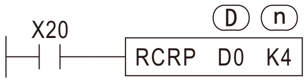
- The bit format of the designated device (D) executes to rotate to the left by the number of (n) bits.
- When X21 = "OFF"→"ON", the 16-bit data of (D1) will be rotated by 4 bits to the left , and the status of the lowest bit that rotated out will be copied to the Carry Flag M9022.



FNC 32	D RCR P (D) (n)	Rotation Right with Carry	1	2	M	3
			○	○	○	○
FNC 33	D RCL P (D) (n)	Rotation Left with Carry	1	2	M	3
			○	○	○	○

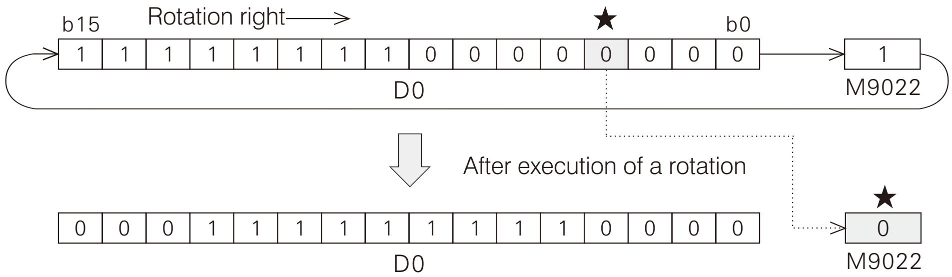
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D								●	●	●	●	●	●	○	●			
n													○			○		

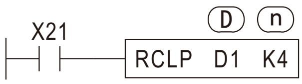
- The 16-bit instruction, n = 1~16 • The 32-bit instruction, n = 1~32
- When D is designated as K*n*Y, K*n*M and K*n*S, the 16-bit instruction can only designate K4Y, K4M and K4S, while the 32-bit instruction can only designate K8Y, K8M and K8S



D : the selected device to be rotated
n : the number of the bits to be rotated

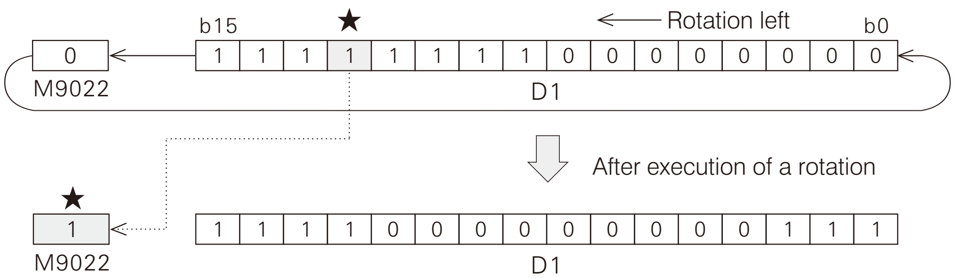
- The bit format of the designated device (D) join with the Carry Flag M9022 that executes to rotate to the right by the number of (n) bits.
- When X20 = "OFF"→"ON", the 16-bit data of (D0) combine with the Carry Flag M9022 that will be rotated by 4 bits to the right.





D : the selected device to be rotated
n : the number of the bits to be rotated

- The bit format of the designated device (D) join with the Carry Flag M9022 that executes to rotate to the left by the number of (n) bits.
- When X21 = "OFF"→"ON", the 16-bit data of (D1) combine with the Carry Flag M9022 that will be rotated by 4 bits to the left.



FNC 34	SFTRP (S) (D) (n1) (n2)	Bit Shift Right	1 ○	2 ○	M ○	3 ○
FNC 35	SFTLP (S) (D) (n1) (n2)	Bit Shift Left	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●	●	●	●	○	○												
D		●	●	●														
n1																○		
n2													○			○		

• S occupies n_2 components • D occupies n_1 components • n_1 = 1~1024 • n_2 = 1~n_1

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

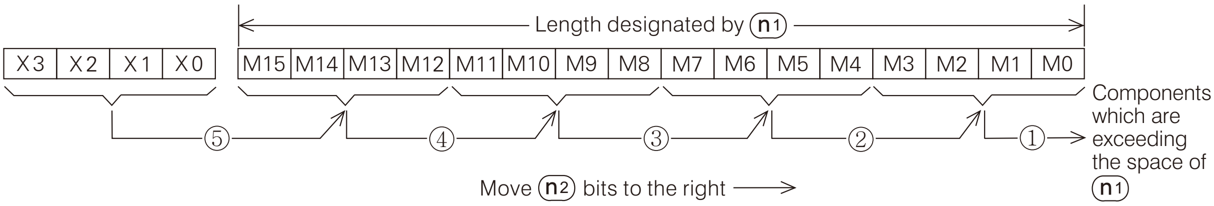
S : the device ID number of source header to be moved in

D : the device ID number of the destination header to be shifted

n1: data length to be shifted at the destination

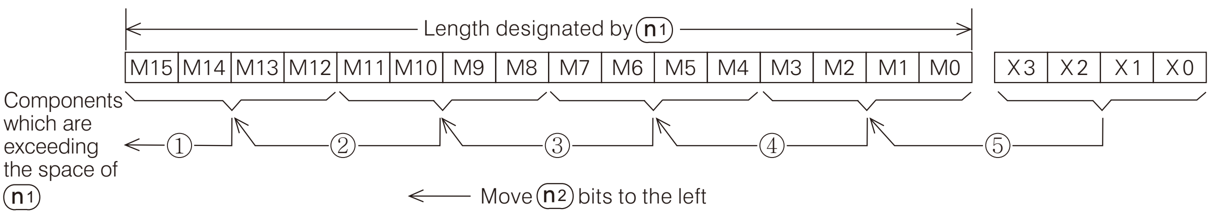
n2: number of the bits in a shift

- Pick the continuous n1 length bit devices destination which headed by D and shift the destination to the right by the length of n2, then use the number of n2 source bit devices which headed by S to fill in the vacancy.
- When X20 = "OFF"→"ON", the devices composed of M0~M15 (16 bits) will be moved 4 bits to the right, also the status of X0~X3 will be moved into M12~M15 to fill in the vacancy of exceeding bits.



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

- Pick the continuous n1 length bit devices destination which headed by D and shift the destination to the left by the length of n2, then use the number of n2 source bit devices which headed by S to fill in the vacancy.
- When X21 = "OFF"→"ON", the devices composed of M0~M15 (16 bits) will be moved 4 bits to the left, also the status of X0~X3 will be moved into M0~M3 to fill in the vacancy of exceeding bits.



FNC 36	WSFRP (S) (D) (n1) (n2)	Word Shift Right	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
n1																○		
n2													○			○		

• S occupies n2 components • D occupies n1 components • n1 = 1~512 • n2 = 1~n1

X20 ── WSFRP D100 D0 K12 K3 (S D n1 n2)

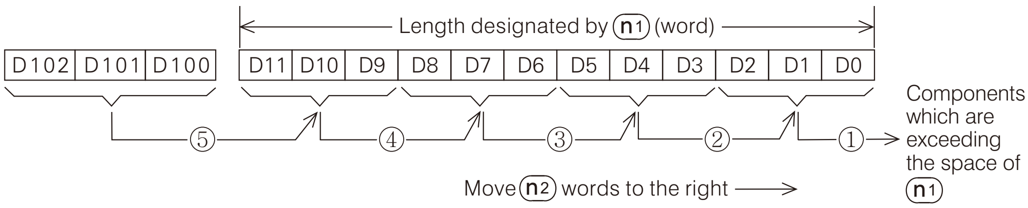
S : the device ID number of source header to be moved in

D : the device ID number of the destination header to be shifted

n1: data length to be shifted at the destination

n2: number of the words in a shift

- Pick the continuous (n1) length word stack destination which headed by (D) and shift the destination to the right by the length of (n2), then use the number of (n2) source word stack which headed by (S) to fill in the vacancy.
- When X20 = "OFF"→"ON", the word stack which is composed of D0~D11 (12 words) will be moved 3 words to the right, also the contents of D100~D102 will be moved into D9~D11 to fill in the vacancy of exceeding words.

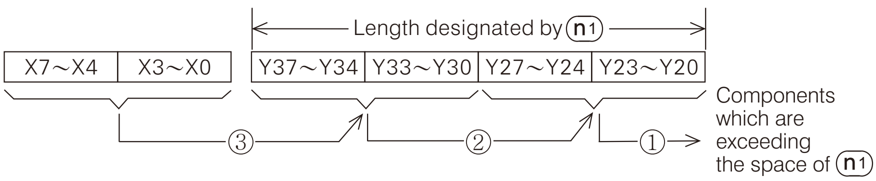


Note

- The properties of devices specified by (S) and (D) must be the same (both are word devices or bit devices).
- When (S) and (D) are designated to organized bit devices, the number of nibbles at the K*n* must be the same.

X20 ── WSFRP K1X0 K1Y20 K4 K2 (S D n1 n2)

the number of K*n* designated by (S) and (D) must be the same



FNC 37	WSFLP (S) (D) (n1) (n2)	Word Shift Left	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
n1																○		
n2													○			○		

• S occupies n2 components • D occupies n1 components • n1 = 1~512 • n2 = 1~n1

```
X21                      (S)   (D)  (n1) (n2)
─┤├──────[ WSFLP  D100  D0  K12  K3 ]
```

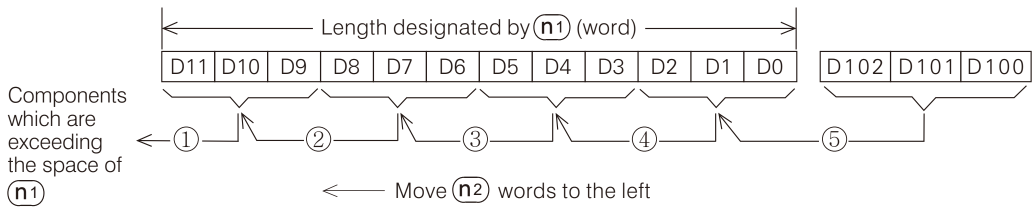
S : the device ID number of source header to be moved in

D : the device ID number of the destination header to be shifted

n1: data length to be shifted at the destination

n2: number of the words in a shift

- Pick the continuous (n1) length word stack destination which headed by (D) and shift the destination to the left by the length of (n2), then use the number of (n2) source word stack which headed by (S) to fill in the vacancy.
- When X21 = "OFF"→"ON", the word stack which is composed of D0~D11 (12 words) will be moved 3 words to the left, also the contents of D100~D102 will be moved into D0~D2 to fill in the vacancy of exceeding words.

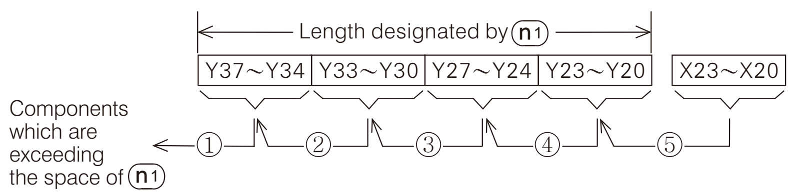


Note

- The properties of devices specified by (S) and (D) must be the same (both are word devices or bit devices).
- When (S) and (D) are designated to organized bit devices, the number of nibbles at the K*n* must be the same.

```
X0                     (S)     (D)    (n1) (n2)
─┤├──────[ WSFLP  K1X20  K1Y20  K4  K1 ]
```

the number of K*n* designated by (S) and (D) must be the same

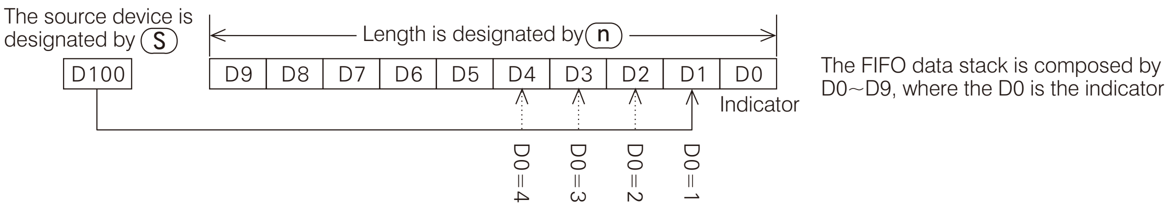


FNC 38	SFWRP (S) (D) (n)	Shift Register Write (FIFO Write)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●		●			
n																○		
• D occupies n components	• n = 2~512																	

S : the source device to be written to a FIFO data stack
D : the header ID number of the FIFO data stack
n : the length of the FIFO data stack

- A series of word data can be used for the FIFO (First In, First Out) data stack access. The FIFO data stack is headed by (D) , the size is assigned by (n) and the stack's first device is designated to be the indicator. When each time this instruction is enabled, the content value of the indicator will be added with “1” firstly, and then the content value of the designated device (S) will be moved to the position that is designated by the indicator in the FIFO data stack.



- Suppose at the very beginning that D0 = 0. At the first time X20 = “OFF”→“ON”, the content value of D0 will become “1” and the content value of D100 will be moved to the D1. At the second time X20 = “OFF”→“ON”, the content value of D0 will become “2” and the content value of D100 will be moved to the D2, and so forth.

- The D0 records the position where it is written to the FIFO data stack at the last time. If the content value of D0 is ≥ ((n) −1) and the instruction is active again, it will not to write data into the stack any more and the value of D0 will remain the previous value also the Carry Flag M9022 = “ON”.

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

- This SFWR instruction can be used jointly with the FNC212 POP instruction to achieve the write / read control of the FILO (First In, Last Out) data stack.

FNC 39	SFRDP (S) (D) (n)	Shift Register Read (FIFO Read)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S								●	●	●	●	●	●		●			
D								●	●	●	●	●	●	○	●			
n																○		
• S occupies n components	• n = 2~512																	

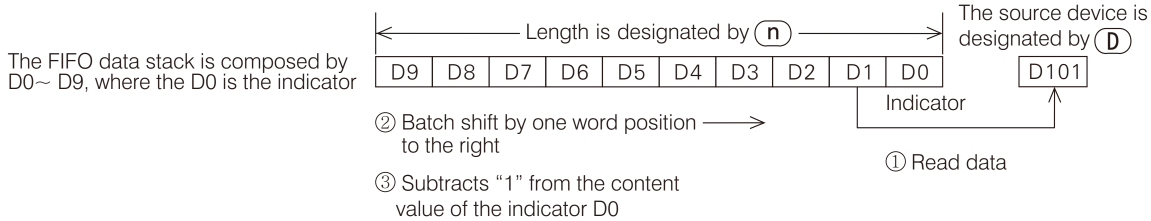
X21 — SFRDP D0 D101 K10 (S: D0, D: D101, n: K10)

S : the header ID number of the FIFO data stack

D : the destination device for storing read data from a FIFO data stack

n : the length of the FIFO data stack

- A series of word data can be used for the FIFO (First In, First Out) data stack access. The FIFO data stack is headed by (S), the size is assigned by (n) and the stack's first device is designated to be the indicator. When each time this instruction is enabled, it moves the content value of the second device in the FIFO data stack to the destination device (D) firstly, then the rest data in the stack (without the moved out device and the indicator) will be shifted one word position to the right, furthermore subtracts “1” from the indicator’s content value.



- Suppose at the very beginning that D0 = 5. At the first time X21 = “OFF”→“ON”, the content value of D1 will be moved to the D101 then the content values at D2~D9 will be shifted one word position to the D1~D8 also the content value of D0 will subtract “1” become “4”. At the second time X21 = “OFF”→“ON”, again the content value of D1 will be moved to the D101 then the content values at D2~D9 will be shifted one word position to D1~D8 also the content value of D0 will become “3”, and so forth.

- When the content value of D0 is equal to “0” and the instruction is active again, it will not execute to read data, decrease the indicator or shift process any more. So the value of (D101) will remain the previous value, and furthermore the Zero Flag M9020 = “ON”.

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

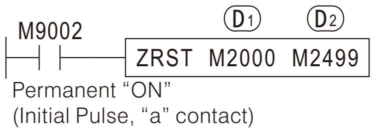
6-6 Data Processing Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
40	ZRSTP D1 D2	Zone Reset	○	○	○	○
41	DECOP S D n	Decode	○	○	○	○
42	ENCOP S D n	Encode	○	○	○	○
43	DSUMP S D	The Sum of Active Bits	○	○	○	○
44	DBONP S D n	Check Specified Bit Status	○	○	○	○
45	DMEANP S D n	Mean	○	○	○	○
46	ANS S m D	Timed Annunciator Set	○	○	○	○
47	ANRP	Annunciator Reset	○	○	○	○
48	DSQRP S D	Square Root	○	○	○	○
49	DFLTP S D	BIN Integer → Binary Floating Point Format	○	○	○	○

FNC 40	ZRSTP (D1) (D2)	Zone Reset	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D1		●	●	●							●	●	●		●			
D2		●	●	●							●	●	●		●			

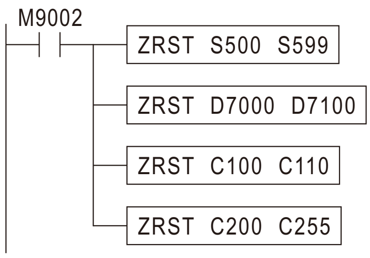
- The ID number of device D1 must be less than or equal to (≤) the device D2.
- The D1 and D2 must use the same device type.



D1 : the starting device for the zone reset operation

D2 : the ending device for the zone reset operation

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



- The ZRST instruction can be used to reset various devices that include bit devices and word devices.
- The (D1) and (D2) must use the same device type. Also, the ID number of the device (D1) must be less then or equal to (≤) the number of (D2) . Only the device which is designated by (D1) 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 if that (D1) designates to a 16-bit counter while (D2) designates to a 32-bit counter.

FNC 41	DECOP S D n	Decode	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●	●	●	●							●	●	●	○	●	●		
D		●	●	●							●	●	●	○	●			
n																○		

- If S is a bit device, it occupies n bits
- If D is a bit device, n = 1~8 and D occupies 2^n bits
- If D is a word device, n = 1~4

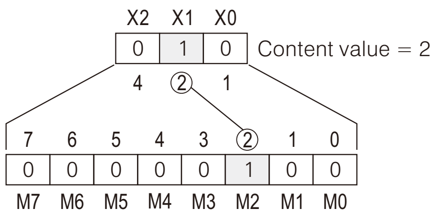
S : the source device to decode

D : the destination device where the decoded result is stored

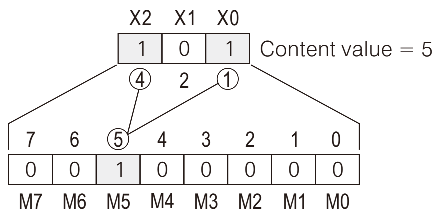
n : the length of bits to decode

- In this example, D is designated to a group of bit devices, so the instruction will occupy consecutive 2^n bit devices that are headed by D for to store the decoded results.

- Suppose the content value of X0~X2 is 2. When X20 = "OFF" → "ON", the DECO instruction will decode the content value of X0~X2 and move the results to M0~M7, where is only to set the M2 = "ON".



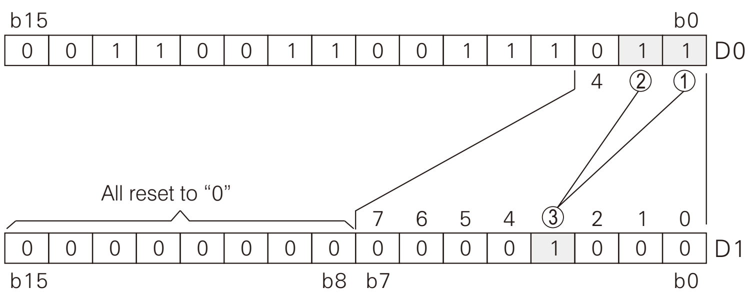
- Suppose the content value of X0~X2 is 5. When X20 = "OFF" → "ON", the DECO instruction will decode the content value of X0~X2 and move the results to M0~M7, where is only to set the M5 = "ON".



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

- In this example, D is designated to a word device, therefore the range of n = 1~4.

- When X20 = "OFF" → "ON", the DECO instruction 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 reset to "0".



FNC 42	ENCOP (S) (D) (n)	Encode	1 ○	2 ○	M ○	3 ○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●	●	●	●							●	●	●	○	●			
D											●	●	●	○	●			
n																○		

• If S is a bit device, n = 1~8 and S occupies 2^n bits • If S is a word device, n = 1~4

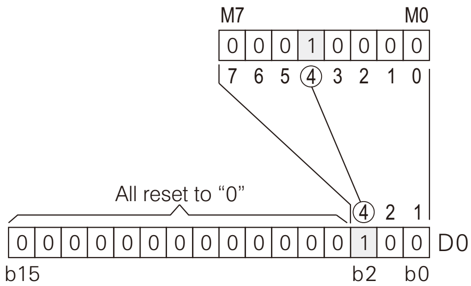
X20 — ENCOP M0 D0 K3 (S) (D) (n)

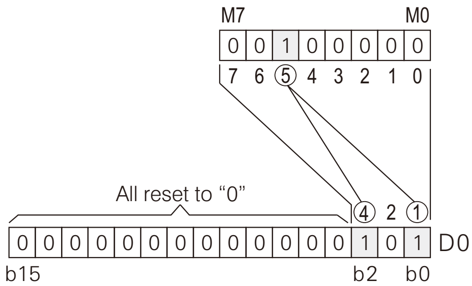
S : the source device to encode

D : the destination device where the encoded result is stored

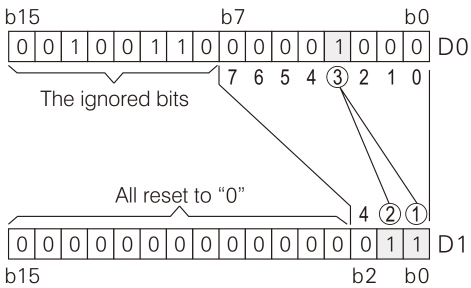
n : the length of bits to encode

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

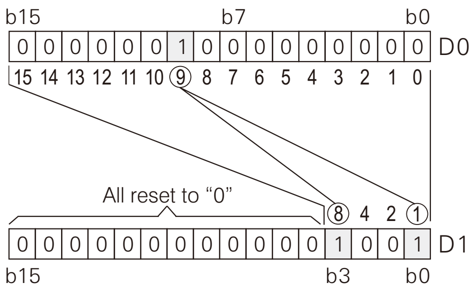




X20 — ENCOP D0 D1 K3 (S) (D) (n)



X20 — ENCOP D0 D1 K4 (S) (D) (n)



Note

- If the status of content bits of (S) that have more than one are equal to "1", the instruction will only focus on the largest ID number.
- If the status of content bits of (S) are all equal to "0", the PLC will regard that as an operational error.
- When the conditional contact turns "OFF", the encoded results (statuses at (D)) will remain as same as the statuses before it is disabled.

FNC 43	DSUMP S D	The Sum of Active Bits	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

X20 — SUM D0 D10 (S D)

S : the source device to be count

D : the destination device to store processed data

- When X20 = "ON", the instruction is to count the number of active bits within the 16 bits of D0, and the amount will be stored in D10. If all of the 16 bits of D0 are inactive (equal to "0"), then the Zero Flag M9020 = "ON".

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

D0 —X20 = ON→ 8 (D10)

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

FNC 44	DBONP (S) (D) (n)	Check Specified Bit Status	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D		●	●	●	○	○												
n													○			○		

• The 16-bit instruction, n = 1~15 • The 32-bit instruction, n = 1~31

X20 — BON D0 M0 K5 (S) (D) (n)

S : the source device
D : the destination device where specified results are stored
n : the designated position bit to be specified

• Copy the status of the (n) th. bit at the designated source device (S) to the destination device (D) .

• When X20 = “ON”, the status of b5 at the D0 will be copied to M0.

• When X20 = “OFF”, the status of M0 will remain as same as the status before it is disabled.

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

FNC 45	DMEANP (S) (D) (n)	Mean	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●	○	●			
n													○			○		

• The 16-bit instruction, S occupies n components • The 32-bit instruction, S occupies (n×2) components • n = 1~64

X20 — MEAN D0 D10 K5 (S) (D) (n)

S : the head ID of source devices to be generated a mean

D : the destination device where the mean is stored

n : the number of consecutive devices to be generated a mean

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

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

Value	Device
100	D0
150	D1
200	D2
88	D3
100	D4

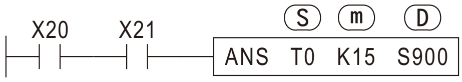
X20 = ON → 127 D10

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

- Ignore the remainder if any remainder appears during the operation process.
- If the some of the consecutive source devices exceed the available range, only the devices within the prescribed range will be processed.

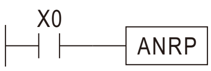
FNC 46	ANS (S) (m) (D)	Timed Annunciator Set	1 ○, 2 ○, M ○, 3 ○
FNC 47	ANRP	Annunciator Reset	1 ○, 2 ○, M ○, 3 ○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●							
m													○			○		
D				●														
• S = T0~T199	• m = 1~32767	• D = S900~S999																



S : the timer is used to be detected and cause alarm
m : the set value of timer configuration
D : the annunciator

- The instruction ANS is used exclusively to drive the output of an annunciator step relay.
- When X20 and X21 turn “ON” for more than 1.5 seconds simultaneously, the annunciator S900 is driven “ON”. After S900 = “ON” and when X20 or X21 turns “OFF”, the contact of the non-retentive T0 becomes “OFF” and its present value is returned as “0” , but the S900 will remain “ON”.
- When both X20 and X21 are “ON” but this duration is less than 1.5 seconds, then either one of them turns “OFF”, the present value of T0 will be returned as “0”.
- Do not use a timer which has been assigned to this instruction.



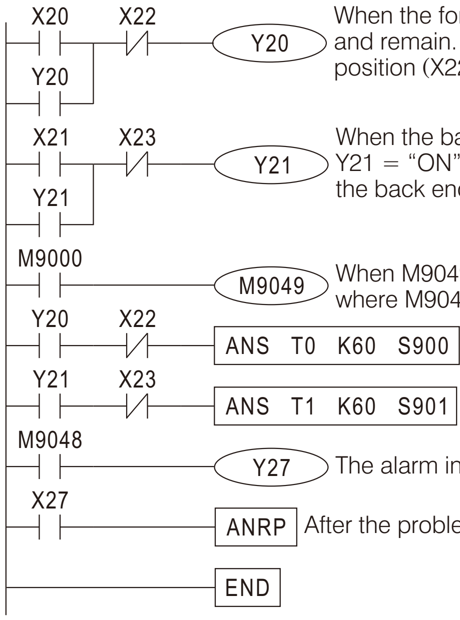
- The instruction ANR is used exclusively to reset the annunciator step relay. When each time the ANR instructinon is operated, annunciators which had been activated are sequentially reset one-by-one.
- When X0 = “OFF” → “ON”, the instruction ANR will be executed and an active annunciator will be reset to “OFF”.
- If the instruction ANR is executed and if there are more than one active annunciator, the active annunciator with smallest ID number will be reset. When the instruction ANR is executed once again, at this moment the active annunciator with the smallest ID number (that was the second smallest at the last time) will be reset. And so forth to reset other active annunciators one-by-one.

Application Example of Timed Annunciator Set

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

- The following chart is an ANS (Timed Annunciator Set) / ANR (Annunciator Reset) application

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 X27 once to reset the annunciator.

FNC 48	DSQRP (S) (D)	Square Root	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●	●		
D													●		●			

• The 16-bit instruction, S = 0~32,767 • The 32-bit instruction, S = 0~2,147,483,647

X20 — SQR D0 D1 (S) (D)

S : the source device for performing mathematical square root

D : the destination device to store the result

- This instruction performs a square root operation on the content value of device (S) and stores the result to the destination device (D).
- It performs a square root operation on the content value of D0 and stores the result at D1 when X20 = “ON”.
- For the result, only the integer part will be taken, while the decimal part will be ignored; when any decimal is ignored, M9021 = “ON”.
- The 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 the M9067 will be set “ON”.

FNC 49	DFLTP (S) (D)	BIN Integer → Binary Floating Point Format	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●			
D													●		●			

- The 16-bit instruction, S occupies 1 and D occupies 2 components
- The 32-bit instruction, S occupies 2 and D occupies 2 components

```
X0          (S)  (D)
─┤ ├───[FLT  D0  D10]
```

S : the source data to be converted

D : the destination device to store the equivalent float format value

- When X0 = "ON", it 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 the converted result is copied to the destination devices (D11, D10).

```
X1           (S)  (D)
─┤ ├───[DFLT  D2  D12]
```

- When X1 = "ON", it 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 the converted result is copied to the destination devices (D13, D12).

- There is not required to use this instruction to convert a constant K or H at a floating calculation, because the constant will convert to binary floating point format automatically when the floating calculation is in operation.

- A floating point number in the system should occupy two consecutive registers, for the storage format of a floating point number in registers, please refere to the section 2-13 "Numerical System".

- Floating point calculation example:

  Use the PLC instruction to calculate the result of circumference

  2 × 3.14 × (D0) ⟶ (D11, D10)

  BIN integer — Binary floating point format number

```
M9000
─┤ ├──┬─[DEDIV  K314  K100  D20]  3.14 → (D21, D20) Binary floating point format number
      ├─[DEMUL  K2  D20  D22]     2×3.14 → (D23, D22) Binary floating point format number
      ├─[FLT  D0  D24]            (D0) BIN integer → (D25, D24) Binary floating point format number
      └─[DEMUL  D22  D24  D10]    2×3.14×(D0) → (D11, D10) Binary floating point format number
```

MEMO

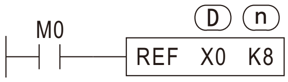
6-7 High Speed Processing Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
50	REFP (D) (n)	I/O Refresh	○	○	○	○
51	REFFP (n)	I/O Refresh and Filter Adjust	○	○	○	○
52	MTR (S) (D1) (D2) (n)	Input Matrix	○	○	○	○
53	DHSCS (S1) (S2) (D)	Software High Speed Counter Set	○	○	○	○
54	DHSCR (S1) (S2) (D)	Software High Speed Counter Reset	○	○	○	○
55	DHSZ (S1) (S2) (S) (D)	Software High Speed Counter Zone Compare	○	○	○	○
56	DSPD (S1) (S2) (D)	Speed Detection	○	○	○	○
57	DPLSY (S1) (S2) (D)	Pulse Y Output	○	○	○	○
58	PWM (S1) (S2) (D)	Pulse Width Modulation	○	○	○	○
59	DPLSR (S1) (S2) (S3) (D)	Pulse Ramp	○	○	○	○

FNC 50	REFP D n	I/O Refresh	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D	○	○																
n																○		

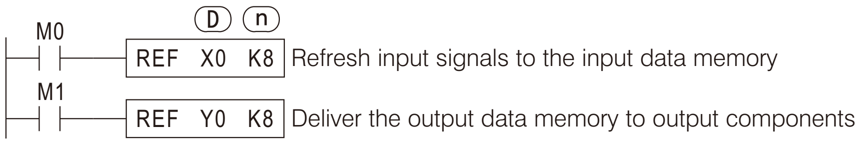
• The D should designate as X0, X10, Y0 or Y10 • When D = X0 or Y0, n = 8 or 16. • When D = X10 or Y10, n = 8.



D : the head address of I/O device to be refreshed

n : the number of I/O devices to be refreshed

- Since the PLC is by the Cyclic Operation, the CPU will read and store "ON"/ "OFF" status of all external inputs to its input data memory once at the beginning of the user program (every time when the PLC processes the step 0). Then until the END instruction is executed, all the "ON"/ "OFF" status in its output data memory will be sent to output components to drive external loadings. However, when there is a need to read "ON"/ "OFF" status of the input X during the execution of the program or to deliver the operation result to the output Y immediately, please use this REF instruction.



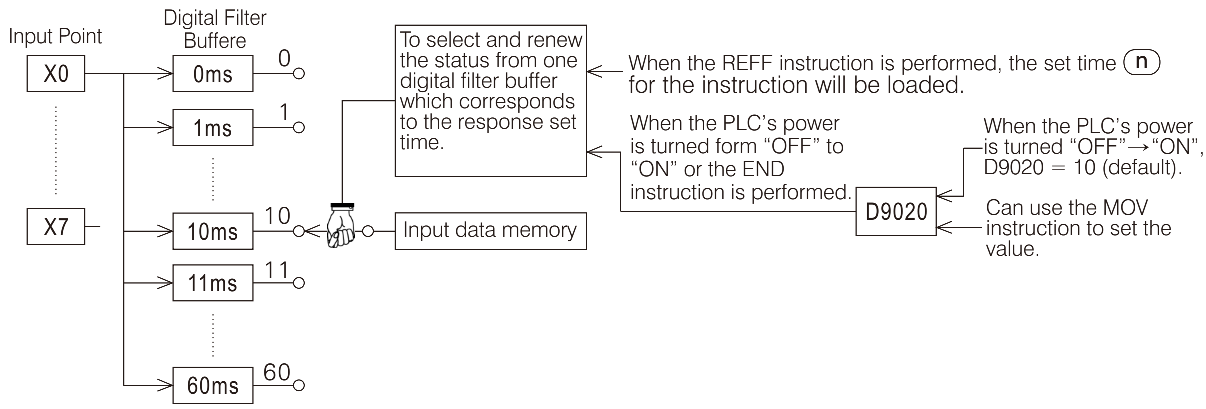
- When M0 = "ON", the input signal statuses of X0~X7 will be reloaded to PLC's status data memory. The PLC can immediately read the external statuses of X0~X7 while performing this instruction, but there is still an approximately 10ms input delay on the input circuit and filter.
- When M1 = "ON", the output statuses of Y0~Y7 will be sent to the output components from the PLC's status data memory. Therefor PLC can immediately send the statuses of Y0~Y7 while performing this instruction, but there still has some delay on the output components (e.g., relay has10ms delay approximately).
- Usually use the REF instruction in interrupt subroutines to acquire real-time input/output statuses.

FNC 51	REFF P (n)	I/O Refresh and Filter Adjust	1	2	M	3
			○	○	○	○

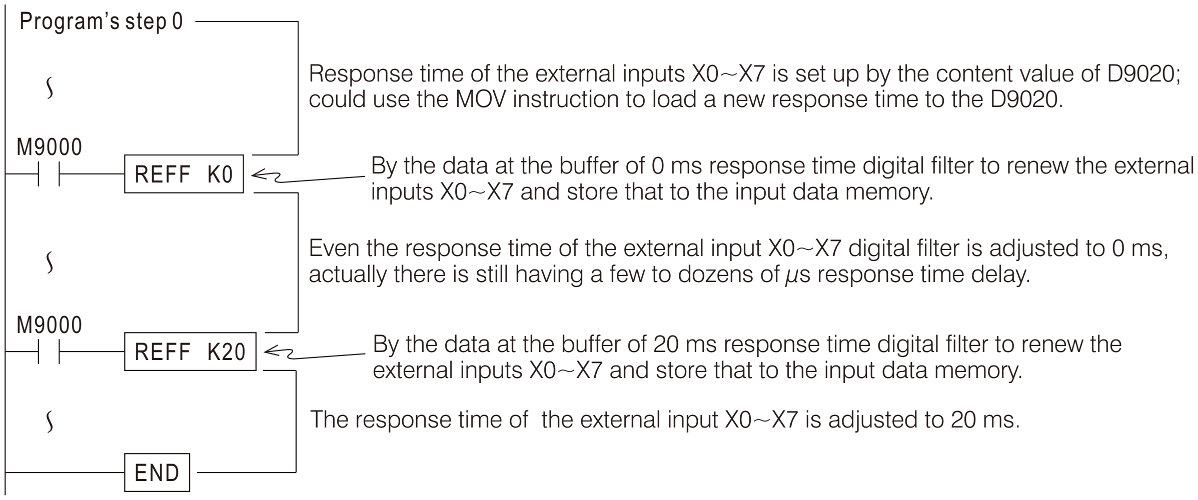
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
n													○			○		
• n = 0~60																		

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

- When X20 = "ON", response time of external inputs X0~X7 will be changed into 1ms and the "ON"/"OFF" statuses of X0~X7 will be reloaded into the input data memory.
- To avoid noise intervention, usually there will be a filter with response time of approximately 10ms on the PLC's input point to filter out noise; Therefore, it is unable to capture a rapid input signal that width is less than 10ms.
- The external inputs X0~X7 are equipped with software digital 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 external inputs X0~X7 have built-in digital filters with 0~60 ms adjustable range. The rules to set the response time of input contacts X0~X7 are described as follows:
  ① When the PLC's power is turned form "OFF" to "ON", the content value of D9020 will be set to 10 and response time will be set to 10ms.
  ② Could adjust the response time by using the MOV instruction to load a new value to the D9020.
  ③ Use the REFF instruction to adjust the response time during the program is in execution.

- 

- When the interrupt function, the high speed counter or the SPD (FNC56) instruction is used in the program, the response time of the corresponding input terminal will be automatically adjusted to 0 μs. However, there is still having a few to dozens of μs response time delay.

FNC 52	MTR S D1 D2 n	Input Matrix	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	○																	
D1		○																
D2		○	○	○														
n																○		

- S should designate to an X with the last digit of the ID is "0" (e.g. X0, X10), S occupies consecutive 8 points.
- D1 should designate to a Y with the last digit of the ID is "0"(e.g. Y0, Y10), D1 occupies consecutive n points.
- D2 should designate a Y, M or S with the last digit of the ID is "0"(e.g. Y0, M10, S20), D2 occupies consecutive (8×n) points.
- n = 2~8

```
        S    D1   D2  n
X0
|-| |----[MTR X20 Y20 M0 K2]
```

S : the head point for the matrix scan inputs

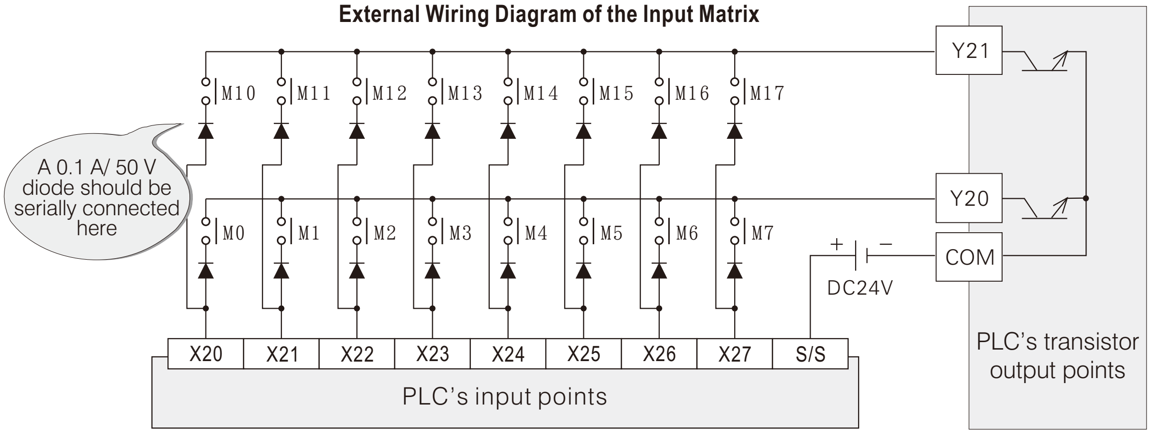
D1 : the head point for the matrix scan outputs

D2 : the head point of the result matrix-table (storage components)

n : the number of array rows of the matrix scan

- This instruction reads the status of switches through the matrix scan technique. It needs to use consecutive 8 inputs (which are headed by S) and n outputs (which are headed by D1). By the sequential outputs to scan and read the status of inputs, then the result is reflected in the storage components which are headed by D2. Therefore, total 8× n of the external "ON"/"OFF" status are received.

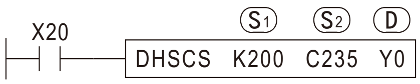
External Wiring Diagram of the Input Matrix



- From the diagram above, X20~X27 and Y20~Y21 constitute two rows of array matrix input circuit. When X0 = "ON", the instruction is ready for execution then 16 (8×2) "ON"/ "OFF" statuses at the array matrix will be read and stored in the components of M0~M7 and M10~M17.
- When X0 = "OFF", the instruction is inactive and the statuses of M0~M7 and M10~M17 will remain as same as the statuses before it is disabled.
- Using the MTR instruction to read one row of the external array switches will take two Scan Times. If the PLC's Scan Time is less than 10ms, then one row in the array will spend 20ms to read the "ON"/ "OFF" status of external inputs. Since this instruction can connect at most 8 of array rows to read 64 (8×8 = 64) external switches, to reload that once will take 16 Scan Time or 160ms. Therefore, the coordination between the response rate of external switches and the loading time of the instruction should be considered when this instruction is used.
- Usually, the instruction's conditional contact uses the M9000 (permanently "ON", "a" contact).
- When this instruction performs to scan and reload status once, it will let the Execution Completed Flag M9029 = "ON" for one Scan Time.
- The MTR instruction can only be used once in the program and should appoint to a transistor output unit.

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2												●						
D		●	●	●	○	○												

• S2 = C235~C255 • D can also designate to IHC0~IHC7



S1 : the set value of comparison

S2 : the ID No. of the software high speed counter to be used

D : the destination of the comparison result output

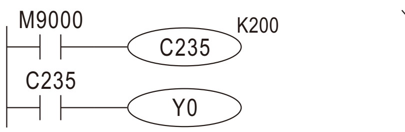
- The DHSCS instruction is used to give an immediate output that is by way of a Softwear High Speed Counter (SHSC). The SHSC receives a pulse stream from corresponding external input and counts by using the interrupt function (for more details about the SHSC, please refer to the section "2-7 Software High Speed Counter").

  When a SHSC is selected by the DHSCS instruction's (S2) and the Present Value of the SHSC changes (increased/decreased by "1") by the input, the DHSCS instruction will immediately perform the "Compare" operation. If the Present Value of the SHSC is equal to the Set Value of comparison (S1), the status of result output (D) will turn "ON". After that, the (D) remains the same status ("ON") even the comparison is not equal.

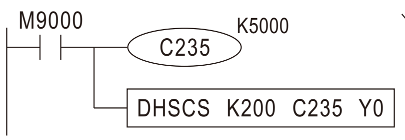
  Generally, the (D) in this instruction is designed to an output point Y0~Y17. When the output point Y0~Y17. is designated, the status "ON" will be carried out immediately to that output component. On the other hand, if the assigned destination is not between Y0~Y17, the comparison result is only effecting to the state of the designated component in the PLC's data memory.

- As the example above, the DHSCS will be enabled if X20 = "ON". When the Present Value of C235 changes from 199 to 200 or from 201 to 200, the status of Y0 becomes "ON" and immediately sent it to the external output point, then the Y0 will remain its "ON" status.

Common Output V.S. DHSCS Instruction Output



The action of the external output Y0 will be affected to the PLC's Scan Time. Therefore, its response is slower.

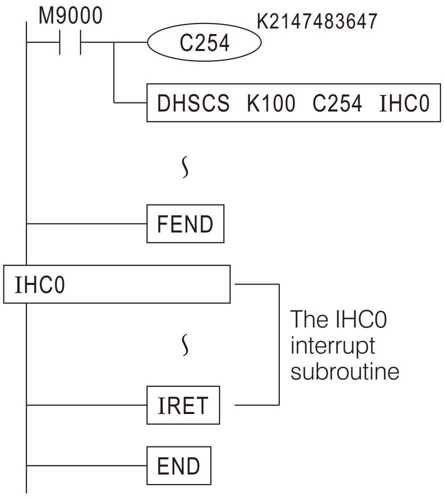


The instruction is by the function of interrupt so the status of Y0 is immediately output to external output, regardless of the PLC Scan Time. Please notice that: There is still an output delay from the relay or transistor output component.

Note

- This instruction is a 32-bit instruction; the DHSCR should be always entered when uses this instruction.
- There is no limitation on the number of using times about the DHSCS, DHSCR and DHSZ instructions, while the DHSCT can only be used once in the program. However, the total of these instructions performed at the same time should not exceed "8".
- Both the Software High Speed Counter and the DHSCS, DHSCR, DHSZ, DHSCT instructions are only performed to compare when there is an external signal to the input of the counter. The compare action shall not be performed if the Present Value of Software High Speed Counter is changed by a transferring instructions, because there is no counter input signals. Thus, there is not have the compare action, of course not has the compare output.

Software High Speed Counter Interrupt



- The comparison result output (D) of the DHSCS instruction can also designate to an Interrupt Pointer of the Software High Speed Counter that serviceable name is IHC0~IHC7. By way of the pointer to perform the interrupt subroutine immediately.
- In the example, when the Present Value of C254 = 100, CPU will jump to the interrupt Pointer IHC0 once to perform the interrupt subroutine.
- When the M9061 = "ON", the Software High Speed Counter interrupt IHC0~IHC7, will be prohibited.

FNC 54	DHSCR S1 S2 D	Software High Speed Counter Reset	1	2	M	3
			○	○	○	○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2												●						
D		●	●	●	○	○						●						

• S2 = C235~C255

• D can also designate to a Software High Speed Counter but the D and S2 must use the same SHSC

```
X20                  S1    S2    D
 |--| |------[ DHSCR K200  C235  Y0 ]
```

S1 : the set value of comparison

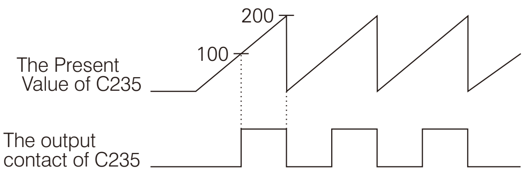
S2 : the ID No. of the software high speed counter to be used

D: the destination will be reset by the compare result

- The DHSCR instruction is used to give an immediate reset output that is by way of a Software High Speed Counter (SHSC).
- When the D is designated to an output point Y0~Y17, the compare result to reset the point will be carried out immediately to that output component. On the other hand, if the assigned destination is not between Y0~Y17, the comparison result is only effecting to the state of the designated component in the PLC's data memory.
- When X20 = "ON" and the Present Value of C235 changes from 199 to 200 or from 201 to 200, the status of Y0 becomes "OFF" and immediately sent it to the external output point, then the Y0 will remain its "OFF" status.

The Counter Self-Reset

```
 M9000                K100
 |--| |---+----( C235 )
          |
          +----[ DHSCR K200 C235 C235 ]
```



- When the S2 and D of the DHSCR instruction are designated to a same Software High Speed Counter, the counter will perform the action of Self-Reset if its input makes the Present Value equal to the S1 .
- When the counted Present Value of C235 = 200, the C235 will be reset immediately (its Present Value will be reset to "0" and the output contact will become "OFF").

Note

- This instruction is a 32-bit instruction; the DHSCR should be always entered when uses this instruction.
- There is no limitation on the number of using times about the DHSCS, DHSCR and DHSZ instructions, while the DHSCT can only be used once in the program. However, the total of these instructions performed at the same time should not exceed "8".
- Both the Software High Speed Counter and the DHSCS, DHSCR, DHSZ, DHSCT instructions are only performed to compare when there is an external signal to the input of the counter. The compare action shall not be performed if the Present Value of Software High Speed Counter is changed by a transferring instructions, because there is no counter input signals. Thus, there is not have the compare action, of course not has the compare result.

FNC 55	DHSZ (S1) (S2) (S) (D)	Software High Speed Counter Zone Compare	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
S												●						
D		●	●	●	○	○												

• S = C235~C255 • D occupies 3 consecutive components, should use Y0~Y5 if the D designates to a Y

```
                    (S1)   (S2)   (S)   (D)
 M9000
|—| |————————[DHSZ  K100   K200   C255  Y0]
```

S1 : the set value of the lower limit at the zone compare

S2 : the set value of the upper limit at the zone compare

S : the ID No. of the software high speed counter to be used

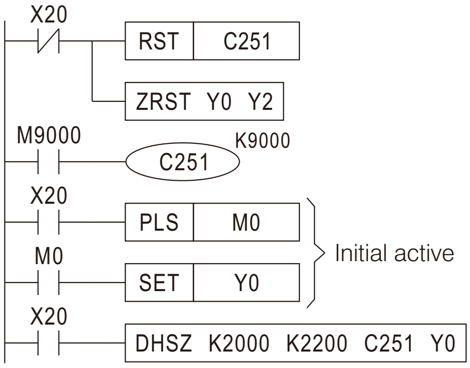
D : the destination of the comparison result outputs

- All of the counting value and result outputs of this instruction are processed with interrupt insertion, the Y0~Y2 will output immediately and irrespective to the Scan Time. The results of the zone comparison are shown as follows:
  When K100 > the Present Value of C255, then Y0 = "ON".
  When K100 ≤ the Present Value of C255 ≤ K200, then Y1 = "ON".
  When K200 < the Present Value of C255, then Y2 = "ON".
- If (S1) > (S2) , then the value of (S1) will become both of the Upper/Lower limits to be compared with the (S).

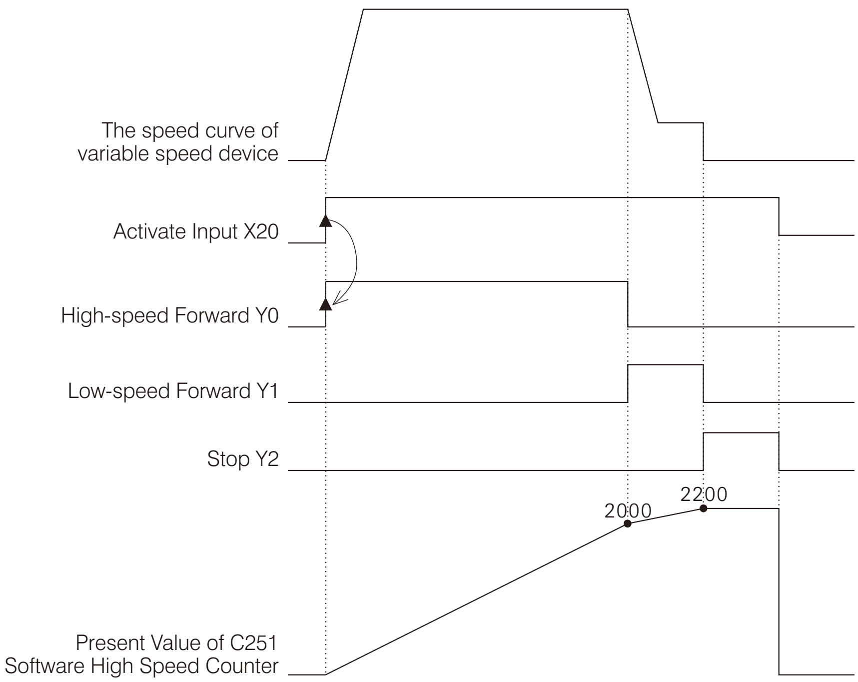
Note

- This instruction is a 32-bit instruction; the DHSZ should be always entered when uses this instruction.
- There is no limitation on the number of using times about the DHSCS, DHSCR and DHSZ instructions, while the DHSCT can only be used once in the program. However, the total of these instructions performed at the same time should not exceed "8".
- Both the Software High Speed Counter and the DHSCS, DHSCR, DHSZ, DHSCT instructions are only performed to compare when there is an external signal to the input of the counter. The compare action shall not be performed if the Present Value of Software High Speed Counter is changed by a transferring instructions, because there is no counter input signals. Thus, there is not have the compare action, of course not has the compare outputs.

Use the HSZ instruction to perform high speed, low speed and 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 the signal input point for activation.
- The DHSZ will generate the outputs only when the external pulses are entering to the C251. For that reason, when X20 = "OFF" → "ON" (the initial active signal of the left program), it must to activate the motor operation (let Y0 = "ON"). Afterwards, the move will produce the pulses to the Software High Speed Counter, so the instruction can perform the corresponding comparison and use the results to drive the outputs Y0~Y2 immediately.



FNC 56	DSPD (S1) (S2) (D)	Speed Detection	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	●																	
S2							●	●	●	●	●	●	●	○	●	●		
D											●	●	●	○				

• S_1 = X0~X7 • D occupies 3 consecutive components

```
X20          (S1) (S2)  (D)
|-| |-------[SPD  X0  K1000  D0]
```

S_1 : the external pulse input point

S_2 : the sampling time for to count pulses (unit: ms)

D : the detection result

- Within the designated (S2) sampling time (unit = ms), this instruction calculates the number of pulses coming from the external (S1) input and stores the result into the register (D) .
- When X20 = "ON", the instruction started up and uses the register D1 to accumulate the number of input pulses from the external input point X0. After 1000ms a period of the sampling time, it stores the D1's accumulated result to D0 then clears the current count value of D1 as "0". And once again, it recalculates the number of input pulses from the X0.
- The D2 displays the remaining time of countdown timer (unit = ms).
- The main purpose of the instruction is to acquire the rotation rate of the rotation device. The rotation rate can be easily acquired from using the content value of D0 with the following equation:

$$N = \frac{60 \times \text{(the count value of D0)}}{n \times t} \times 10^3 \text{ (rpm)}$$

N : Rotation rate

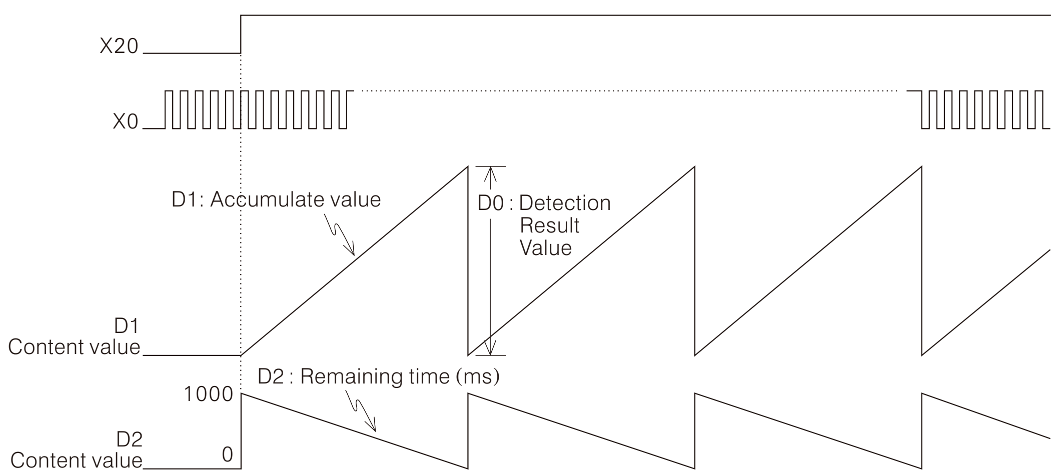
n : Number of pulses is generated by each turn of the rotation device

t : Content value of (S2)

As in the equation referred above, suppose that n = 100 and (D0) = 3000, then we will have

$$N = \frac{60 \times 3000}{100 \times 1000 \text{(ms)}} \times 10^3 = 1800 \text{ (rpm)}$$

- The exterior input designated by the (S1) in the instruction cannot be reused for any other high speed input function.
- Since this instruction is by the Software High Speed Counter to get the pulse, please refer to the section "2-15 The X0~X7 High Speed Input Function Description" to get the specification about external input.



FNC 57	DPLSY S1 S2 D	Pulse Y Output	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D		●																

• VS1 / VS2 series S1 = 1~50,000 • VS2 / VS3 series S1 = 1~200,000 • VSM-28ML S1 = 1~1,000,000

• 16-bit instruction S2 = 1~32,767 • 32-bit instruction S2 = 1~2,147,483,647 • D = Y0~Y3

```
X20              S1    S2    D
─┤ ├─────[PLSY  K500  D100  Y0]
```

S1 : the frequency of pulses to be output

S2 : the number of pulses to be output

D : the pulse output point

- When X20 = "ON", the Y0 uses 500 Hz frequency rate (i.e. 500 pulses per second) to produce the specified quantity of pulses that number is assigned by the content value of D100.

  S1 designates the output pulse frequency the available range of the PLC:
  The VS1 or VS2 series is from 1 to 50,000Hz; VSM or VS3 series is from 1 to 200,000Hz; VSM-28ML is from 1 to 1,000,000Hz.
  If the value of S1 is less than 1, regards it as 1.
  If the value of S1 is exceeding the upper limit of the range, regards it as the maximum value.

  S2 designates the number of output pulses, the available range is 0~2,147,483,647.
  If the value of S2 is equal to 0, the quantity of pulses is unlimited for continuous outputs.
  If the value of S2 is less than 0, the PLC will regard that as an operating error and the M9067 will be "ON".

  D designates the pulse output point. It can only designate to the point Y0~Y3 to be the pulse output device and must use a transistor main unit.

- When the conditional contact X20 becomes "OFF" during the pulse output, pulse outputs will be stopped and the pulse output point will also turns "OFF"; When the X20 becomes "ON" again, it will restore to generate pulses from the first one.

- Since the pulse output function at the Y0~Y3 are the same, below we use the Y0 as an example to describe the related special devices.
  The M9340 displays the status of the Y0 axis.
  Any instruction which is related to the axis Y0 and is executed successfully, will continuously occupy the axis Y0, until the condition contact becomes "OFF".
  Therefore, before to activate this instruction, should pay attention to the status of the M9340.
  If the M9340 is already "ON", it indicates that axis Y0 is occupied by other relevant instruction to generate pulse or positioning. So, this instruction will not be executed even though the condition contact is "ON".
  When the M9340 is "OFF", it indicates that axis Y0 is READY to accept and execute this instruction.

  The M9341 is a pulse output monitor flag of the axis Y0.
  When the axis Y0 is outputting pulse, the M9341 turns "ON".

  The M9342 is a positioning completed flag of the axis Y0.
  When the number of pulse outputs assigned by S2 is arrived, the M9342 turns "ON".
  When the condition contact is "OFF", the M9342 will also turn "OFF".

  The M9343 is a flag for abnormal stop at the axis Y0.
  The stop flag turns "ON" at the moment that the Y0 is generating pulses and the number of pulses assigned by S2 is not arrived yet, the pulse outputting will be stopped promptly. The M9343 will turn "ON" to indicate the stop is abnormal.
  When the condition contact is "OFF", the M9343 will also turn "OFF".

  The M9345 or M9346 is the stop signal of the axis Y0.
  When either one of the signals is "ON", the Y0's pulse outputting at this instruction will be stopped promptly. After the stop signal is clear, this instruction must be reactivated to enable pulse outputting again.

  The (D9351, D9350) display the current speed of axis Y0.
  When the instruction is executed, that will display the current pulse frequency of the axis Y0's output.

  The (D9353, D9352) display the number of pulses that the axis Y0 has been output.
  When the power switches "ON", the initial value is 0. Afterwards, that will display the accumulated number of the axis Y0's output pulses.

• The related special devices are summarized below: (■ : Means read only.)

Relay ID No.	Description
■M9340	Y0 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y0 has been occupying.
■M9341	Y0 axis's pulse output monitor. "ON" means pulse is generating.
■M9342	Y0 axis's positioning completed flag.
■M9343	Y0 axis's positioning abnormal stop flag.
M9345	Y0 axis's stop flag (with gradually slow down).
M9346	Y0 axis's immediately stop flag.
■M9360	Y1 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y1 has been occupying.
■M9361	Y1 axis's pulse output monitor. "ON" means pulse is generating.
■M9362	Y1 axis's positioning completed flag.
■M9363	Y1 axis's positioning abnormal stop flag.
M9365	Y1 axis's stop flag (with gradually slow down).
M9366	Y1 axis's immediately stop flag.
■M9380	Y2 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y2 has been occupying.
■M9381	Y2 axis's pulse output monitor. "ON" means pulse is generating.
■M9382	Y2 axis's positioning completed flag.
■M9383	Y2 axis's positioning abnormal stop flag.
M9385	Y2 axis's stop flag (with gradually slow down).
M9386	Y2 axis's immediately stop flag.
■M9400	Y3 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y3 has been occupying.
■M9401	Y3 axis's pulse output monitor. "ON" means pulse is generating.
■M9402	Y3 axis's positioning completed flag.
■M9403	Y3 axis's positioning abnormal stop flag.
M9405	Y3 axis's stop flag (with gradually slow down).
M9406	Y3 axis's immediately stop flag.

Register ID No.	Description	
■D9350	Lower 16 bits	The Y0's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9351	Upper 16 bits	
D9352	Lower 16 bits	The Y0's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9353	Upper 16 bits	
■D9370	Lower 16 bits	The Y1's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9371	Upper 16 bits	
D9372	Lower 16 bits	The Y1's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9373	Upper 16 bits	
■D9390	Lower 16 bits	The Y2's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9391	Upper 16 bits	
D9392	Lower 16 bits	The Y2's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9393	Upper 16 bits	
■D9410	Lower 16 bits	The Y3's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9411	Upper 16 bits	
D9412	Lower 16 bits	The Y3's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9413	Upper 16 bits	

• During this instruction is being executed, any modify to the content value of S2 will be treated as invalid, but the value of S1 can be altered by the program to change the frequency.

• There is no limitation on the using number of this instruction in a program, and the output points Y0~Y3 can generate pulses at the same time.

FNC 58	PWM (S1) (S2) (D)	Pulse Width Modulation	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D		●																

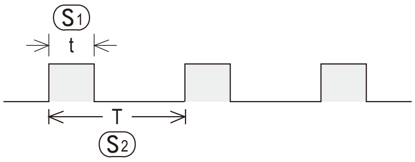
• S_1 = 0~32767 • S_2 = 1~32767 • D = Y0~Y3

S_1 : the width of "ON" at an output pulse, t = 0~32,767 × time base

S_2 : the entire period of an output pulse, T = 1~32,767 × time base

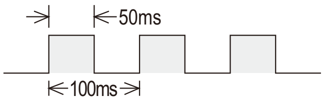
D : the pulse output point

- The PWM instruction is operated to generate pulse width modulation, the t / T time ratio characteristics of the diagram which is shown below.
  Besides, the time base is defined by the M9035.
  When the M9035 = "OFF", the time base is 1ms; otherwise when the M9035 = "ON" the time base is 0.1ms.

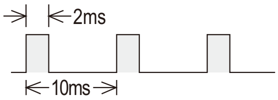


- When the conditional contact is "ON", the consecutive pulses will be generated at the designated component (D). That each one of the entire period with a "T" cycle period (designated by (S2)) and the "t" width of "ON" duration (designated by (S1)).

- When X20 = "ON", M9035 = "OFF" and suppose the D0 = 50, then Y0 will output the following pulses.



- When X20 = "ON", M9035 = "ON" and suppose the D0 = 20, then Y0 will output the following pulses.



- If X20 becomes "OFF", Y0 will also become "OFF".
- If the value of "t" is larger than the "T", the PLC will regard that as an operational error.
- Each output point of Y0~Y3 in the program can only be applied once for the PWM instruction. That is, two PWM instructions cannot assign the same Y output point in the program.
- The specified pulse output point of this instruction cannot be reused to the output which is specified by the PLSY, PLSR or any other positioning instruction.
- The related special devices are summarized below:

Relay ID No.	Description
M9035	Assign the PWM operating time base. When M9035="OFF", the parameters for the PWM are by the unit of 1ms.; when M9035= "ON", by the unit of 0.1ms.

FNC 59	D PLSR (S1) (S2) (S3) (D)	Pulse Ramp	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
S3							●	●	●	●	●	●	●	○	●	●		
D		●																

• VS1 / VS2 series S_1 = 1~50,000 • VS2 / VS3 series S_1 = 1~200,000 • VSM-28ML S_1 = 1~1,000,000

• S_2 = 1~2,147,483,647 • S_3 = 0~5,000 • D = Y0~Y3

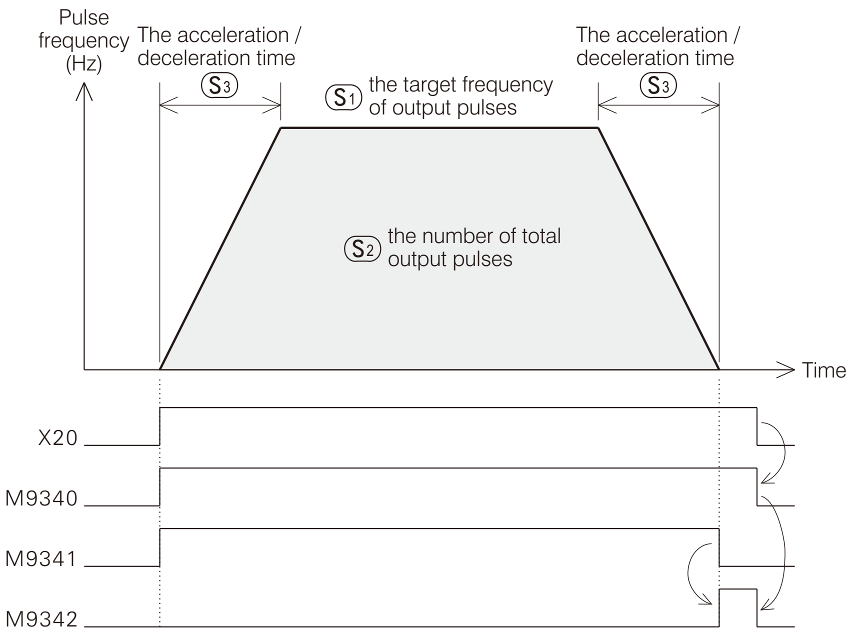
```
          (S1)   (S2)  (S3)   (D)
 X20
─┤├───[ PLSR  K1000  D100  K1000  Y0 ]
```

S_1 : the target frequency of output pulses

S_2 : the number of total output pulses

S_3 : the ramp time for acceleration or deceleration (unit: ms)

D : the pulse output point



• When X20 = "ON", Y0 will output the specified pulses by the content value of D100 as the diagram above.

(S1) designates the target frequency of output pulses.
At the VS1 & VS2 series, the available target output frequency is from 1 to 50,000Hz.
At the VSM & VS3 series, the available target output frequency is from 1 to 200,000Hz.
At the VSM-28ML, the available target output frequency is from 1 to 1,000,000Hz.
When the value of (S1) is less than 1, it will be treated as 1.
When the value of (S1) is larger than the range, it will be treated as the maximum value.

(S2) designates the number of total output pulses.
It can specify the range from 1 to 2,147,483,647 pulses.
When the value of (S2) is less than or equal to 0, the PLC will regard it as an operating error and the M9067 will be "ON".

(S3) designates the ramp time for acceleration or deceleration (unit = ms).
The available range is from 0 to 32,000. If the value of (S3) is less than 0, it will be treated as 0; if the value of (S3) is larger than 32,000, it will be treated as 32,000.

(D) designates the pulse output point.
It can only designate to the point Y0~Y3 to be the pulse output device and must use a transistor main unit.

• When the condition contact X20 turns "OFF" during pulse output, the output frequency will gradually slow down then stop.

• During this instruction is being executed, the content value of (S1) can be altered by the program to change the frequency; Changes in other parameters will be treated as invalid.

- There is no limitation on the using number of this instruction in a program, and the different output points Y0~Y3 can generate pulses at the same time.

- Since the pulse output function of Y0~Y3 are the same, below we use the Y0 as an example to describe the related special devices.

  The M9340 displays the status of the Y0 axis.

  Any instruction which is related to the axis Y0 and is executed successfully, will continuously occupy the axis Y0, until the condition contact becomes "OFF".
  Therefore, before to activate this instruction, should pay attention to the status of the M9340.

  If the M9340 is already "ON", it indicates that axis Y0 is occupied by other relevant instruction to generate pulse or positioning. So, this instruction will not be executed even though the condition contact is "ON".

  When the M9340 is "OFF", it indicates that axis Y0 is READY to accept and execute this instruction.

  The M9341 is a pulse output monitor flag of the axis Y0.

  When the axis Y0 is outputting pulse, the M9341 turns "ON".

  The M9342 is a positioning completed flag of the axis Y0.

  When the number of pulse outputs assigned by S2 is arrived, the M9342 turns "ON".

  When the condition contact is "OFF", the M9342 will also turn "OFF".

  The M9343 is a flag for abnormal stop at the axis Y0.

  The stop flag turns "ON" at the moment that the Y0 is generating pulses and the number of pulses assigned by S2 is not arrived yet, the pulse outputting will be stopped promptly. The M9343 will turn "ON" to indicate the stop is abnormal.
  When the condition contact is "OFF", the M9343 will also turn "OFF".

  The M9345 is the stop signal of the axis Y0.

  When the signal is "ON", the pulse output frequency of this instruction at Y0 will slow down then stop After the stop signal is clear, this instruction must be reactivated to enable pulse outputting again.

  The M9346 is the immediately stop signal of the axis Y0.

  When the signal is "ON", the Y0's pulse outputting will be stopped promptly. After the stop signal is clear, this instruction must be reactivated to enable pulse outputting again.

  The (D9351, D9350) display the current speed of axis Y0.

  When the instruction is executed, that will display the current pulse frequency of the axis Y0's output.

  The (D9353, D9352) display the number of pulses that the axis Y0 has been output.

  When the power switches "ON", the initial value is 0. Afterwards, that will display the accumulated number of the axis Y0's output pulses.

• The related special devices are summarized below: (■ : Means read only.)

Relay ID No.	Description
■M9340	Y0 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y0 has been occupying.
■M9341	Y0 axis's pulse output monitor. "ON" means pulse is generating.
■M9342	Y0 axis's positioning completed flag.
■M9343	Y0 axis's positioning abnormal stop flag.
M9345	Y0 axis's stop flag (with gradually slow down).
M9346	Y0 axis's immediately stop flag.
■M9360	Y1 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y1 has been occupying.
■M9361	Y1 axis's pulse output monitor. "ON" means pulse is generating.
■M9362	Y1 axis's positioning completed flag.
■M9363	Y1 axis's positioning abnormal stop flag.
M9365	Y1 axis's stop flag (with gradually slow down).
M9366	Y1 axis's immediately stop flag.
■M9380	Y2 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y2 has been occupying.
■M9381	Y2 axis's pulse output monitor. "ON" means pulse is generating.
■M9382	Y2 axis's positioning completed flag.
■M9383	Y2 axis's positioning abnormal stop flag.
M9385	Y2 axis's stop flag (with gradually slow down).
M9386	Y2 axis's immediately stop flag.
■M9400	Y3 axis's status. "OFF" means the Y0 is in the READY status, it is available for a positioning instruction; while "ON" = BUSY, the Y3 has been occupying.
■M9401	Y3 axis's pulse output monitor. "ON" means pulse is generating.
■M9402	Y3 axis's positioning completed flag.
■M9403	Y3 axis's positioning abnormal stop flag.
M9405	Y3 axis's stop flag (with gradually slow down).
M9406	Y3 axis's immediately stop flag.

Register ID No.		Description
■D9350	Lower 16 bits	The Y0's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9351	Upper 16 bits	
D9352	Lower 16 bits	The Y0's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9353	Upper 16 bits	
■D9370	Lower 16 bits	The Y1's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9371	Upper 16 bits	
D9372	Lower 16 bits	The Y1's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9373	Upper 16 bits	
■D9390	Lower 16 bits	The Y2's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9391	Upper 16 bits	
D9392	Lower 16 bits	The Y2's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9393	Upper 16 bits	
■D9410	Lower 16 bits	The Y3's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).
■D9411	Upper 16 bits	
D9412	Lower 16 bits	The Y3's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.
D9413	Upper 16 bits	

6-8 Handy Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
61	D SER P S1 S2 D n	Search a Data Stack	○	○	○	○
62	D ABSD S1 S2 D n	Absolute Drum Sequencer	○	○	○	○
63	INCD S1 S2 D n	Incremental Drum Sequencer	○	○	○	○
64	TTMR D n	Teaching Timer	○	○	○	○
65	STMR S m D	Special Timer	○	○	○	○
66	ALT P D	Alternate State	○	○	○	○
67	RAMP S1 S2 D n	Ramp Variable Value	○	○	○	○
69	SORT S m1 m2 D n	Sort Tabulated Data	○	○	○	○

FNC 61	DSERP (S1) (S2) (D) (n)	Search a Data Stack	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●			
S2							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●		●			
n													○			○		

- The 16-bit instruction, n = 1~256, S1 occupies n components, D occupies 5 components.
- The 32-bit instruction, n = 1~128, S1 occupies (2×n) components, D occupies 10 components

```
         (S1) (S2) (D)  (n)
X20
-| |----[SER  D0  D10  D20  K10]
```

S1 : the head device ID number of a defined data stack to be searched
S2 : the compare data source at the searching
D : the search result's storage head device ID number
n : the search stack length of the searched data

- The data stack is assigned by (n) consecutive devices which headed with (S1) . This instruction will compare the content value at the specified (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", this instruction uses the D10 to compare with the stack D0~D9 and store the result into D20~D24. (Assune the content value of parameter D10 = 100.)

Data Position Number	Data Stack for Searching	Content Value of D0 ~ D9	Compare Data Source	Compare Decision
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)

The result of the search

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

- The content value at the (D)+3 and (D)+4 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.

FNC 62	D ABSD (S1) (S2) (D) (n)	Absolute Drum Sequencer	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●			
S2												●						
D		●	●	●	○	○												
n																○		

- The 16-bit instruction, S_1 occupies (2×n) components, D occupies n components.
- When S_1 designates to K*n*X, K*n*Y, K*n*M or K*n*S, that K*n* must be K4 and the ID number of the X, Y, M or S must be a multiples of 8

- The 32-bit instruction, S_1 occupies (4×n) components, D occupies (2×n) components.
- When S1 designates to K*n*X, K*n*Y, K*n*M or K*n*S, that K*n* must be K8 and the ID number of the X, Y, M or S must be a multiples of 8

- The 16-bit instruction, S_2 = C0~C199 • The 32-bit instruction, S_2 = C200~C255 • n = 1~64

```
        X20        (S1) (S2) (D) (n)
|---| |-------[ ABSD  D0   C0  M0  K4 ]
```

S_1 : the head device ID number of the comparison table

S_2 : the ID number of the counter for the comparison

D : the number device ID number of the comparison result

n : the number of comparison section groups

- The instruction is a multi-section compare instruction and generally is operated for multi-section absolute drum sequencer.

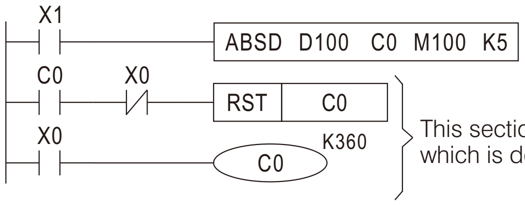
	Lower Limit	Upper Limit	Comparison Value	Comparison Result
(n)	(S1) D0 = 50	D1 = 200	(S2) C0 = 100	(D) M0 = 1
	D2 = 0	D3 = 50		M1 = 0
	D4 = 80	D5 = 120		M2 = 1
	D6 = 120	D7 = 300		M3 = 0

- When X20 = "ON", the Present Value of the selected counter C0 is compared against a user defined data table [(D0, D1), (D2, D3), (D4, D5) and (D6, D7), there are 4 groups of upper/lower limit], and the results are stored on M0~M3 respectively.
  At each group, if the condition is met [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 "ON"/ "OFF" statuses at M0~M3 will remain as same as before it is disabled.

Program Example

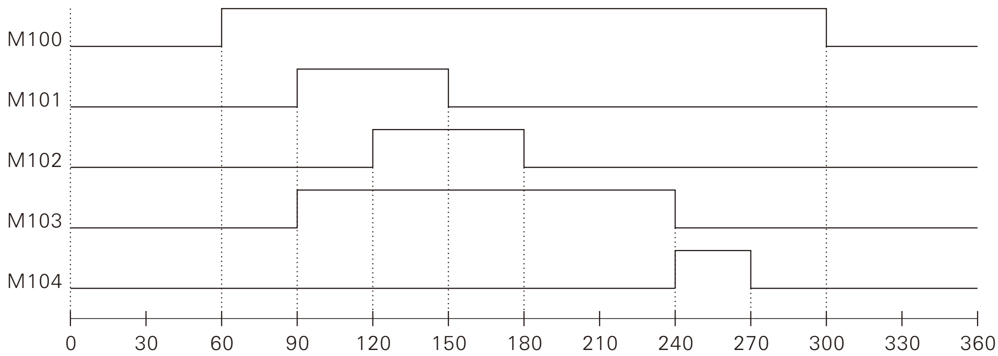
Suppose that a drum-controlled rotor sends a pulse to the input terminal X0 when it rotates by one degree, then the following program will perform the checkup and control actions of the drum degree.



When X1 = "ON", it activate the multi-section compare

This section of program that enables the C0 to be the counter of rotor which is depending on counting the rotation degree of the rotor.

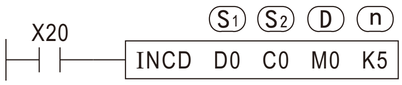
Lower Limit	Upper Limit	Comparison Value	Comparison Result
D100 = 60	D101 = 300	C0	M100
D102 = 90	D103 = 150		M101
D104 = 120	D105 = 180		M102
D106 = 90	D107 = 240		M103
D108 = 240	D109 = 270		M104



FNC 63	INCD (S1) (S2) (D) (n)	Incremental Drum Sequencer	1	2	M	3
			○	○	○	○

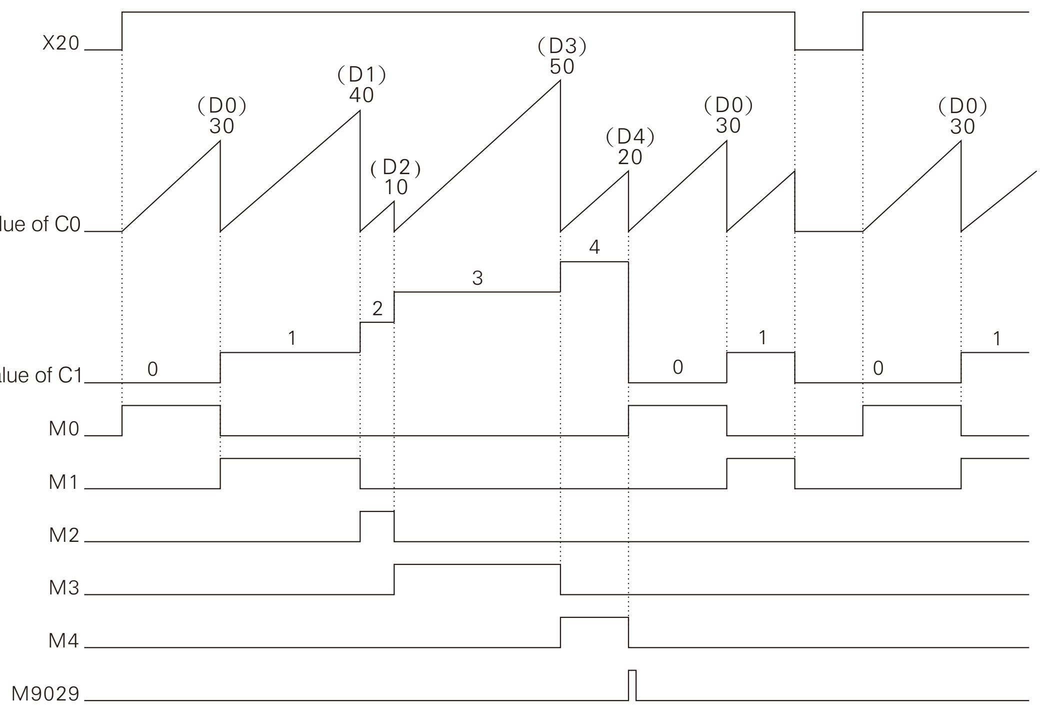
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●			
S2												●						
D		●	●	●	○	○												
n																○		

- S1 occupies n components • S2 = C0~C198 • D occupies n components • n = 1~64
- When S1 designates to K*n*X, K*n*Y, K*n*M or K*n*S, that K*n* must be K4 and the ID number of the X, Y, M or S must be a multiples of 8



S1 : the head device ID number of the comparison table
S2 : the ID number of the counter for the comparison
D : the head device ID number of the comparison result
n : the number of comparison section groups

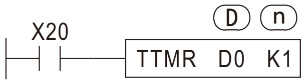
- The INCD instruction is a multi-section incremental drum-controlled instruction.
- If the (S1) is designated to the D0, (n) = 5 and (D) is designated to the M0, then the comparison values are stored in the table D0~D4, while M0~M4 are served as the outputs. Suppose that (D0) = 30, (D1) = 40, (D2) = 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, the instruction's Execution Completed Flag M9029 will turn "ON" for a Scan Time while a circulation is completed.
- When X20 = "ON" → "OFF", the Present Values of C0 and C1 will be reset to "0" and M0~M4 will be turned "OFF".

FNC 64	TTMR D n	Teaching Timer	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D													●					
n													○			○		
• D occupies 2 components • n = 0~2																		

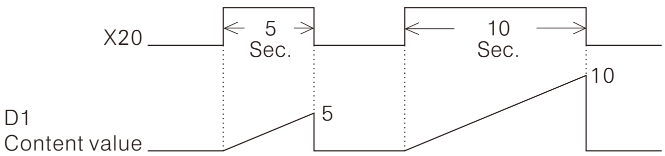


D : the ID number of the register which can store the timed data of “ON” duration (e.g., From a push button switch)

n : the setting for multiplier

- Suppose X20 is the external push button switch.

- When X20 = “ON” (it is pressed), the content value of D1 (by the unit of second) will respond to the duration of X20 = “ON”.
  If X20 =“ON” for 5 seconds, then D1 = 5. Because of the n = 1, the D0 = 50.
  If X20 =“ON” for 10 seconds, then D1 = 10. Because of the n = 1, the 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 content value of D1 will be reset to “0” but the content value of D0 will remain as same as the value before it is disabled.

FNC 65	STMR (S) (m) (D)	Special Timer	1	2	M	3
			○	○	○	○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●							
m													○			○		
D		●	●	●	○	○												

(Operand columns above are all under the heading "Devices".)

• S = T0~T199 • m = 1~32,767 • D occupies 4 components

X20 — STMR T0 K20 Y20 (operands: S m D)

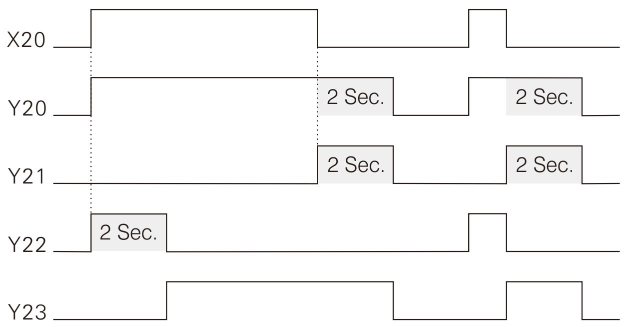
S : the ID number of designated Timer

m : the setting value of the Timer (unit = 100ms)

D : the head ID number of the output device

• The STMR instruction is operated exclusively to produce the OFF-delay, the delay trigger and a flashing circuit.

• When X20 = "ON", the STMR instruction starts to be performed. Since the (m) = 20, the T0 become a 2 seconds setting value Timer.



• The Y20 is an OFF-delay output.

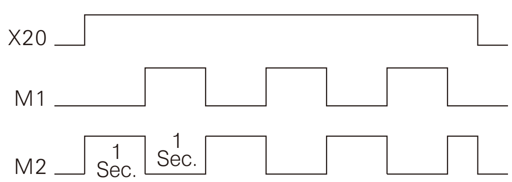
• The Y21 will have one delay output when that input signal turned from "ON" to "OFF".

• The Y22 and Y23 are designed for output signals those are exclusively to compose the flashing circuit. The following example is an applied method for the flashing circuit.

• In the program, do not reuse the Timer ID number which has been used by this instruction.

Flashing Circuit

X20 M3 — STMR T0 K10 M0



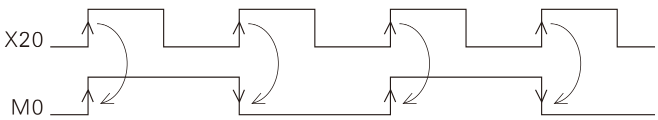
• Perform a serial link "b, NC" contact of M3 after X20, then the M1 and M2 will perform the flashing circuit.

FNC 66	ALTP D	Alternate State	1	2	M	3
			○	○	○	○

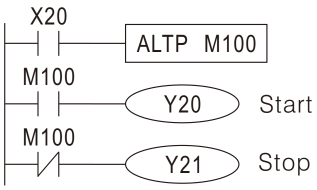
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D		●	●	●	○	○												

D : the 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".

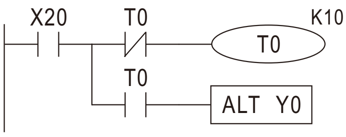


Control Start / Stop by One Push Button

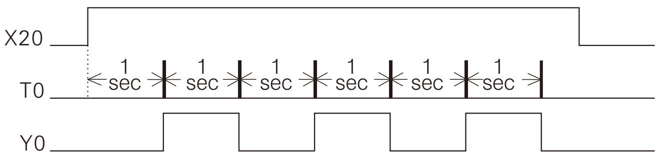


- When X20 = "OFF" → "ON" for the first time, M100 = "ON"; Y20 = "ON", Y21 = "OFF" and the operation will be activated.
- When X20 = "OFF" → "ON" for the second time, M100 = "OFF"; Y20 = "OFF", Y21 = "ON" and the operation will be stopped.

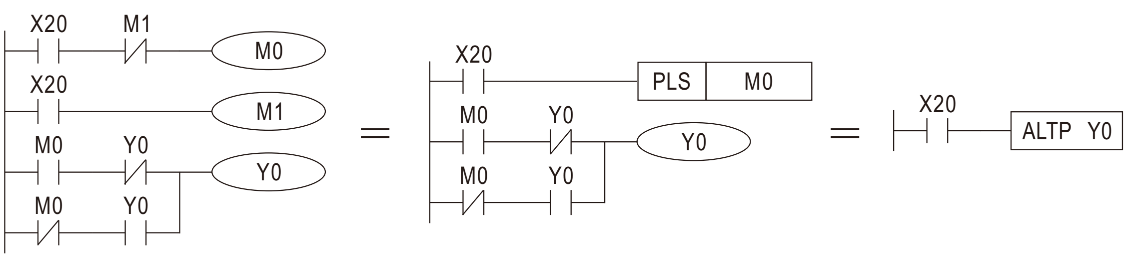
Produce the Flashing State



- 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.



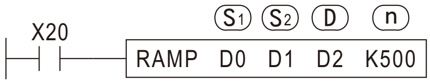
Traditional or Instruction to Make the Function of the 1st Trigger "ON" / 2nd Trigger "OFF"



FNC 67	RAMP (S1) (S2) (D) (n)	Ramp Variable Value	1	2	M	3
			○	○	○	○

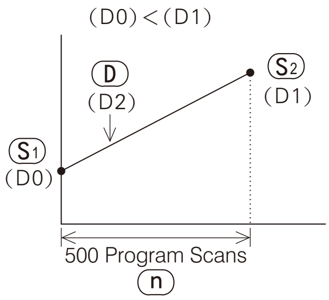
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●					
S2													●					
D													●					
n													○			○		

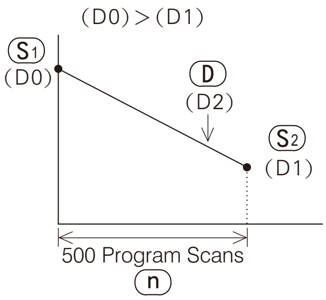
• D occupies 2 components • n = 1~32,767



S1 : the initial value of the ramp variation
S2 : the destination value of the ramp variation
D : the value of the course of the ramp variation
n : to assign the using number of times that ramp action is completed

- Write the initial point value of the ramp variation to D0 and write the destination value of the ramp variation to D1.
  When X20 = "ON" and (D0)<(D1), then the current value of D2 will increase from the smaller set value of D0 to the bigger set value of D1.
  When X20 = "ON" and (D0)>(D1), then the current value of D2 will decrease from the bigger set value of D0 to the smaller set value of D1.
  It takes 500 of PLC's Scan Times for the current value shifted from the set value of D0 to the set value of D1.





When the instruction is activated, the "course" value of the ramp variation will be reflected on (D2) while the number of the scanned times will be reflected on (D3).

- As shown in the diagram above, whether the continuous 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 the wanted characteristic of the RAMP instruction's result is a Linear Gradient, the interval of the RAMP instruction is performed must be equal at each time. In terms of this purpose, it's suggested 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 restart from the beginning.

- 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.

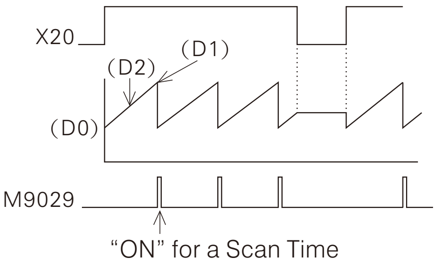
- The instruction can work with the analog output to incorporate the action of the buffered start/stop.

- If X20 = "ON" and PLC turns from STOP to RUN, please clear D3 as "0" at the beginning of the program.

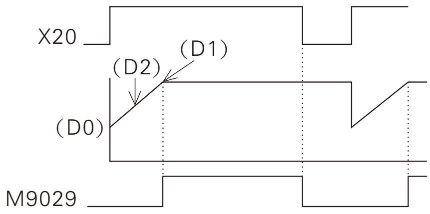
Operation Modes (Swappable by Flag M9026)

When the RAMP instruction is performed, the operation mode will depend on the status of Special Relay M9026.

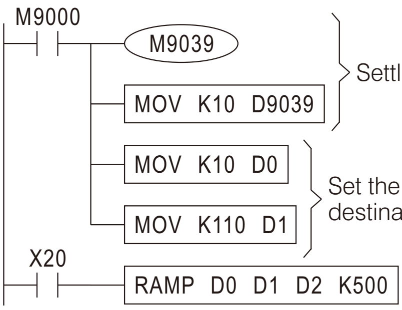
- If M9026 = “OFF”, it will generate contiguous ramp results. (Repeat Mode)



- If M9026 = “ON” , it will generate only one ramp results. (Hold Mode)



Program Model by Using the Fixed Scan Time Function



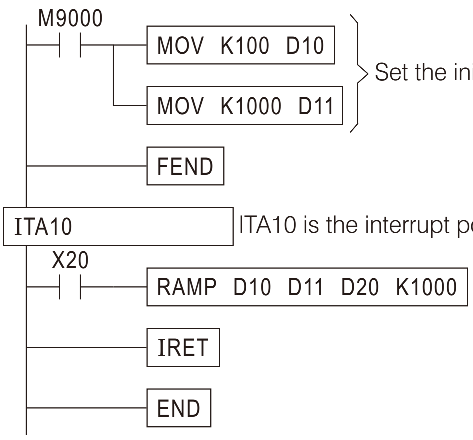
Settle the Scan Time at 10ms.

Set the initial value of the ramp variation at 10 and the destination at 110.

Time of the entire process of the RAMP instruction (the current value of D2 changes from the set value of D0 to the set value of D1) will be 10ms × 500 = 5,000ms = 5 Sec.

Since the PLC's Scan Time is settled at 10ms (each Scan Time is fixed) the ramp variation is like a Linear Gradient. However, in the program model referred above, it should be noted that the set value of the fixed Scan Time is larger than the maximum value of the actual Scan Time a little. Otherwise, the fixed Scan Time function would be useless. To oversee the D9012 register will get the maximum value of the actual Scan Time.

Program Model by Using the Time Interrupt Function



Set the initial value of the ramp variation at 100 and the destination at 1000.

ITA10 is the interrupt pointer of the time interrupt at 10ms.

Time of the entire process of the RAMP instruction (the current value of D20 changes from the set value of D10 to the set value of D11) will be 10ms × 1,000 = 10,000ms = 10 Sec.

FNC 69	SORT (S) (m1) (m2) (D) (n)	Sort Tabulated Data	1	2	M	3
			○	○	○	○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													○					
m1																○		
m2																○		
D													○					
n													○			○		

• Each of the S and D respectively occupie ($m_1 \times m_2$) components • m_1 = 1~32 • m_2 = 1~6 • n = 1~m_2

```
            (S)  (m1) (m2)  (D)    (n)
 X20
 -| |----[SORT  D0   K5   K4   D100   D200]
```

S : the head register ID number of the original data array

m_1 : the number of data record sets to be sorted

m_2 : the number of data at an arrangement for each set

D : the head register ID number of the data array where to store the sorted results

n : to assign the n^{th} data in an arrangement as the sort basis

Original Data Array
(Starting from the appointed register (S))

Data Arrangement			
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 an data array that has a serial of data record sets (the number of sets is designated by (m1)). Each set has a grouped arrangement (the number of grouped data at each arrangement set is designated by (m2)) while (n) is used to assign the n^{th} arrangement as the routing basis to sort the data array.
The (S) is to designate the head register ID number of the original data array to be sorted and the (D) is to designate the head register ID number of the data array where to store the sorted results.

• When X20 = "ON", the SORT instruction is starting to perform. This instruction needs take 4 (the content of (m1)) scan cycles to complete the sort action. When the sort process is completed, the instruction's Execution Completed Flag M9029 = "ON" for a Scan Time and the sort action will be stopped.

• Both of the arrays (S) and (D) will separately occupy (m1) × (m2) consecutive registers.

• The SORT instruction can be used once only in the program.

Sorted Data Result Array
(Starting from the appointed register (D))
when D200 = 2

Data Arrangement			
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)

Sorted Data Result Array
(Starting from the appointed register (D))
when D200 = 4

Data Arrangement			
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.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
70	DTKY S D1 D2	Ten Key Input	○	○	○	○
71	DHKY S D1 D2 D3	Hexadecimal Key Input	○	○	○	○
72	DSW S D1 D2 n	Digital Switch (Thumbwheel) Input	○	○	○	○
73	SEGDP S D	Seven Segment Decoder	○	○	○	○
74	SEGL S D n	Seven Segment with Latch	○	○	○	○
76	ASC S D	Convert Letters to ASCII Code	○	○	○	○
77	PR S D	Print ASCII Code	○	○	○	○
78	DFROMP m1 m2 D n	Read FROM a Special Module		○	○	○
79	DTOP m1 m2 S n	Write TO a Special Module		○	○	○

FNC 70	D TKY (S) (D1) (D2)	Ten Key Input	1	2	M	3
			○	○	○	○

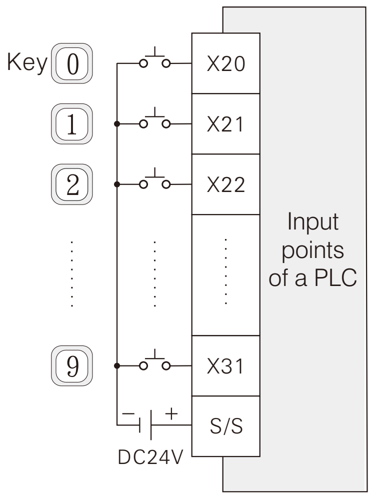
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●	●	●	●	○	○												
D1								●	●	●	●	●	●	○	●			
D2		●	●	●	○	○												
• S occupies 10 components					• D2 occupies 11 components													

S : the initial device of the key input

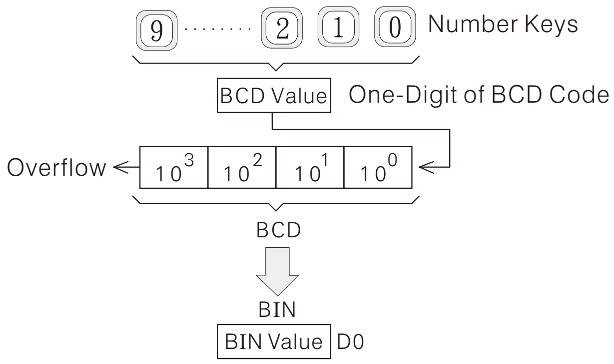
D1 : to place where the key input value is stored

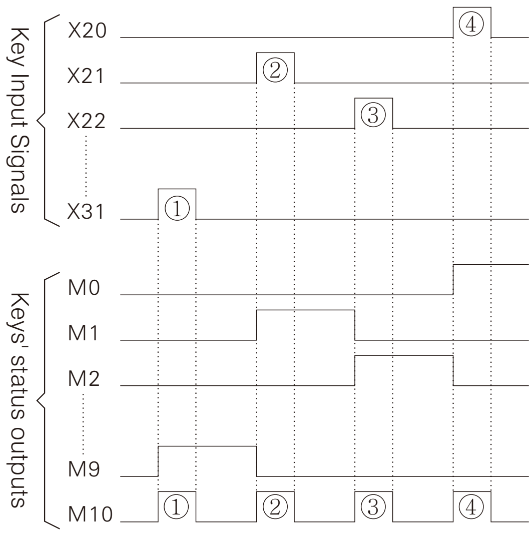
D2 : the initial destination device of the keys' status outputs

- The instruction designates consecutive ten input devices and those are starting from (S), therefore each one of the input is to represent a decimal number 0~9 in order. Since these 10 external input devices are connected to 10 keys and these are pressed in 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, then the input value will be placed in (D1). Also the instruction occupies 11 consecutive devices which starting from (D2) will be used 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 starts 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 the related X20~X31 are input in order ① ② ③ ④ then the result 9,120 is stored in D0.
- When X31 is triggered (key #9 is pressed), M9 will turn "ON" and remain "ON", until the next key is pressed (when the X21 = "ON" → M9 = "OFF"). The same situation applies to other keys.
- If any of the keys for the X20~X31 is pressed, M10 = "ON" at the moment and the corresponding device M0~M9 will be turned "ON".
- When the status of X0 "ON" → "OFF", the value of D0 will remain but the M0~M10 will all turn "OFF".

FNC 71	D HKY (S) (D1) (D2) (D3)	Hexadecimal Key Input	1 ○	2 ○	M ○	3 ○

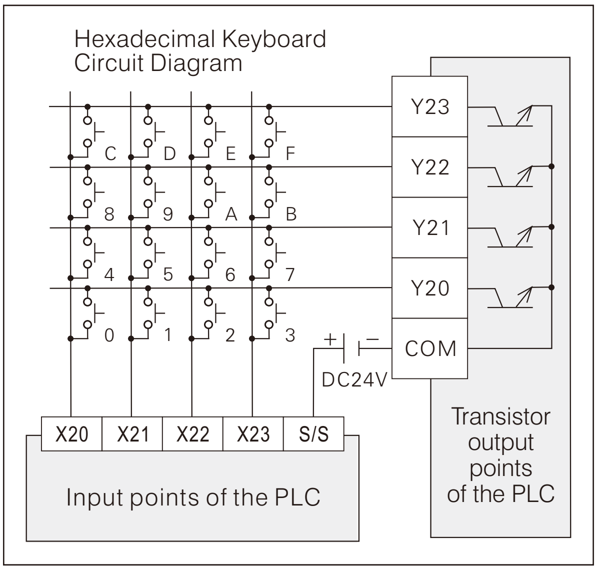
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●																	
D1		●																
D2											●	●	●	○	●			
D3		●	●	●	○	○												

• S occupies 4 components • D1 occupies 4 components • D3 occupies 8 components

X0 ─┤├── HKY X20 Y20 D0 M0 (S) (D1) (D2) (D3)

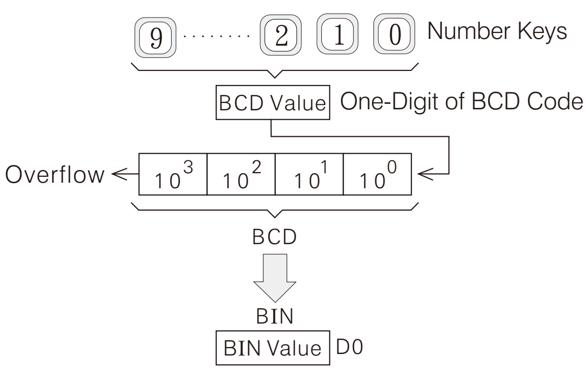
S : multiplex scanning initial point of the inputs from the keyboard
D1 : multiplex scanning initial point of the outputs to the keyboard
D2 : to place where the keyboard input value is stored
D3 : the initial destination device of the keys' status outputs

- The instruction creates the Hexadecimal Keyboard 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 keyboard 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 connected with X20~X23 and Y20~Y23. When X20 = "ON", the instruction is active.
  The value input by the keyboard is placed in D0 and the status of those keys are 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 first key activated 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 model.

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 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 will be overflowed.

Function Key Input

F	E	D	C	B	A	Function Keys
M5	M4	M3	M2	M1	M0	Keys' Status Outputs

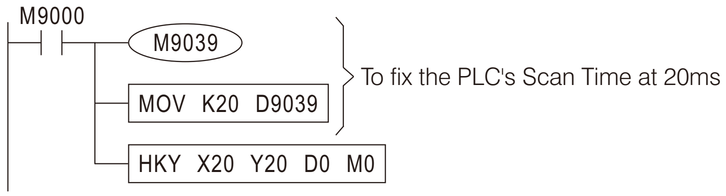
- The A~F keys are defined as function keys.
- If a function key is pressed, the corresponding key's status output will turn "ON" and remain the same status, until other function key has been pressed then the previous output will change "ON" → "OFF".
  For example, when A is pressed, M0 will turn and remain "ON". And if F is pressed after that, M5 will turn turn and remain "ON" while M0 = "OFF".

Status Outputs of the Keys

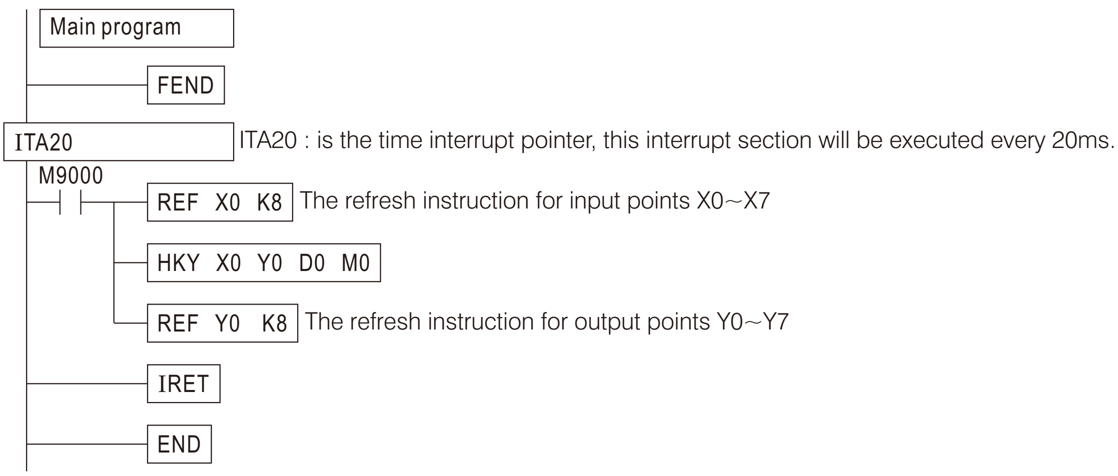
- If the keys A ~ F are pressed, the corresponding key's status output M0~M5 will turn "ON".
- During the period when any one of the function keys A ~ F is pressed, M6 = "ON"; And M6 = "OFF" when all the function keys are released.
- During the period when any one of the number keys 0 ~ 9 is pressed, M7 = "ON"; And M7 = "OFF" when all the number keys are released.
- When the conditional contact X0 = "OFF", the input value will remain as same as the statuses before it is disabled; However, M0~M7 will all turn "OFF".

Notice

- When the instruction is performed, it should take 8 PLC's Scan Times to effectively capture a key. When the PLC's Scan Time is too long or too short, it may affect to read the key input signal incorrectly. The solution is shown as follows:
- If the PLC's Scan Time is too short, this may possibly does not have enough time to pick the status from input points then it will cause to read the keyboard incorrect. Please use the constant Scan Time function to fix the Scan Time at 20ms.



- If the PLC's Scan Time is too long, this will cause the keyboard's reading response slow down. Please use the timer interrupt function to do the keyboard scan function at every 20 ms.

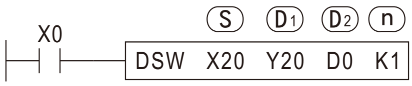


FNC 72	DSW (S) (D1) (D2) (n)	Digital Switch (Thumbwheel) Input	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●																	
D1		●																
D2											●	●	●	○	●			
n																○		

• n = 1~2

• When n = 1, S occupies 4 components and D1 occupies 4 components, D2 occupies 1 components
• When n = 2, S occupies 8 components and D1 occupies 4 components, D2 occupies 2 components



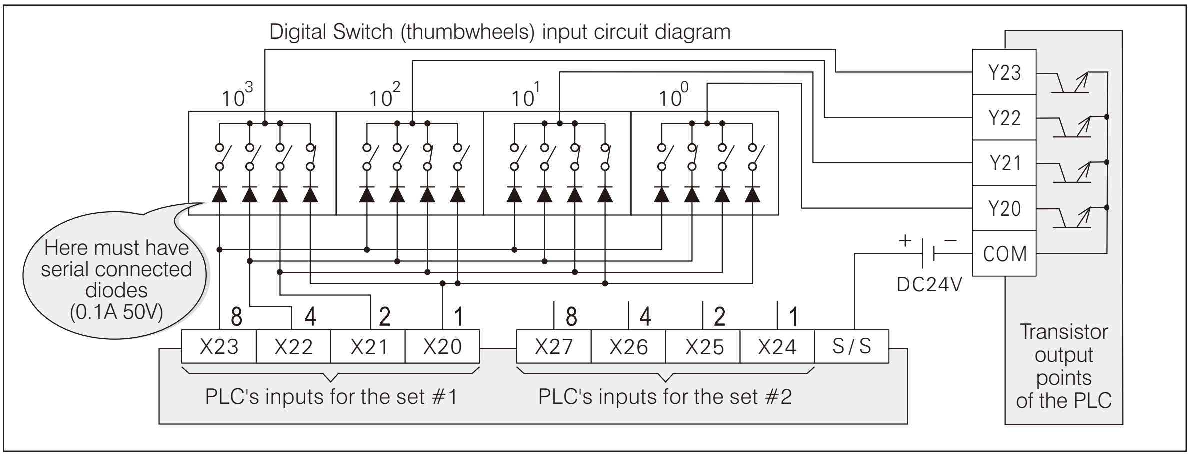
S : the initial input point of multiple scans from the digital switch

D1 : the initial output point of multiple scans to the digital switch

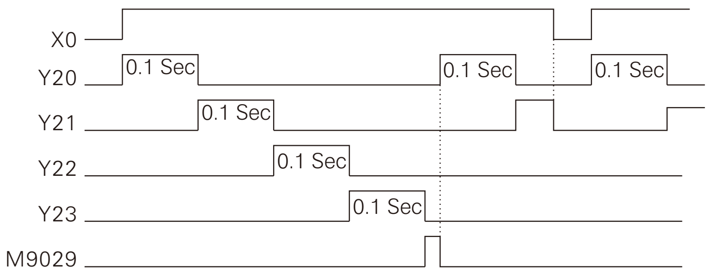
D2 : to place where the switches' input value is stored

n : the number of digital switch sets is connected

- The instruction reads one set (or two sets) of four-digit thumbwheel switches by 4 (or 8) consecutive input points, those consecutive inputs are starting from (S). Also, it uses 4 output points to scan which are starting from (D1). If the (n) = K1, it reads up one set of digital switch and puts a four-digit value to (D2); besides, If the (n) = K2, it reads up two sets of digital switches and puts two four-digit values to (D2) and (D2) +1.

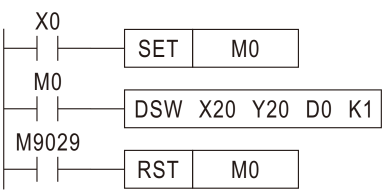


- The diagram shown above is a circuit of a multiplexed 4 thumbwheels switch which is composed by X20~X23 and Y20~Y23. When X0 = "ON", the instruction will start to be performed, the value of the thumbwheels will be read and converted into a BIN format then the value will be stored in D0. If (n) = K2 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 the DSW's scan sequence. When X0 = "ON", Y20~Y23 will automatically cycle the scan. If each cycle is completed, the Execution Completed Flag M9029 will "ON" for a Scan Time.
- Better use a transistor output unit for the multiplex scan outputs Y20~Y23.
- The instruction can only be used once in the program.

Method of Using the Relay Output Unit for the Switch Scan



- X0 uses the push-button switch.
- The DSW instruction will read the value of the thumbwheel switch once when X0 is triggered. At the rest time, the DSW instruction is inactive and not to scan. Therefore, could reduce the scan problem at the relay output.

FNC 73	SEGD P (S) (D)	Seven Segment Decoder	1	2	M	3
			○	○	○	○

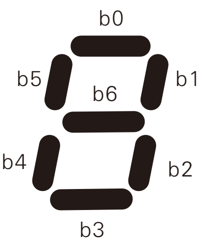
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

```
          (S)  (D)
X20
|-| |----[SEGD  D0  K2Y20]
```

S : the source device to be decoded

D : devices of decoded outputs

- When X20 = "ON", decode the content value (nibble format 0~F) of D0's lowest four bits (b0~b3) 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)		Composition of the seven segment display	(D)								Data Displayed
Hexadecimal Number	Bit Format		b7	b6	b5	b4	b3	b2	b1	b0	
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

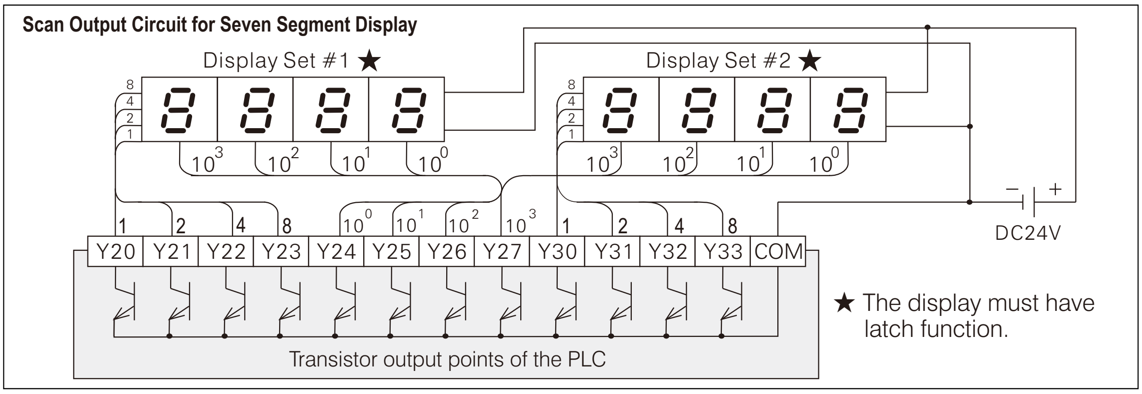
FNC 74	SEGL (S) (D) (n)	Seven Segment with Latch	1 ○	2 ○	M ○	3 ○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D		●																
n																○		

- n = 0~7
- When n = 0~3, S occupies 1 component and D occupies 8 components
- When n = 4~7, S occupies 2 components and D occupies 12 components

S : the source decimal value to be shown in the seven segment display
D : the initial point for the scan outputs of the seven segment display
n : the polarity designated for data signals and latch signals

- This instruction scan outputs to one (two) set(s) of four-digit seven segment display will occupy 8 (12) consecutive output points initiating from (D) and demonstrates the content value of (S) on the seven segment display. The content of (n) is not only to assign one or two sets of four-digit display for the scan outputs, but also to designate the polarity combination for the terminal of PLC's outputs and the displayer's inputs.



- 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 on the Display 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 the Display Set #2.
- When X20 = "ON", Y24~Y27 will cycle the output scan automatically. It takes 12 PLC's Scan Time for a display cycle and M9029 will turn "ON" for a PLC's Scan Time when each cycle is completed.

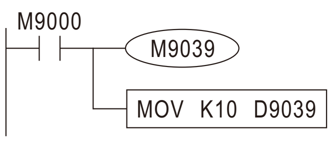
Set Value about (n)

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 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's output terminal and the displayer's data input terminal	Same		Different		Same		Different	
Polarity of the PLC output terminal and the displayer's data input terminal	Same	Different	Same	Different	Same	Different	Same	Different
n	0	1	2	3	4	5	6	7

The output will relate to the table above. If the displayer's character is unsure, can orderly insert a number 0~3 or 4~7 to (n) until the seven segment display can demonstrate correctly.

Notice



When the instruction is performed, each output scan will need 10ms at least. If the PLC's 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 (S) (D)	Convert Letters to ASCII Code	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	Key-in eight English letters from computer																	
D											●	●	●		●			

• D occupies (the inputted numbers of English letters in S ÷ 2) components

```
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 (Higher 8 bits)	b0 (Lower 8 bits)	
42H(B)	41H(A)	D0
44H(D)	43H(C)	D1
46H(F)	45H(E)	D2
48H(H)	47H(G)	D3

• If M9161 = “ON”, each English letter will take a whole register for to store a converted ASCII code, where the lower 8 bits (b7~b0) of the register is to store the ASCII code and higher 8 bits (b15~b8) will all fill with "0".

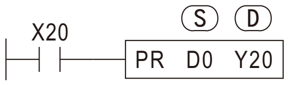
b15 (Higher 8 bits)	b0 (Lower 8 bits)	
00H	41H(A)	D0
00H	42H(B)	D1
00H	43H(C)	D2
00H	44H(D)	D3
00H	45H(E)	D4
00H	46H(F)	D5
00H	47H(G)	D6
00H	48H(H)	D7

• 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 S D	Print ASCII Code	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D		●																

• S occupies 4 components • D occupies 10 components



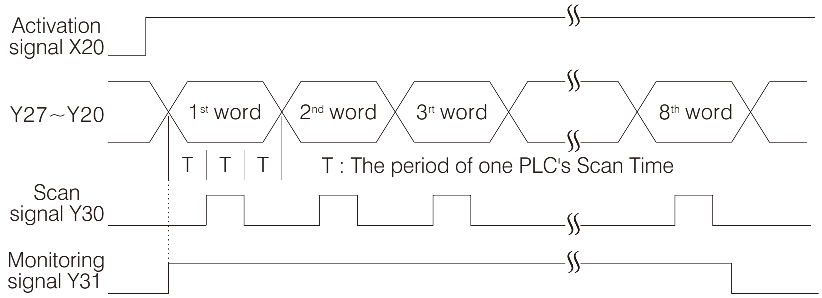
S : the source devices where ASCII codes are stored

D : the output points to export the ASCII codes

- The instruction will read ASCII codes of 4 (or 8) source registers (started from S and each coed will take a byte). And then, orderly output the ASCII codes to the designated consecutive 8 output points (started from D).
- The process referred above designates 8 points from Y27 (the first bit) to Y20 (the last bit), those 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:

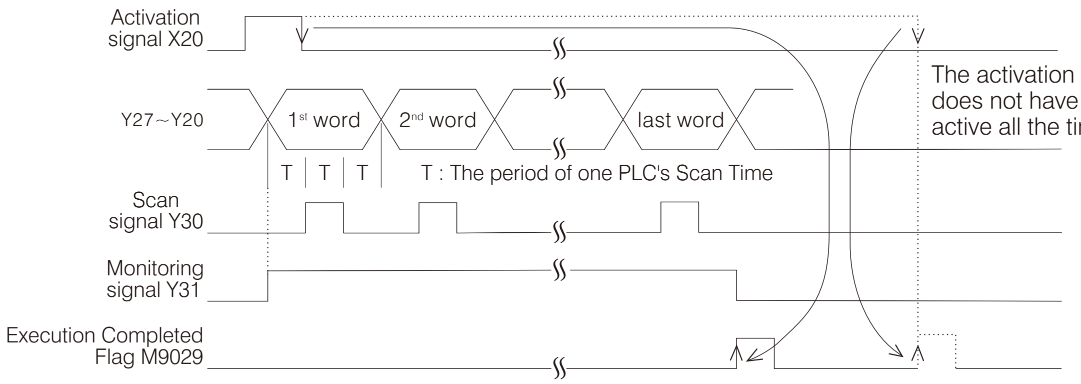


S	Up 8 bits	Down 8 bits
D0	2nd word	1st word
D1	4th word	3rd word
D2	6th word	5th word
D3	8th word	7th word

- 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 export is 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 PLC's 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.

FNC 78	DFROMP m1 m2 D n	Read FROM a Special Module	1	2	M	3
				○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
m1													○			○		
m2													○			○		
D								●	●	●	●	●	●	○				
n													○			○		

• For the VS2 or VSM series, m1 = 1~8; for the VS3 series, m1 = 1~16 • m2 = 0~32,767

• The 16-bit instruction, D occupies n components • The 32-bit instruction, D occupies (2×n) components • n = 1~32,767

X20 — FROM K2 K5 D0 K4 (m1 m2 D n)

m1 : the position number of the specified special module
m2 : the initial number of the BFMs to be read
D : the initial device of storage space for reading up BFMs
n : the number of BFMs to be read from the special module

- The Main Unit of the VS series PLC uses this instruction to read BFMs data of the Special Module.
- Since the m1 = K2, m2 = K5, n = K4 and when X20 = "ON", 4 word data at the BFM #5~BFM #8 in the specified No. 2 Special Module will be read and store into D0~D3.
- The m1 is to appoint a Special Module, the available number is from K1 to K16. For the Main Unit, K1 represents to access the closest Special Module, and so on.
- When X20 is "OFF", the instruction will not be performed but the data which had read previously will still remain.

The BFM Number m2

- The VS series Special Module has the components of BFMs (Buffer Memory) which are used to store the setting values and various operation statuses about the module. Each BFM is a 16-bit space. The different type of special module has a different number of BFM registers. The ID number of BFM registers is coded in a decimal method, such as #0, #1,.... #9, #10,...
- If a Main Unit is through the BFM to manage the module, this module is called the Special Module.

The Number of Data Groups n to be Transferred

- The 16-bit instruction

FROM K1 K0 D0 K4 (n)

D0	←	BFM #0
D1	←	BFM #1
D2	←	BFM #2
D3	←	BFM #3

- The 32-bit instruction

DFROM K1 K4 D100 K2 (n)

D100, D101	←	BFM #4, BFM #5
D102, D103	←	BFM #6, BFM #7

- The number of the data groups to be transferred is determined by the n . n = 4 in the 16-bit instruction has the same meaning with n = 2 in the 32-bit instruction.

- The related special device for this instruction:

Relay ID No.	Description
M9028	Preventing to operate the FROM/TO repeatedly. When M9028 is "OFF", disallows interrupt during FROM/TO is in operation. When M9028 is "ON", FROM/TO in an interrupt subroutine is ineffective.

FNC 79	DTOP m1 m2 S n	Write TO a Special Module	1	2	M	3
				○	○	○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
m1													○			○		
m2													○			○		
S							●	●	●	●	●	●	●	○		●		
n													○			○		

• For the VS2 or VSM series, m1 = 1~8; for the VS3 series, m1 = 1~16 • m2 = 0~32,767

• The 16-bit instruction, S occupies n components • The 32-bit instruction, S occupies (2×n) components • n = 1~32,767

```
       m1 m2 S  n
X20
─┤├───[TO K2 K0 D0 K1]
```

m1 : the position number of the specified special module

m2 : the initial number of the BFMs to be written

S : the initial device of data source storage space

n : the number of BFMs to be written to the special module

• The Main Unit of the VS series PLC uses this instruction to write data into BFMs at the Special Module.

• Since the m1 = K2, m2 = K0, n = K1 and when X20 = "ON", an 1 word data at the D0 of the Main Unit will be written into specified No. 2 Special Module's BFM #0.

• The m1 is to appoint a Special Module, the available number is from K1 to K16. For the Main Unit, K1 represents to access the closest Special Module, and so on.

• When X20 is "OFF", the instruction will not be performed but the BFMs' data which had written previously will still remain.

The BFM Number m2

• The VS Special Module has the components of BFMs (Buffer Memory) which are used to store the setting values and various operation statuses about the module. Each BFM is a 16-bit space. The different type of special module has a different number of BFM registers. The ID number of BFM registers is coded in a decimal method, such as #0, #1,.... #9, #10,...

• If a Main Unit is through the BFM to manage the module, this module is called the Special Module.

The Number of Data Groups n to be Transferred

• The 16-bit instruction

```
                    n
─┤├───[TO K1 K0 D0 K4]

D0 → BFM #0
D1 → BFM #1
D2 → BFM #2
D3 → BFM #3
```

• The 32-bit instruction

```
                       n
─┤├───[DTO K1 K4 D100 K2]

D100, D101 → BFM #4, BFM #5
D102, D103 → BFM #6, BFM #7
```

• The number of the data groups to be transferred is determined by the n . n = 4 in the 16-bit instruction has the same meaning with n = 2 in the 32-bit instruction.

• The related special device for this instruction:

Relay ID No.	Description
M9028	Preventing to operate the FROM/TO repeatedly. When M9028 is "OFF", disallows interrupt during FROM/TO is in operation. When M9028 is "ON", FROM/TO in an interrupt subroutine is ineffective.

MEMO

6-10 Serial Communication Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
80	RS S m D n n1	Receive/Send Communication Instruction	○	○	○	○
81	DPRUNP S D	Parallel Run (Octal Mode)	○	○	○	○
82	ASCIP S D n	Convert HEX to ASCII	○	○	○	○
83	HEXP S D n	Convert ASCII to HEX	○	○	○	○
84	CCDP S D n	Check Code	○	○	○	○
87	CPUL S1 S2 n	CPU Link Communication Instruction	○	○	○	○
88	PID S1 S2 S3 D	PID Control Loop	○	○	○	○
89	LINK S1 S2 n	Easy Link Communication Instruction	○	○	○	○

FNC 80	RS (S) (m) (D) (n) (n1)	Receive/Send Communication Instruction	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●					
m													○			○		
D													●					
n													○			○		
n1																○		

• m, n = 0~4096 • For the VS1, VS2 or VSM series, n_1 = 1~3; for the VS3 series, n_1 = 1~5

```
        (S)   (m)    (D)    (n)   (n1)
X20
-| |----RS  D0  D200  D100  D201  K1
```

S : the head register ID number for the data to be sent

m : the length of the data to be sent

D : the head register ID number for the receiving data

n : the length of the data want to receive

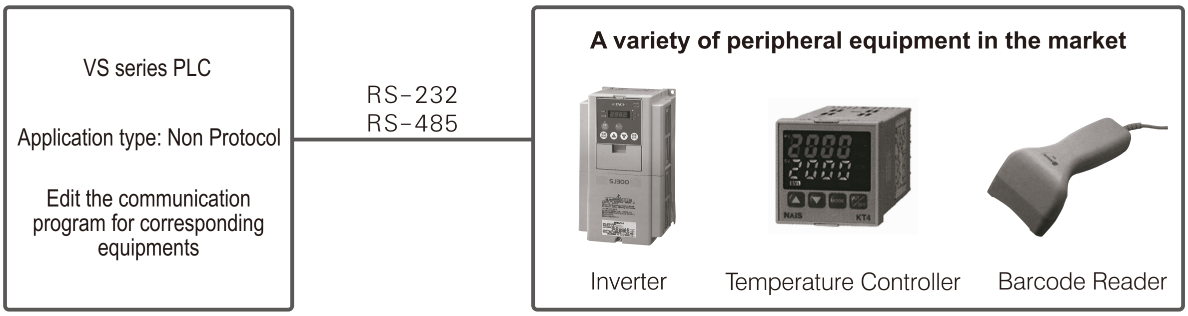
n_1 : to appoint the communication port, 1~5 = CP1~CP5

- The VS series PLC uses this instruction to send out or receive the data via its Communication Port CP1~CP5, and that is by the serial communication interface to communicate with external peripheral equipment.

- As many peripheral equipments in the market (e.g. inverters, barcode readers, card readers, electronic displays, etc.) are equipped with serial communications interfaces and have their own protocols. In order to transfer data between the PLC and those peripherals, a PLC user need to write the program with communicable function therefor this RS instruction is to adapt with these communication protocols.

- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the “Application type:” become the “Non Protocol”. Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the “Project” -- “COM Port Setting”.

- As shown in the figure below, use the Ladder Master S to set the communication port's “Application type:” of the VS series PLC as the “Non Protocol” and at the same page to set other relevant parameters. Set the station No. and related parameters for each peripheral. Next, in the VS series PLC, according to the communication protocol of the peripheral equipments, compile the program with relevant communication.



- Designate (m) to be K0 when there is not need to send out data and designate (n) to be K0 when there is not need to receive data.

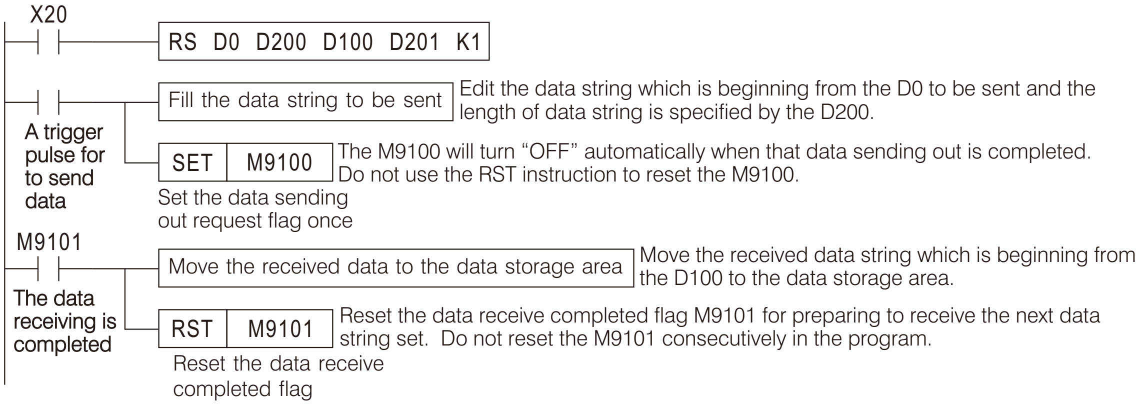
- The data transmissions can be divided into the 16-bit mode (M9161 = “OFF”) or 8-bit mode (M9161 = “ON”) when the RS instruction is performed. Also, the mode flag M9161 should be set before the RS instruction is started.

• The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9100	CP1 RS instruction data sending out request flag.
M9101	CP1 RS instruction data receive completed flag.
M9102	CP1 RS instruction data receive time-out flag.
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
M9110	CP2 RS instruction data sending out request flag.
M9111	CP2 RS instruction data receive completed flag.
M9112	CP2 RS instruction data receive time-out flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
M9120	CP3 RS instruction data sending out request flag.
M9121	CP3 RS instruction data receive completed flag.
M9122	CP3 RS instruction data receive time-out flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
M9130	CP4 RS instruction data sending out request flag.
M9131	CP4 RS instruction data receive completed flag.
M9132	CP4 RS instruction data receive time-out flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
M9140	CP5 RS instruction data sending out request flag.
M9141	CP5 RS instruction data receive completed flag.
M9142	CP5 RS instruction data receive time-out flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.

Register ID No.	Description
■D9101	The CP1's amount of residual data to be sent out by the instruction RS.
■D9102	The CP1's amount of the data already received by the instruction RS.
■D9111	The CP2's amount of residual data to be sent out by the instruction RS.
■D9112	The CP2's amount of the data already received by the instruction RS.
■D9121	The CP3's amount of residual data to be sent out by the instruction RS.
■D9122	The CP3's amount of the data already received by the instruction RS.
■D9131	The CP4's amount of residual data to be sent out by the instruction RS.
■D9132	The CP4's amount of the data already received by the instruction RS.
■D9141	The CP5's amount of residual data to be sent out by the instruction RS.
■D9142	The CP5's amount of the data already received by the instruction RS.

Sequence of Data Sending and Receiving (Using the CP1 as an example)



The Related Flags and Data Registers

① The Data Sending Out Flag M9100:

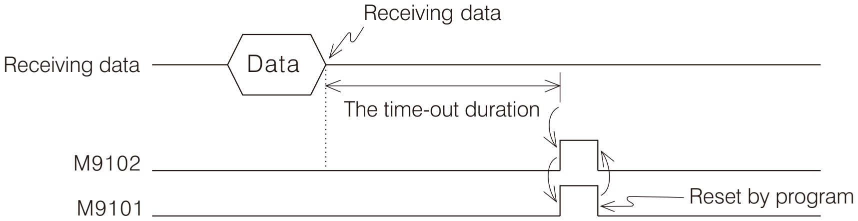
- When the conditional contact X20 = "ON", the RS instruction is performed. At this time, if a pulse signal forces the status of M9100 to be "ON", the content values of the registers which are beginning from D0 will be sent out via the serial communication port CP1. After the data sending is completed, the M9100 will be reset to "OFF" automatically.

② The Receive Completed Flag M9101:

- When the conditional contact X20 = "ON", the RS instruction is performed. The CP1 of PLC is ready to receive communication data.
- When the data receiving is completed, the M9101 = "ON". At this moment, the received data in the buffer should be moved to the data storage area, and then M9101 could be reset to "OFF". Afterwards, PLC will be ready for the status of receiving immediately.

③ The Receive Time-out Flag M9102:

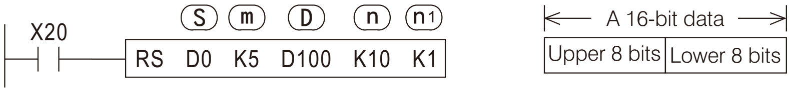
- When the data of receiving is paused and the waiting time exceeds the Time-out duration (designated by the "COM Port Setting"), the M9102 will turn "ON" to represent the happening of Time-out also the Receive Completed flag M9101 will be forced "ON" to terminate the data receiving action.
  The M9102 will not reset automatically, must use an instruction in the program to reset the status of M9101 then the M9102 will be reset too.
- By using the function of the Time-out flag, the PLC could receive the data of transferring from peripheral equipments which without the particular length or "End of Text".
- The Time-out duration is specified by using the programming software Ladder Master S – "Project" – "COM Port Setting".



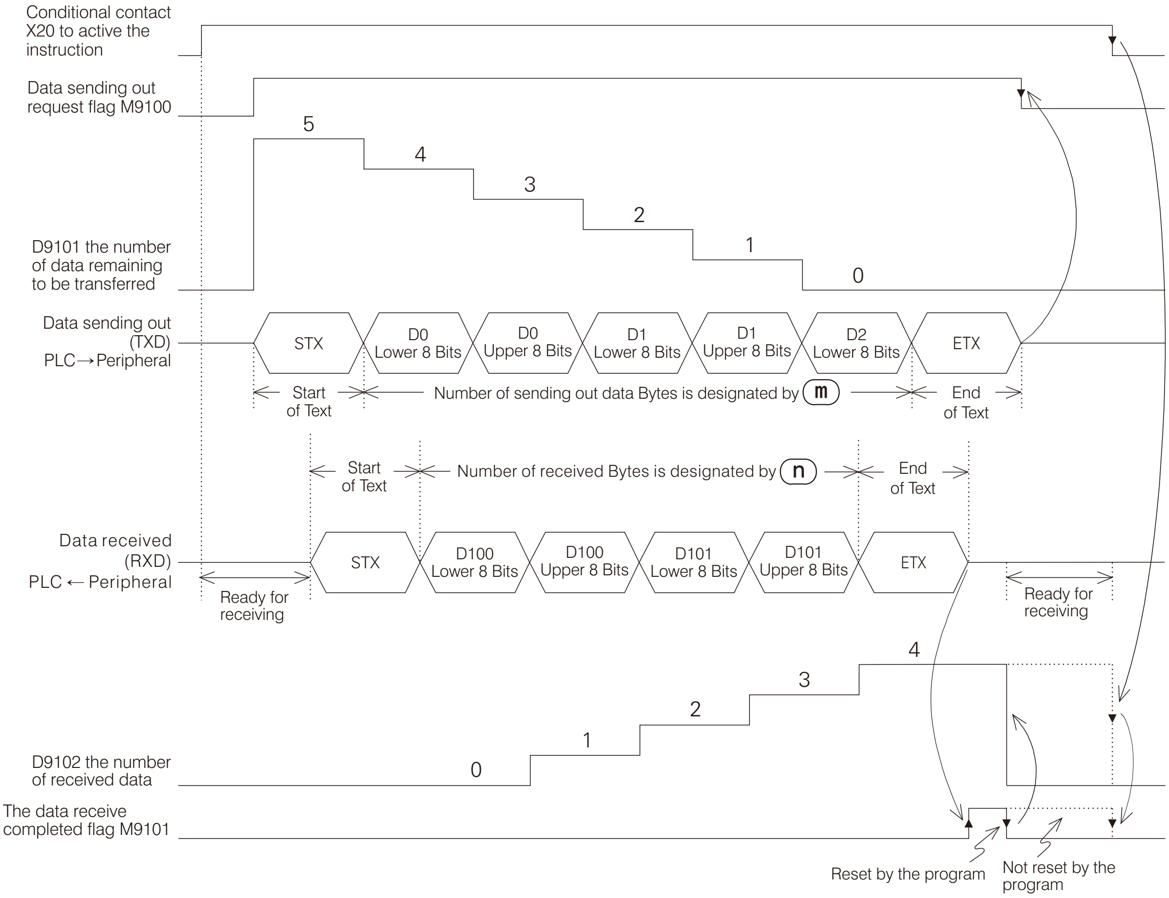
④ The Communication Abnormal Flag M9103:

- When the RS instruction occurs Parity error or Framing error, the Communication Abnormal Flag M9103 will turn "ON". The M9103 must use an instruction in the program to reset the status of M9103.

Description of Data Sending / Receiving (by the CP1): 16-bit Mode (M9161 = "OFF")



Divide the 16-bit data into two Bytes of data: Upper 8 bits data and Lower 8 bits data.



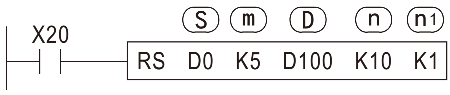
Either one of the condition below is completed will cause the receive completed flag M9101 “ON” and stop to receive.

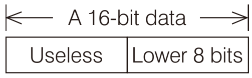
① The number of assigned n bytes is satisfied.

② The End of Text is received.

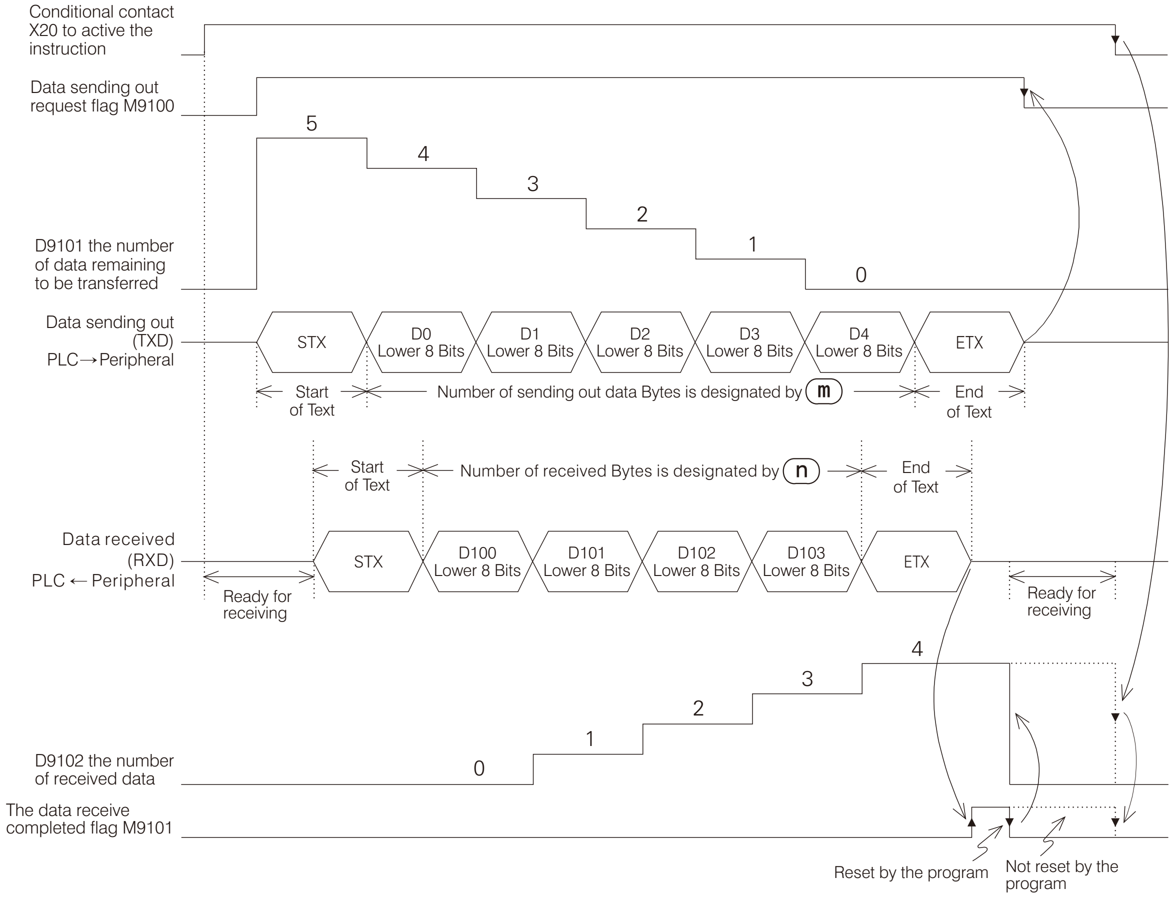
③ The occurrence of Time-out.

Description of Data Sending / Receiving (by the CP1): 8-bit Mode (M9161 = "ON")





The Lower 8 bits of the 16-bit data is treated as one Byte data. (the Upper 8 bits data are useless)



Either one of the condition below is completed will cause the receive completed flag M9101 “ON” and stop to receive.

① The number of assigned n bytes is satisfied.

② The End of Text is received.

③ The occurrence of Time-out.

FNC 81	DPRUNP (S) (D)	Parallel Run (Octal Mode)	1 ○	2 ○	M ○	3 ○

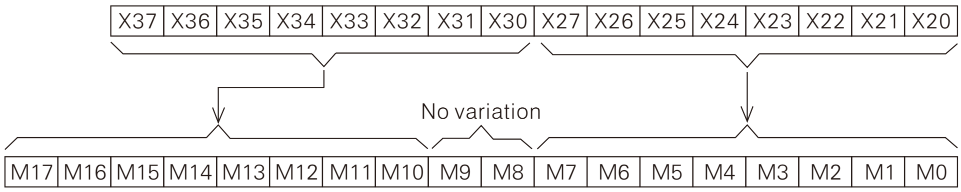
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●		●									
D								●	●									

- The ID number of the X, Y or M in the K*n*X, K*n*Y or K*n*M must assign to a number which the last digit is a zero “0”.
- When S = K*n*X, D must be K*n*M;   when S = K*n*M, D must be K*n*Y.

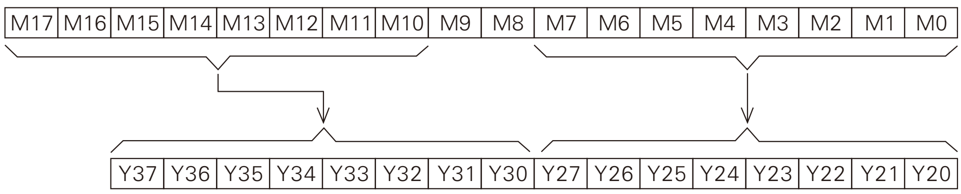
S : the transmission source devices

D : the transmission destination devices

- Transfer the source devices those are designated by (S) to the destination devices (D), the operation is by the octal number system format.
- When X0 = “ON”, transfer the content of K4X20 to K4M0 in the octal number system format.



- When X1 turns from “OFF” to “ON”, the content of K4M0 is transferred to K4Y20 in the octal number system format.



FNC 82	ASCI P (S) (D) (n)	Convert HEX to ASCII	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●		●			
n													○			○		

- When S or D is designated to the K*n*X, K*n*Y, K*n*M or K*n*S, that K*n* has to be K4
- n = 1~256

```
          (S)  (D)   (n)
 X20
|-| |----ASCI  D0  D100  K8
```

S : the head ID of data source
D : the head ID of the device where conversion results are stored
n : the number of hexadecimal data characters is selected

- When the instruction is performed, it converts a series of HEX characters at the source devices (starting from (S)) into a series of ASCII codes and transfers the result to the designated devices (D). The number of the converted characters is determined by (n).

- To convert each 0~F HEX value to a corresponding ASCII code is according to the table below:

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 8 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 starting from (D) into Upper 8 bits and Lower 8 bits, where two ASCII codes are stored respectively in a register.

(D)	n=8	n=7	n=6	n=5	n=4	n=3	n=2	n=1 (n)
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 starting from (D) into Upper 8 bits and Lower 8 bits, while the used registers' Upper 8 bits are filled with zero ("0") and Lower 8 bits are to store 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 (n)
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 (S) (D) (n)	Convert ASCII to HEX	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●	●		
D								●	●	●	●	●	●	○	●			
n													○			○		

• When S or D is designated to the KnX, KnY, KnM or KnS, that Kn has to be K4 • n = 1~256

```
      X21              (S)   (D)  (n)
 ──┤ ├────────┤ HEX  D100  D0   K8 │
```

S : the head ID number of data source

D : the head ID number of the position where conversion results are stored

n : the number of ASCII codes converted

- When the instruction is performed, it converts a series of HEX characters at the source devices (starting from (S)) into a serial of HEX values and transfers the result to the designated devices (D) . The number of ASCII codes converted is determined by (n) .
- into a serial of HEX values and transfers the result to the designated devices

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", the instruction converts 8 ASCII codes which are starting from D100 into a series of HEX values, and transfers the result to (D0) and (D1).
- If one of the content designated by the data source (S) is not an ASCII code in the table above (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) at 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

(D) A nibble (n)

D1 b15			D1 b0	D0 b15			D0 b0	
				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 at each Lower 8 bits of designated devices starting from (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

(D) (n)

D1 b15			D1 b0	D0 b15			D0 b0	
				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	CCDP (S) (D) (n)	Check Code	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●					
n													○			○		

• When S or D is designated to the KnX, KnY, KnM or KnS, that Kn has to be K4 • n = 1~256 • D occupies 2 components

```
       X20        (S)  (D)   (n)
 |—| |————[CCD   D0   D100  K8]
```

S : the head ID of continuous data source
D : the device where the result of SumCheck is stored
n : the number of source data to be checked

- Sum up the content of (n) bytes (by the unit of 8-bit) data headed with (S), the 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 the communication, the data's "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, the 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 make use of both the Upper 8 bits and Lower 8 bits at every related device (started from (S)) become a series of 8-bit data source to 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 only make use of the Lower 8 bits (ignore the Upper 8 bits) at every related device (started from (S)) become a series of 8-bit data source to 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 87	CPUL (S1) (S2) (n)	CPU Link Communication Instruction	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
n																○		
• S2 occupies 4 components • For the VS1, VS2 or VSM series, n = 1~3; for the VS3 series, n = 1~5																		

S1 : the communication table that describes the data receiving and transmitting

S2 : the working area for the instruction that occupies 4 registers

n : to assign the communication port, 1~5 = CP1~CP5

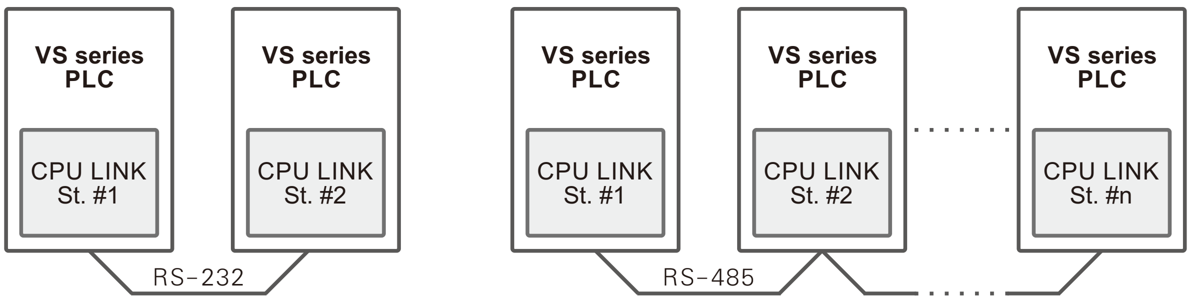
- The VS series PLC uses this instruction to share the particular data via its Communication Port CP1~CP5 with other VS PLCs, that achieved the purpose of distributed control.

- The main character of the CPU LINK is to share instant data.
  What we call "Instant" means all the communication work is dealt with instantly without being affected by the PLC scan time. Other communication method, such as Easy LINK, MODBUS communication and Non Protocol communication, will follow the PLC's executive cycle and thus is affected by the PLC scan time.
  What we call "share" means the way of the CPU LINK communication work is that each station in the communication network automatically transfers a part of data orderly to other stations in the network. This way the operation of the whole communication network can be understood through each station in the communication network.
  Through the function of CPU LINK that instantly sharing data with other PLCs, a distributed control system can be established.

- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the "Application type:" become the "CPU Link". Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the "Project" -- "COM Port Setting".

- This instruction could be used to transfer for various devices that is including all the Y, M, S, T, C, D and R (except for special M and special D).

- As shown in the figure below, first use the Ladder Master S to orderly open every PLC's project and set its "Application type:"as the "CPU Link" then to set an unique station No. (ranging from 0~254) hereafter connect to the programming port of every PLC to load the project to each PLC.
  Afterward, write this instruction in one of the PLC's program then edit the CPUL communication table to designate data sharing actions, hereafter to load this project to the specific PLC. Following these procedures, the purpose of sharing instant data between the PLCs is reached.



- When X20 = "ON", the CPUL instruction starts execution. According to the contents described in the "DISTRIBUTED_CTRL" communication table, it will arrange sharing particular data between stations, while the D100~D103 will be occupied to store the state of instruction execution.

- When the contents of the communication table specified by the S1 are execution completed from beginning to end, it will start over again from the first item set of the table.

- When the X20 turns from "ON" to "OFF", this instruction stops and the data sharing immediately stop but data which has transferred previously will still remain.

- The communication table is assigned by the S1 of the instruction.
  An example of the "CPU Link" provided by programming tool the Ladder Master S is shown below.

Item No.	Station No.	Device Range	Word/Bit	Disable Contact
1	0	D0 – D9	—	M1
2	1	D10 – D19	—	M1
3	2	D20 – D29	—	M2
4	3	D30 – D39	—	—

※ When the device is selected to be the bit device (Y, M, S or the contact of T, C), the initially ID number at the device range must be a multiplier of 8, such as Y0, Y10, Y20, M0, M24 or T8. Also, the length must be a multiplier of 8 as well, such as M0~M23 (24 devices), T8~T15 (8 devices), Y10~Y27 (16 devices), M24~M63 (40 devices).

The first item set in the table means that Station #0 transmits its D0~D9 to the D0~D9 of other stations in the communication network.

The second item set in the table means that Station #1 transmits its D10~D19 to the D10~D19 of other stations in the communication network.

The third item set in the table means that Station #2 transmits its D20~D29 to the D20~D29 of other stations in the communication network.

The fourth item set in the table means that Station #3 transmits its D30~D39 to the D30~D39 of other stations in the communication network.

The last column of the table is for the Disable Contact. If that specified contact is "ON", the communication item set will be ignored. For example, if M1 = "ON", the first and second item sets in the table will skip. This is the new function of the VS series, which can help designers effectively manage the operation of the communication table. A communication item set does not need to specify a Disabled Contact (such as the fourth item set), so that the command of the set does not have the disable control function.

- The working area of the executive instruction is starting from the S2 (using D100~D103 as the example).

S2		Description
D100	Lower 8 bits	The record of the station number when the first communication error is occurred
	Upper 8 bits	Instruction working status 0: Normal data transmitting / receiving A: The communication setting is normal but no response from the Slave station (Time-out occurs) B: Abnormal communication
D101 ~ D103	The working area is required for the system when this instruction is performed	

If any communication error occurs during the execution of the instruction, a code will be recorded in D100. Only when the content value of D100 is 0, the recording action can be executed.
Therefor, when there are possibilities of several errors, users can use the program to move out the content value of D100 then reset it to 0. This way allows the D100 to record the next error.

- To edit a communication table S1
  Use the Ladder Master S to set up a CPUL communication table and through its interactive window can set up and edit a communication table easily.
  In the structure of VS series PLC, the communication tables are a part of the project. When the programmer to copy or access the project, those tables will be duplicated automatically with the program.

FNC 88	PID (S1) (S2) (S3) (D)	PID Control Loop	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													○		○			
S2													○		○			
S3													○					
D													○		○			

• S3 occupies 25 components

```
         (S1) (S2)  (S3)    (D)
 X0
─┤ ├────┤PID  D0   D1   D100  D200├
```

S_1 : the set point value (SV)

S_2 : the measured present value (PV)

S_3 : the initial register ID of the parameters

D : the control output value (MV)

- This instruction takes a measured present 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 will start to perform; When X0 = “OFF”, this process will stop but the content value of D200 will remain as same as the value before it is disabled.
- 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 provide the “Auto-tuning” function, it can help users to decide three of parameters in the PID instruction. (Please refer to follow pages.)
- Since the Sampling time of the PID instruction is by way of to accumulate the number of scanned times of this instruction. Therefore, to plan the PID instruction in the program must pay attention on two following points:

  ① This PID instruction allows to be used in a subroutine, interrupted subroutine, step ladder diagram or related to the conditional jump instruction. However, at the time that the PID instruction is active, must make sure this instruction can be processed once and only once at every Scan Time. If this instruction has been processed more than once or had not been executed, that will cause the sampling error or imprecise operation.

  ② When the sampling time is shorter than a Scan Time, it would make a PID process error. Then the PLC will automatically treat 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 (S1)
PV_n : The measured present value (S2)
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
KP : The proportion constant
α : The constant of input filter
TS : The sampling time
TI : The integral time constant
TD : The time derivative constant
KD : The derivative filter constant

- The description of the parameter block (S_3 ~ S_3+24)

Parameter	Parameter Name/Function	Bit	Description	Available Range
S_3	Sampling time (Ts)		The time interval should longer than the PLC's Scan Time and the update period of the measuring system	1 ~ 32767ms
S_3+1	Direction of the PID operation and alarm control	b0	0 : Forward operation	—
			1 : Reverse operation	
		b1	0 : Disable the Process Value (PV_{nf}) input deviation alarm	
			1 : Enable the Process Value (PV_{nf}) input deviation alarm	
		b2	0 : Disable the Output Value (MV) deviation alarm	
			1 : Enable the Output Value (MV) deviation alarm	
		b3	Reserved	
		b4	0 : The Auto-Tuning function is not active	
			1 : To start the Auto-Tuning function, the flag will return to 0 after the Auto-Tuning is finished	
		b5	0 : Disable the range limit of the Output Value	
			1 : Enable the range limit of the Output Value	
		b6 ~ b15	Reserved	
S_3+2	Input filter (α)		Modify the parameter of input filter to smooth the huge changes from the measured present 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)		To adjust the proportion between the derivative output and the change of the Process Value (PVnf)	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	Set point of positive deviation alarm for Process Value (PVnf)		To set the trigger point of positive deviation alarm (upper limit); Active when S_3+1's b1 = "ON"(1)	0 ~ 32767
S_3+21	Set point of negative deviation alarm for Process Value (PVnf)		To set the trigger point of negative deviation alarm (lower limit); Active when S_3+1's b1 = "ON"(1)	
S_3+22	Set point of positive deviation alarm for the Output Value (MV)		To set the trigger point of positive deviation alarm (upper limit); Active when S_3+1's b2 = "ON"(1)	0 ~ 32767
	The maximum limit of Output Value (MV)		The maximum limit of Output Value (MV) is effective when S_3+1's b5 = "ON"(1)	−32768 ~ 32767
S_3+23	Set point of negative deviation alarm for the Output Value (MV)		To set the trigger point of negative deviation alarm (lower limit); Active when S_3+1's b2 = "ON"(1)	0 ~ 32767
	The minimum limit of Output Value (MV)		The minimum limit of Output Value (MV) is effective when S_3+1's b5 = "ON"(1)	−32768 ~ 32767
S_3+24	Alarm flags (for read only)	b0	The positive deviation alarm for Process Value (PVnf) is arrived	—
		b1	The negative deviation alarm for Process Value (PVnf) is arrived	
		b2	The positive deviation alarm for Output Value (MV) is arrived	
		b3	The negative deviation alarm for Output Value (MV) is arrived	

- S_3 +1's b2 and b5 should not be active at the same time.

- When any one of the S_3+1's b1, b2 or b5 is "ON", the parameters' block for the PID at S_3 will occupy S_3 ~ S_3+24 total 25 consecutive registers.

- When all of the S_3+1's b1, b2 and b5 are "OFF", the parameters' block for the PID at S_3 will occupy S_3 ~ S_3+19 total 20 consecutive registers.

The Description of the Forward or 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 measured input 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.
  For example, An 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 measured input Process Value (PVnf) < the Set Point Value (SV), it will generate a negative deviation then the change to increase the effect is called reverse operation.
  For example, 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.

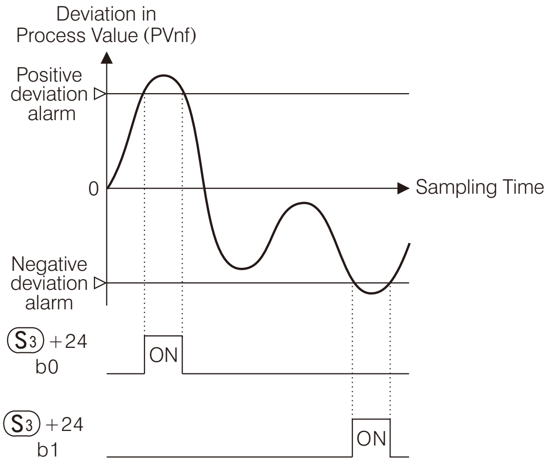
The Description of the Process Value (PVnf) and Output Value (MV) Deviation Alarm Functions

- If the of (S3)+1's b1 = "ON", PID instruction provides the Process Value (PVnf) deviation alarm. The parameters setting of the Process Value's deviation 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.

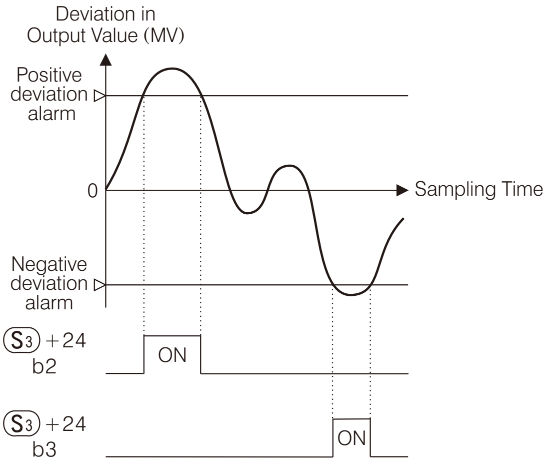
- If the (S3)+1's b2 = "ON", PID instruction provides the Output Value (MV) changing alarm. The parameters setting of the Output Value's changing alarm are stored in (S3)+22 and (S3)+23 then the results will put in (S3)+24's b2 and b3. The content of (S3)+23 is used as a negative value.

- The definition of the deviation in Manipulation Values: Deviation = (Present value) – (Previous Present value)

- The Process Value (PVnf) deviation alarm function:



- The Output Value (MV) deviation alarm function:



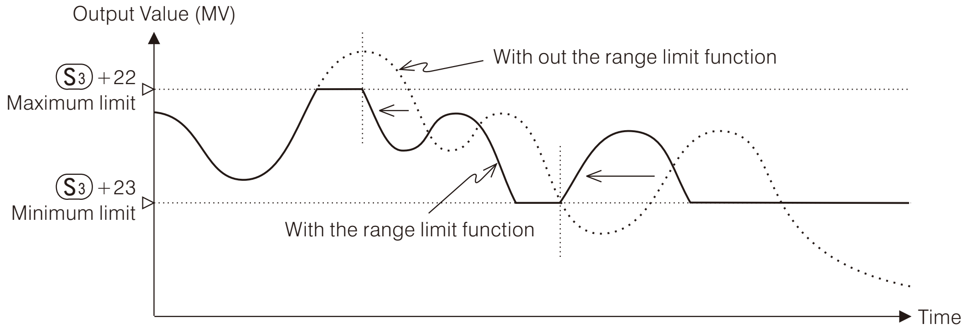
The Description of the Output Value (MV) Limit Function

- If the parameter of (S3)+1's b5 = "ON", the PID instruction provides the range limit function at the Output Value (MV). The parameters about the settings of the Output Value limits are stored in the (S3)+22 and (S3)+23.

- Since both of the limit and alarm functions are occupying to the same parameter registers (S3)+22 and (S3)+23, therefore only one of the function can be selected and the control bits in the (S3)+1's b2 and b5 should not be "ON" at the same time.

- This function is very useful for to limit the accumulative raise of the PID derivative value.

- The diagram of the Output Value (MV) Limit Function:



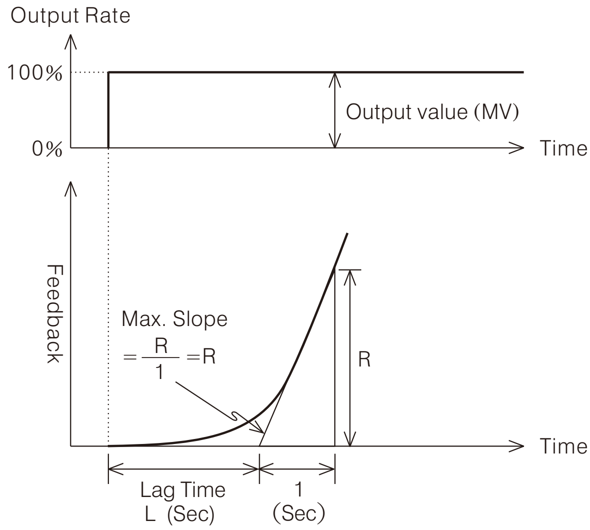
The Error Information of the PID Instruction

- If a setting value of parameter is not correct or the operation of a PID instruction occurs error, the Special Relay M9067 will be turned “ON”. And the Special Register D9067 will store the error code.

Error Code	Error Cause	Effect to the Instruction
6730	The setting value of Sampling Time (Ts) is beyond the range ($Ts < 1$)	The PID instruction will stop to operate
6732	The setting value of Input Filter (α) is beyond the range ($\alpha < 0$ or $\alpha >= 100$)	
6733	The setting value of Proportion Gain 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 Derivative Time Constant (T_D) is beyond the range ($T_D < 0$)	
6740	The Sampling Time $<=$ The Scan Time of PLC	The PID instruction will continue to operate
6742	The change of the measured Present Value is too large ($\triangle PV < -32768$ or $\triangle PV > 32767$)	
6743	The change 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 Gain (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 Gain (K_P), Integral Time Constant (T_I) and Derivative Time 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 to get the parameters is through to control the output rate became either 0 or 100% a few times. And then, observes the variation between processes and feedbacks, by those dynamic characteristics to get those parameters of PID.



Use the curve to get the PID's parameters

Control Method	Proportion Gain K_P(%)	Integral Time Constant T_I (× 100ms)	Derivative Time Constant T_D (× 10 ms)
P	$\frac{1}{R\times L}$ × Output value (MV)	—	—
PI	$\frac{0.9}{R\times L}$ × Output value (MV)	33×L	—
PID	$\frac{1.2}{R\times L}$ × Output value (MV)	20×L	50×L

Auto-Tuning Function

- The VS series provided the Auto-Tuning function which can uses some PID correlative parameters from user (such as: the operational direction at S3+1, Sampling Time Ts, Input Filter (α), Derivative Filter Gain 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 those 5 basic parameters: the operational direction at the b0 of S3+1, Sampling Time Ts S3, Input Filter (α) S3+2, Derivative Gain K_D S3+5 and Set Point Value S1.

② Fill in the parameters S3+14 and S3+15.

Parameters	Parameter Name/Function	Description
S3 +14	The Max. Output Value	The output when the AT is at 100% output rate
S3 +15	The Mini. Output Value	The output when the AT is at 0% output rate

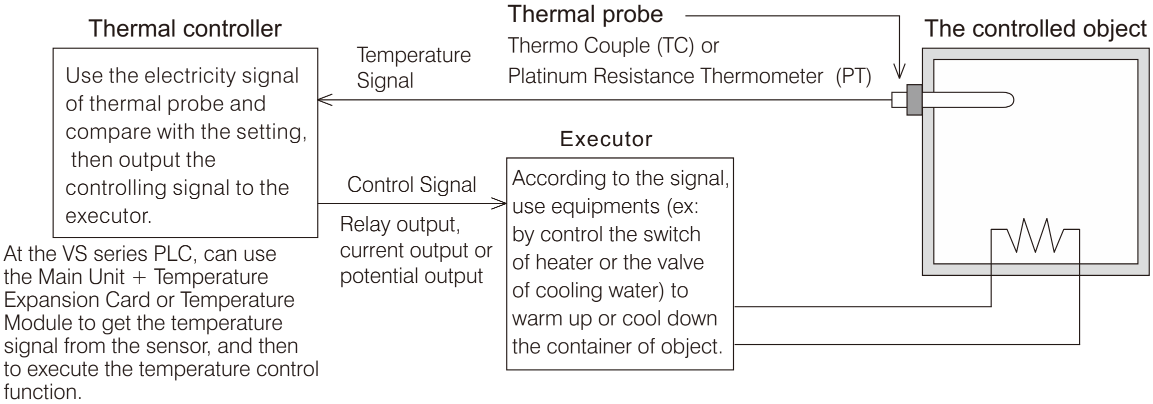
③ Let the parameter of S3+1's b4 = "ON", then it will start to execute the Auto-Tuning operation.

④ When the Auto-Tuning operation is finished, the parameters at S3+1's b4 will turn "OFF" also the contents of S3+14 and S3+15 will reset to zero automatically.

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 brief explanation of the thermal control

To set up the value of set point in the temperature controller and let it operates. 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 strong reaction, should make the response slow.

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.

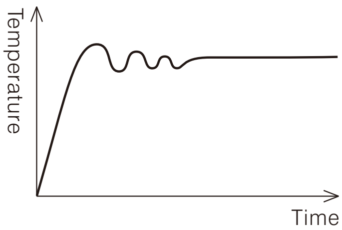


Chart (1), the Overshooting and Waving Response

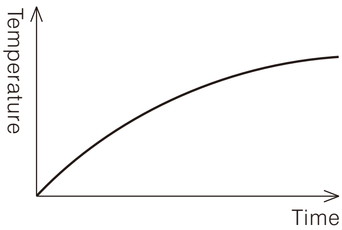


Chart (2), the Inert Response

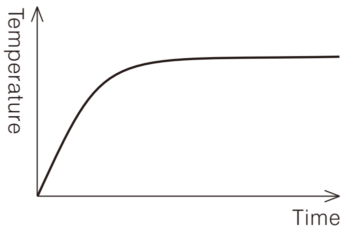


Chart (3), the Ideal Response

- 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.

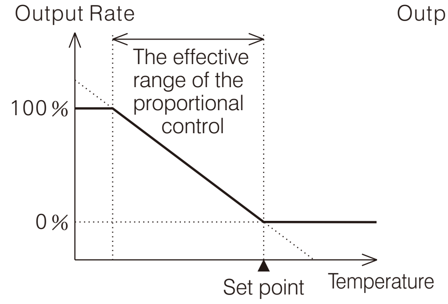
(4) External Disturbances : Some of the interference changes the temperature. For example, a door of the constant temperature furnace is opened.

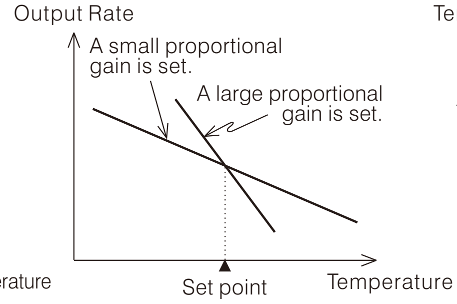
- The PID Parametric Explanations

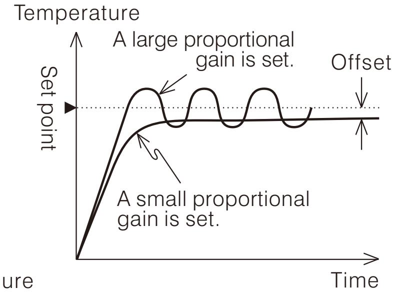
(1) P (Proportional Control) Action

The control action is used for obtaining the output in proportion to the input.

This control operation is to generate an output rate that is related to the difference between the set point and the measured temperature, by way of the multiplication product of the difference and the proportional gain. Therefore, the control operation rate related to the difference and the gain is called the proportional control.

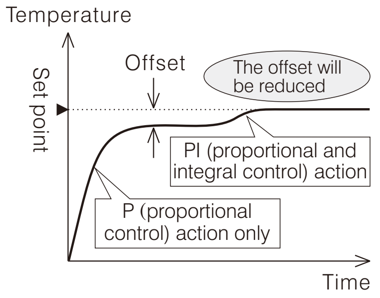


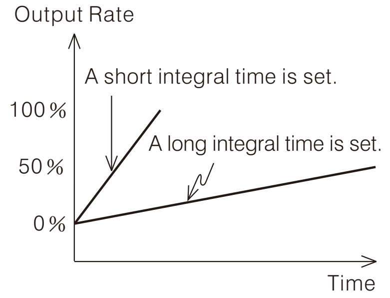


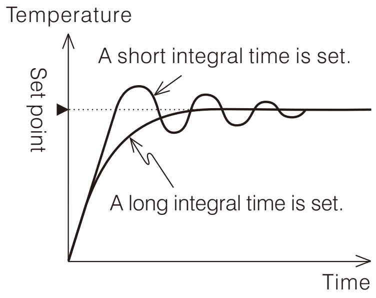


(2) I (Integral Control) Action

The I (integral) action is the output rate adjustment by the content value of Integral time constant (T_I).
The P action will cause an offset definitely. 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 be reduced.

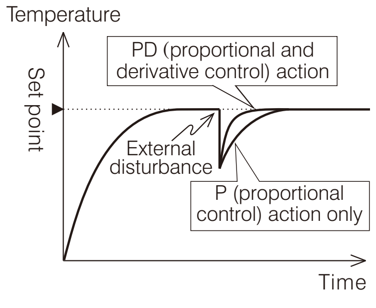


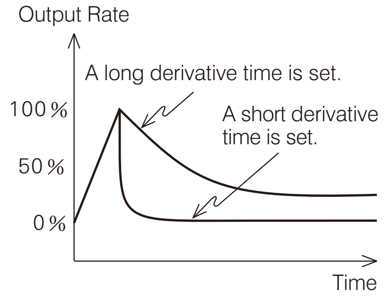


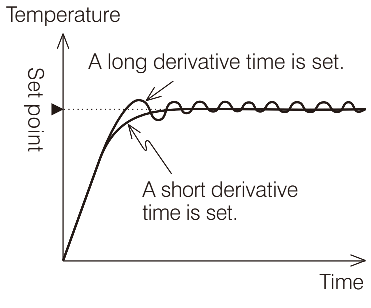


(3) D (Derivative Control) Action

The D (derivative) action is the output rate adjustment by the content value of Derivative time constant (T_D).
It provides a rapid output rate response when the measured temperature is changed.
The action of the proportional control and the integral control are by way of to correct the control result. Therefore, the response of proportional and integral control actions are slow when the temperature change is quick, 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.

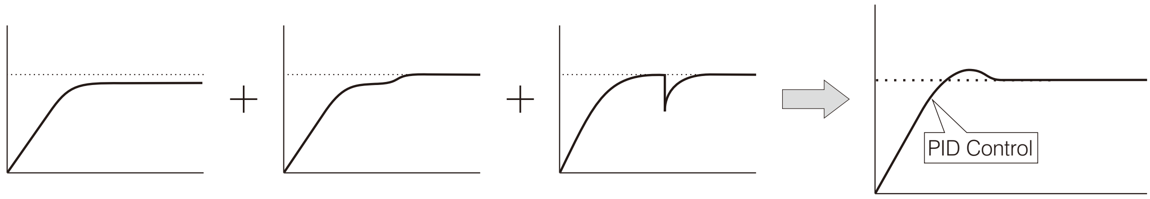






(4) PID Control

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.

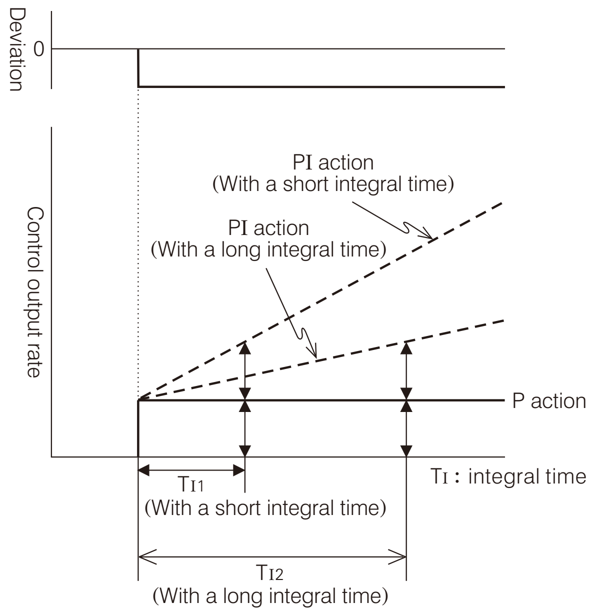


• Control Cycle and Time-Proportioning Control Action

When the temperature control is used with a relay or SSR to control the output, it will follow the designated cycle time to intermittently turn “ON” or “OFF” in a specified period. 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.

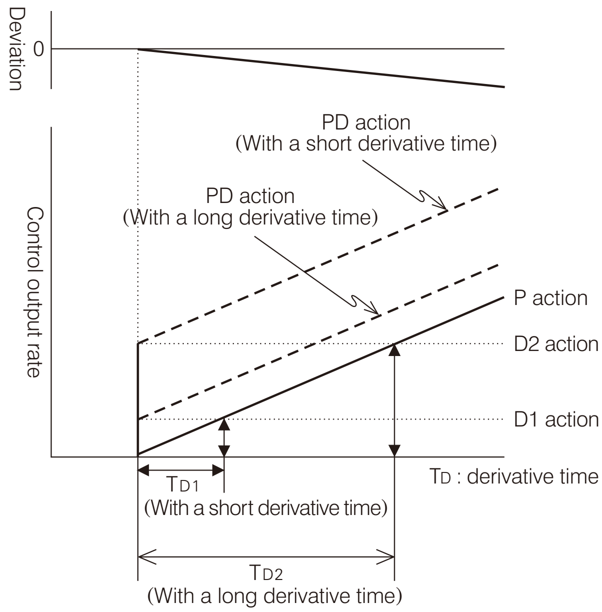
• The Definition of Integral Time

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 will cause the stronger integral control action. If the integral time is too short, it will cause a quick and huge correction then may have the temperature waving.



• The Definition of Derivative Time

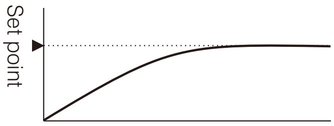
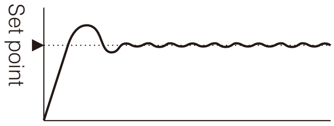
Derivative time is the period required for a ramp-type deviation in derivative control (e.g., the deviation shown in the following graph) to coincide with the control output in proportional control action. The longer the derivative time will cause the stronger derivative control action.



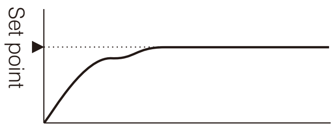
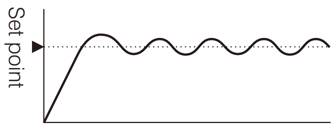
- Auto-Tuning
  All PID process/temperature controllers require the adjustment of the P, I, D and other parameters in order to allow accurate control of the load. There have been a variety of conventional methods but the Auto-Tuning methods make it possible to obtain PID constants suitable to a variety 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.

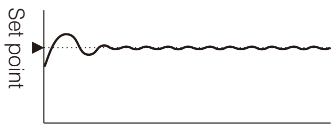
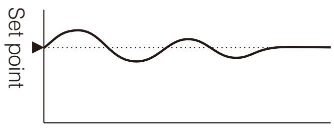
Response to Change in Proportional Gain

Narrower		It is possible to suppress overshooting although a comparatively long startup time and reaction time will be required.
Wider		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.

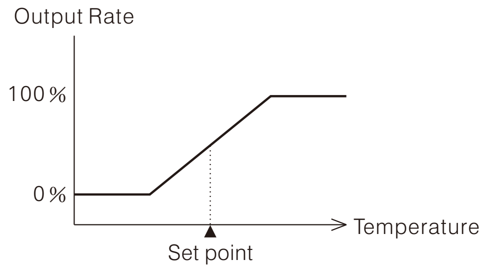
Response to Change in Integral Time

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.

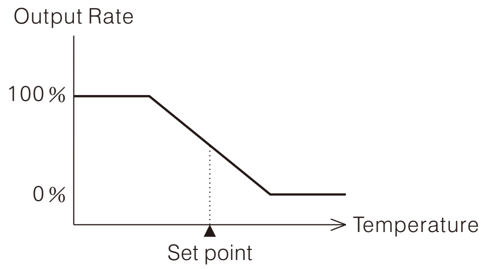
Response to Change in Derivative Time

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.

- The Definition of the Forward Operation
  To increase the control output rate when the temperature of the object is higher than the set point (positive deviation). Usually using in a cooling control.

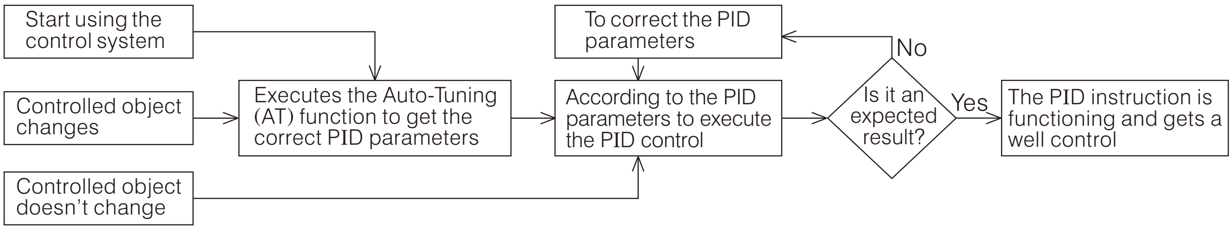


- The Definition of the Reverse operation
  To increase the control output rate when the temperature of the object is lower than the set point (negative deviation). Usually using in a heating control.

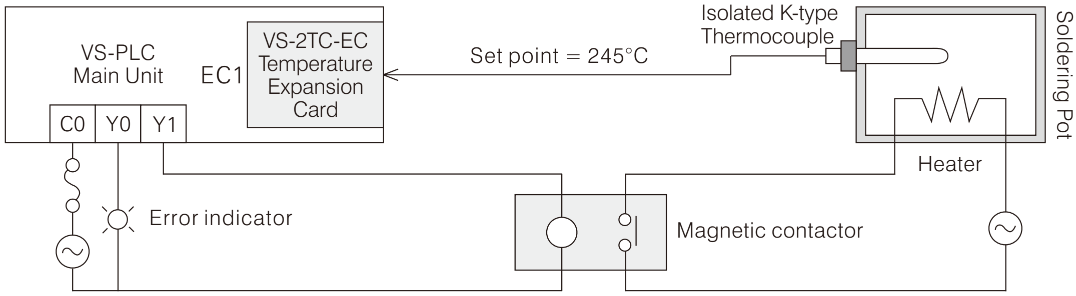


The Example of PID Temperature Control

- When design a PID temperature control program, the method below is the recommendable procedure to perform the PID instruction.

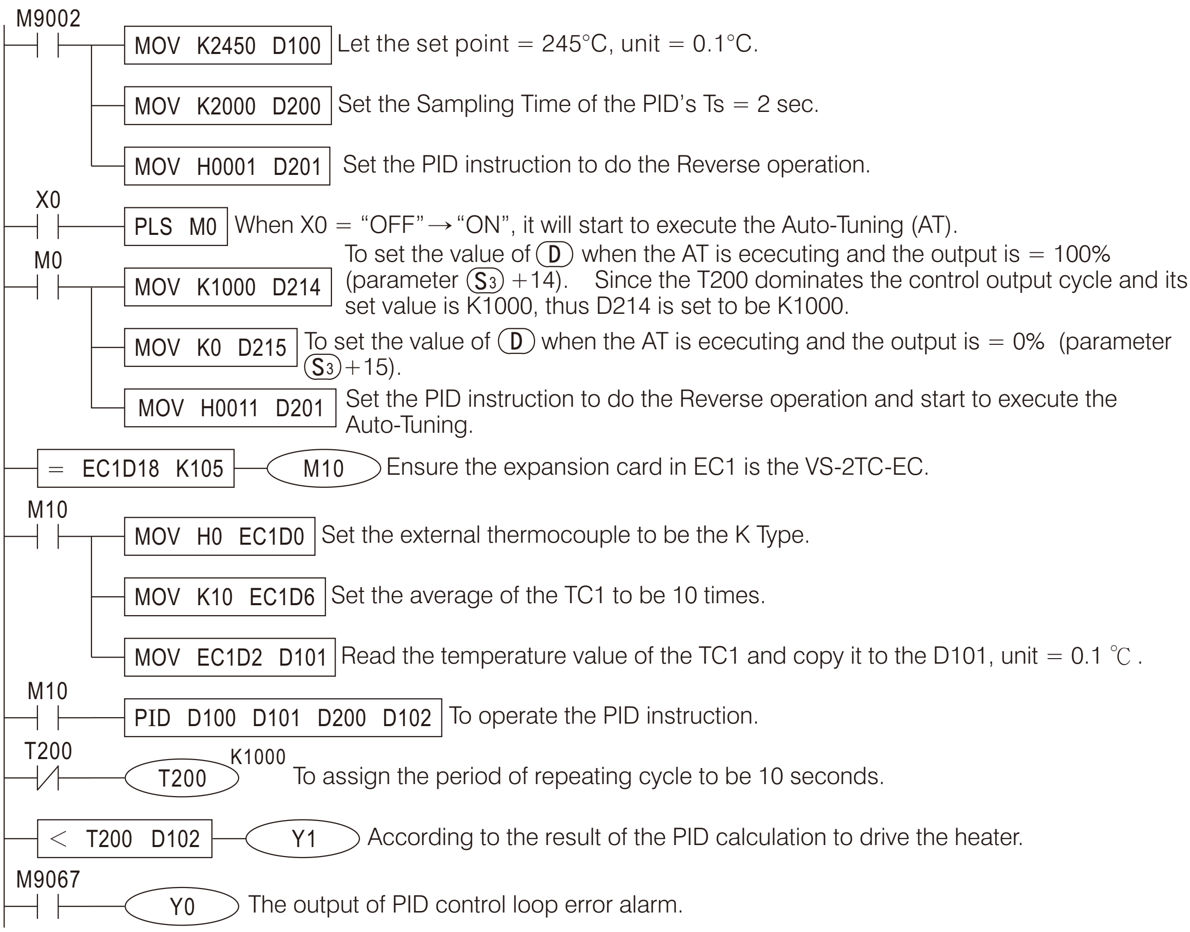


- The System Structure of Temperature Control



- 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 (5 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 89	LINK S1 S2 n	Easy Link Communication Instruction	1	2	M	3
			○	○	○	○

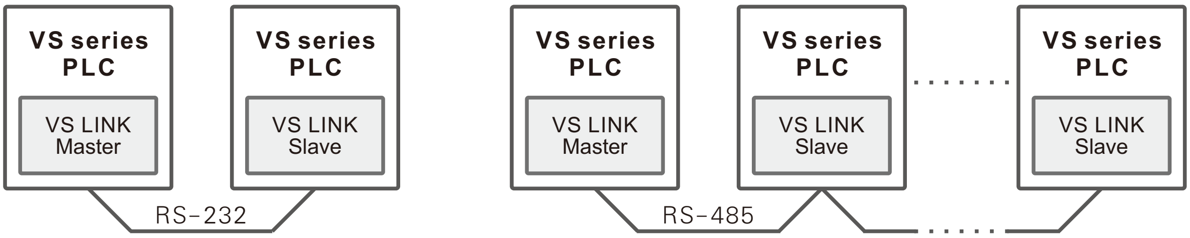
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
n																○		
• For the VS1, VS2 or VSM series, n = 1~3; for the VS3 series, n = 1~5																		

S1 : the communication table that describes the data receiving and transmitting

S2 : the working area for the instruction that occupies 4 registers

n : to assign the communication port, 1~5 = CP1~CP5

- The VS series PLC uses this instruction to transmit or get the designated data via its Communication Port CP1~CP5 with other VS PLCs.
- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the "Application type:" become the "VS Computer Link Master". Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the "Project" -- "COM Port Setting".
- This instruction could be used to transfer for various devices that is including all the Y, M, S, T, C, D and R (except for special M and special D).
- As shown in the figure below, select one of these linked PLCs as the Master station and the rest as the Slave stations. First, use the Ladder Master S to orderly open every Slave PLC's project and set its "Application type:" as the "VS Computer Link Slave" then to set an unique station No. (ranging from 0~254). After that, connect to the programming port of every PLC to load the project to each Slave PLC.
  Next, write this instruction in the Master PLC's program and set its "Application type:" as the "VS Computer Link Master", then edit the LINK communication table to designate data transferring actions. After that, to load this project to the Master PLC. Following these procedures, the purpose of data delivery between the PLCs can be reached.



- When X20 = "ON", the LINK instruction starts execution. According to the contents described in the "CP1_DATA_TRANS" communication table, it proceeds data write or read actions to the appointed Slave PLC stations, while the D100~D103 will be occupied to store the state of instruction execution.
- When the contents of the communication table specified by the S1 are execution completed from beginning to end, the M9104 will be "ON" for a scan time then the procedure will start over again from the first item set of the table.
- When the X20 turns from "ON" to "OFF", this instruction stops and the data sharing immediately stop but data which has transferred previously will still remain.
- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
■M9104	CP1 LINK / MBUS instruction on execution table complete once flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
■M9114	CP2 LINK / MBUS instruction on execution table complete once flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
■M9124	CP3 LINK / MBUS instruction on execution table complete once flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
■M9134	CP4 LINK / MBUS instruction on execution table complete once flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.
■M9144	CP5 LINK / MBUS instruction on execution table complete once flag.

• The communication table is assigned by the (S1) of the instruction.

An example of the "LINK" provided by programming tool the Ladder Master S is shown below.

Item No.	Command	Device at Master	Direction	Slave St. No.	Device at Slave	Length	Word / Bit	Disable Contact
1	Read	D0	<--	1	D0	10	—	M1
2	Write	D10	-->	1	D10	5	—	M1
3	Read	D20	<--	2	D0	10	—	M2
4	Write	D30	-->	2	D10	5	—	—
5	Write	D40	-->	255	D40	20	—	M0

The first item set in the table means that the Master reads the data D0~D9 from the Slave #1 and stores them at the D0~D9 of the Master.

The second item set in the table means that the Master writes its data at D10~D14 to the D10~D14 of the Slave #1.

The third item set in the table means that the Master reads the data D0~D9 from the Slave #2 and stores them at the D20~D29 of the Master.

The fourth item set in the table means that the Master writes its data at D30~D34 to the D10~D14 of the Slave #2.

The fifth item set in the table means that the Master writes its data at D40~D59 to all the D40~D59 of the Slaves. To set the Slave station #255 means that the Master writes the data to all of the Slaves. Note that Read Command cannot be used in this application.

The last column of the table is for the Disable Contact. If that specified contact is "ON", the communication item set will be ignored. For example, if M1 = "ON", the first and second item sets in the table will skip. This is the new function of the VS series, which can help designers effectively manage the operation of the communication table. A communication item set does not need to specify a Disabled Contact (such as the fourth item set), so that the command of the set does not have the disable control function.

• The working area of the executive instruction is starting from the (S2) (using D100~D103 as the example).

<table>
<tr><th>(S2)</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">D100</td><td>Lower 8 bits</td><td>The record of the Slave's station number when the first communication error is occurred</td></tr>
<tr><td>Upper 8 bits</td><td>Instruction working status
0: Normal data transmitting / receiving
4: The error is caused by the assigned device is inappropriate
7: The error is caused by the communication command is incorrect
A: The communication setting is normal but no response from the Slave station (Time-out occurs)
B: Abnormal communication</td></tr>
<tr><td>D101 ~ D103</td><td colspan="2">The working area is required for the system when this instruction is performed</td></tr>
</table>

The LINK instruction will reset the D100 to be 0 before the first item set at the table is executed.

If any communication error occurs during the execution of the instruction, a code will be recorded in D100. Only when the content value of D100 is 0, the recording action can be executed. Therefor, when there are possibilities of several errors, users can use the program to move out the content value of D100 then reset it to 0. This way allows the D100 to record the next error.

• To edit a communication table
Use the Ladder Master S to set up a LINK communication table and through its interactive window can set up and edit a communication table easily.
In the structure of VS series PLC, the communication tables are a part of the project. When the programmer to copy or access the project, those tables will be duplicated automatically with the program.

MEMO

6-11 Handy Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
90	DBRDP S D	Read Data From Data Bank	○	○	○	○
91	DBWRP S D	Write Data Into Data Bank	○	○	○	○
92	TPID S1 S2 S3 S4 n	Temperature PID Control	○	○	○	○
93	DTRDP S1 S2 D n	Read Data From Data Table	○	○	○	○
102	ZPUSHP D	Batch Store of All Index Register				○
103	ZPOPP D	Batch Recover of All Index Register				○

FNC 90	DBRDP (S) (D)	Read Data From Data Bank	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													○			●		
D													●					
n													○			○		

- S is a 32-bit component, S = 0~655,359
  When S designates to D or R as the object, that occupies 2 registers.
  When S uses K or H and modifies by V, Z index, that occupies a pair of V, Z index registers.
- D is a 16-bit component, but when uses V, Z index to modify, that occupies a pair of V, Z index registers.
- n = 1~32,767

```
             (S)   (D)    (n)
 X20
|--| |----[DBRDP  D0  D1000  K100]
```

S : the initial address number at the Data Bank to be read
D : the initial register ID number to store the read data
n : the length of the data to be read (Unit: Word)

- Then X20 = "OFF"→"ON" and (D1, D0) = 100, it will execute to read a group of data at the addresses 100~199 in the Data Bank and put the data in D1000~D1099. Due to the (n) = 100, it will read a group data that contents with 100 words.
- When a Main Unit is equipped with a VS-MC or VS-MCR memory card, it will be provided with the Data Bank. After that, for the application which need to store or get a large number of data is convenient.
- The Data Bank in the VS-MC or VS-MCR card is stored in the component of Flash ROM. It can store 655,360 words of data and there has no limit on the number of reading times.
- When X20 = "OFF", the instruction will not be performed but the data (which has read previously) will still remain.

FNC 91	DBWRP (S) (D)	Write Data Into Data Bank	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●					
D													○			●		
n													○			○		

- S is a 16-bit component, but when uses V, Z index to modify, that occupies a pair of V, Z index registers.
- D is a 32-bit component, D = 0~655,359
  When D designates to D or R as the object, that occupies 2 registers.
  When D uses K or H and modifies by V, Z index, that occupies a pair of V, Z index registers.
- n = 1~32,767

```
X20            (S)    (D)  (n)
─┤├──────DBWRP D1000  D0   K100
```

S : the initial register ID number for a group of data that writes to the Data Bank

D : the initial address number at the Data Bank to be write

n : the length of the data to be write (Unit: Word)

- When X20 = "OFF"→"ON" and (D1, D0) = 100, it will execute to read a group of data at the Main Unit's D1000~D1099 and store the data into the addresses 100~199 in the Data Bank. Due to the (n) = 100, it will write a group data that contents with 100 words.
- When a Main Unit is equipped with a VS-MC or VS-MCR memory card, it will be provided with the Data Bank. After that, for the application which need to store or get a large number of data is convenient.
- The Data Bank in the VS-MC or VS-MCR card is stored in the component of Flash ROM. Even though, in every section 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 this DBWR instruction to rewrite data into the card, better change it to the DBWRP instruction. The DBWRP can avoid useless operate of rewrite, and then extend the lifespan of the Flash ROM.
- When the instruction rewrites data to the memory card, every section (128 words) needs 15ms to execute the function. The CPU will wait for the execution to complete and then continuously perform the rest of the program. At that moment, the current value of the Watch Dog Timer will be automatically extended. Thus, when using this instruction to store a large amount of data, should pay attention to the effect which write action may cause.

FNC 92	TPID (S1) (S2) (S3) (S4) (n)	Temperature PID Control	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1													○					
S_2													○					
S_3													○					
S_4													○					
n																○		

- S_1 and S_2 individually occupie n consecutive registers
- S_3 occupies $(10\times n)+10$ consecutive registers
- S_4 occupies $6\times n$ consecutive registers
- $n = 1\sim16$

```
                  (S1)    (S2) (S3)   (S4)   (n)
 X0
─┤ ├──────TPID  D7000  D0   D100  D7100  K16
```

S_1 : the initial register ID of the Set Value (SV) block

S_2 : the initial register ID of the Present Values (PV) block

S_3 : the initial register ID of the parameters of instruction & outputs

S_4 : the initial register ID of the parameters of PID & other Set 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.
- The instruction uses the difference between (S1) (one in the set value block) and (S2) (correlated one in the present value block), then via the content 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 result 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 allows to be used in a subroutine, interrupted subroutine, step ladder diagram or related to the conditional jump instruction. However, at the time that the PID instruction is active, must make sure this instruction can be processed once and only once at every Scan Time. If this instruction has been processed more than once or had not been executed, that will cause the sampling error or imprecise operation.
- The specification of Set 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 Set Value (SV) for the first object channel; the content value of (S1)+1 is the Set Value (SV) for the second object channel; and so on.
- The specification of the measured 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 description of the parameter block S_3 :

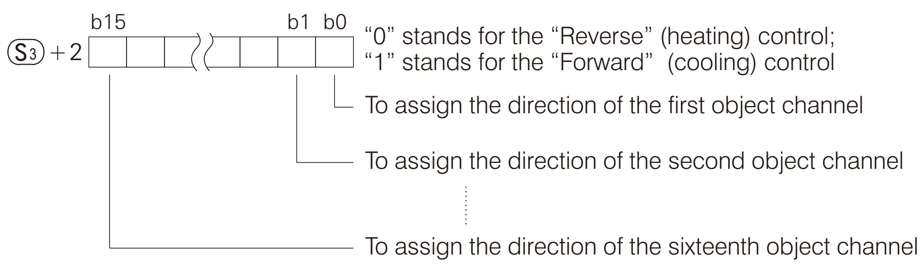
Para- meter	Parameter Name/Function	Description	Available Range
S_3	Control Cycle Setting	To assign the outputs period interval (the length of one ON/OFF cycle)	10～32767 × 10ms
S_3+1	Responsive Sensitivity	To assign the responsive level of the instruction which is for all channels (“0”: Fast / .. / “3”: Slow	0～3
S_3+2	Operational Directions	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 bits	By relative bits at this register to assign the control method of channels (“0”: Automatic / “1”: Manual)	H0000～HFFFF
S_3+4	AT Command bits	By relative bits at this register to start the Auto-Tuning (AT) function of channels (“1”: AT; automatically reset to "0" when AT is 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 objects of this instruction.
  And, to add each object channel will occupy extra 10 registers, those object channels will occupy form S_3+10 to S_3+(10×n+9).

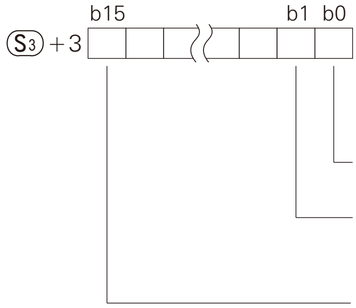
- The parameter at S_3 is the control output period setting for this instruction.
  Usually, the length of control period depends on the type of loading.
  If the the equipment is driven by a Magnetic Contactor (MC), to set the value bigger than 1000 (1000×10ms. = 10 Sec.) is recommend that is for extend 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 response to get a better control.
  The value 0~3 in this parameter represents the level of responsive sensitivity from the fast “0” to the slowest “3”. Which “3” has the least possibility to cause temperature waving.

- Each bit at S_3+2 is for set up control direction of every single object channel.
  When the measured Present Value (PV) < the Set Value (SV), it will generate a negative deviation and increase the control effect, that is called the “Reverse” operation. E.g. An oven: before the heater of the oven turns on. Usually the temperature of the oven is lower than the Set value. (PV) < (SV), this is a typical “Reverse” operation control sample.
  When the measured Present Value (PV) > the Set Value (SV), it will generate a positive deviation and increase the control effect, that is called the “Forward” operation. E.g. An air conditioning system: before the system turns on. Usually the indoor temperature is higher than the Set value. (PV) > (SV), this is a typical “Forward” operation control sample.



• Each bit at S_3+3 is for set up Auto/Manual control of every single object channel.



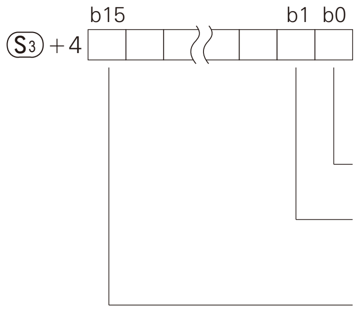
"0" stands for PID automatic control; "1" stands for manual control.
When using the manual control mode, should input the expected output value (0 ~ 1000) directly to relative register.

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 S_3+4 is for trigger the Auto-Tuning (AT) function of every single object channel.



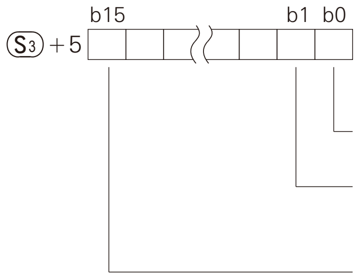
To trigger the b0 "ON", the first object channel will start to process the AT function; when its AT function finished, the b0 will be reset to "OFF" and the parameters P (K_P), I (T_I) and D (T_D) will put into relative registers in S_4 block.

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 S_3+5 is for output the "ON" / "OFF" control signal of every single object channel.



By the S_3 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

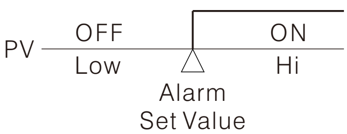
To output the control signal of the sixteenth object channel

Each object channel will also generate an analog PID output value.
The results are output to S_3+(10×m) (m = 1~n). E.g. The S_3+10 is the output value of the first object channel; the S_3+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 S_3+5 outputs are using those values in S_3+(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.

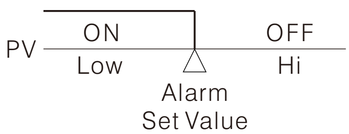
• This instruction provides two alarm signals for each object channel. See the illustrations below.

① Limitation Alarm

When a object channel uses the "Reverse" operation, the Limitation Alarm will "ON" if the PV is higher then the alarm set value.

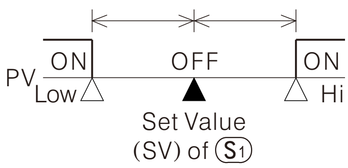


When a object channel uses the "Forward" operation, the Limitation Alarm will "ON" if the PV is lower then the alarm set value.



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.

b15 b1 b0

S_3+6

The status of Limitation Alarm from the first object channel

The status of Limitation Alarm from the second object channel

The status of Limitation Alarm from the sixteenth object channel

A object channel has a set value of Limitation Alarm which is put in S_4+(6×m +4) (m = 1~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.

b15 b1 b0

S_3+7

The status of Deviation Alarm from the first object channel

The status of Deviation Alarm from the second object channel

The status of Deviation Alarm from the sixteenth object channel

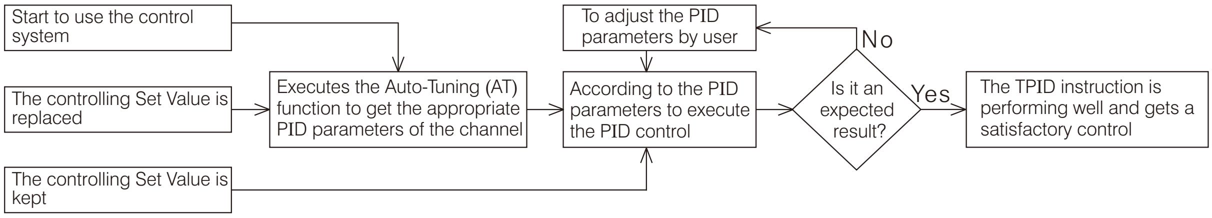
An object channel has a set value of Deviation Alarm which is put in S_4+(6×m +5) (m = 1~n–1); the S_4+5 is for the first object channel; the S_4+11 is for the second object channel; and so on.

● The exposition of parameter block (S4) (K_P, T_I, T_D & other set values)

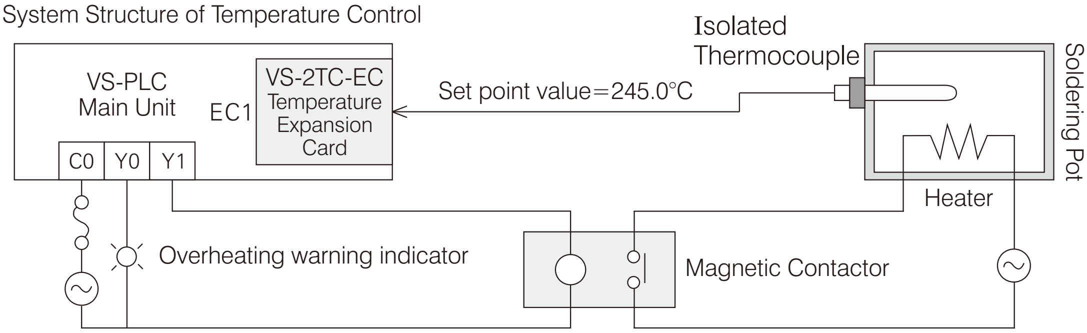
Para-meter	Parameter Name/Function	Description	Available Range
S_4	Proportional Gain (K_P) of the First Object Channel	The P (Proportional) part of the PID loop	1～32767×0.01
S_4+1	Integral Time Constant (T_I) of the First Object Channel	The I (Integral) part of the PID loop (this parameter could disable the effect of the I if it is “0”)	0～32767×100ms
S_4+2	Derivative Time Constant (T_D) of the First Object Channel	The D (Derivative) part of the PID loop (this parameter could disable the effect of the D if it is “0”)	0～32767×10ms
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 Set Value of the First Object Channel	For the “Reverse” operation: Limitation Alarm “ON” if PV > this set value. For the “Forward” operation: Limitation Alarm “ON” if PV < this set value.	–32768～32767
S_4+5	Deviation Alarm Set Value of the First Object Channel	Deviation Alarm “ON” if PV > (SV + this set value) or PV < (SV − this set 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 could disable the effect of the I if it is “0”)	0～32767×100ms
S_4+8	Derivative Time Constant (T_D) of the Second Object Channel	The D (Derivative) part of the PID loop (this parameter could disable the effect of the D if it is “0”)	0～32767×10ms
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 Set Value of the Second Object Channel	For the “Reverse” operation: Limitation Alarm “ON” if PV > this set value. For the “Forward” operation: Limitation Alarm “ON” if PV < this set value.	–32768～32767
S_4+11	Deviation Alarm Set Value of the Second Object Channel	Deviation Alarm “ON” if PV > (SV + this set value) or PV < (SV − this set value)	–32768～32767
⋮	⋮	⋮	⋮

- The register block starting from (S4) is for storage every channel’s K_P, T_I, T_D parameters, starting Overshoot Repression and two alarm set values.
  Every channel will occupy 6 sequential registers. The (S4)～(S4)+5 keep parameters for the first channel; the (S4)+6 ～ (S4)+11 keep parameters for the second channel; and so on.
- Registers for the block of (S4) 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 (S4) , please refer to the instruction of (S3)+6 and (S3)+7.

- **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.

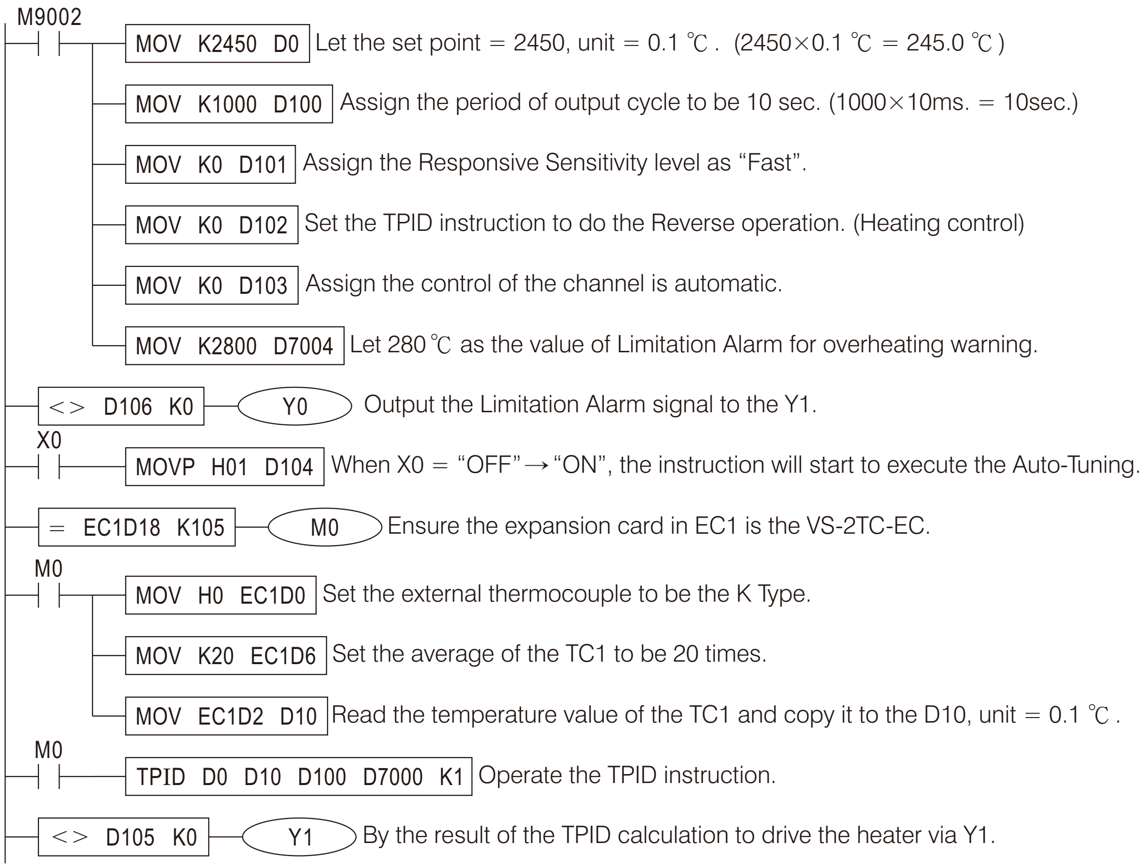


- The System Structure of Temperature Control



- 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**

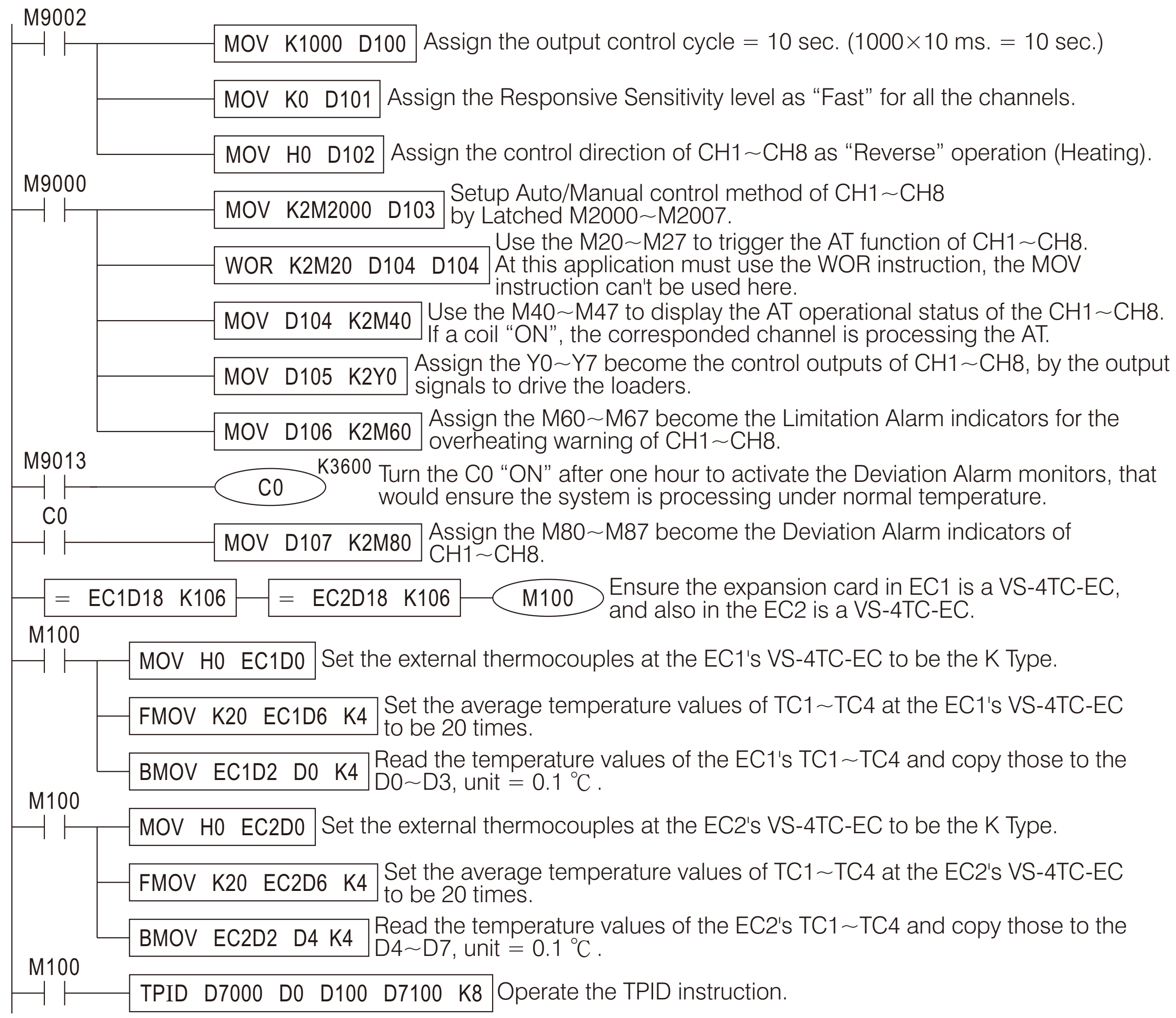
  Below provides an example of an 8 channel PID temperature control. This needs a 32 point VS2 series Main Unit and 2 of the VS-4TC-EC expansion cards. Also, a human-machine interface (HMI) is required to proceed for data setting and status display.

- The list of devices used in this example:

Controlled CH #	8	7	6	5	4	3	2	1
Auto/Manual Select	M2007	M2006	M2005	M2004	M2003	M2002	M2001	M2000
AT Command	M27	M26	M25	M24	M23	M22	M21	M20
AT Status	M47	M46	M45	M44	M43	M42	M41	M40
Output Point	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Limitation Alarm Status	M67	M66	M65	M64	M63	M62	M61	M60
Deviation Alarm Status	M87	M86	M85	M84	M83	M82	M81	M80
Temp. Set Value (SV)	D7007	D7006	D7005	D7004	D7003	D7002	D7001	D7000
Temp. Present Value (PV)	D7	D6	D5	D4	D3	D2	D1	D0
Parameter of P Phase (K_P)	D7142	D7136	D7130	D7124	D7118	D7112	D7106	D7100
Parameter of I Phase (T_I)	D7143	D7137	D7131	D7125	D7119	D7113	D7107	D7101
Parameter of D Phase (T_D)	D7144	D7138	D7132	D7126	D7120	D7114	D7108	D7102
Overshoot Repression Value	D7145	D7139	D7133	D7127	D7121	D7115	D7109	D7103
Limitation Alarm Value	D7146	D7140	D7134	D7128	D7122	D7116	D7110	D7104
Deviation Alarm Value	D7147	D7141	D7135	D7129	D7123	D7117	D7111	D7105

Besides the devices on the table above, this instruction will occupy the registers D100~D189. When actually use this instruction, some unnecessary control items (e.g. Auto/Manual control selection) could be removed from the program then those items would not occupy components.

- Program Example



FNC 93	DTRDP S1 S2 D n	Read Data From Data Table	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													●			●		
D													●					
n													●			●		

```
                S1                        S2     D      n
 X20
─┤ ├────┤ DTRDP  FORMULA_TABLE  K0  D1000  K100 ├
```

S1 : the Table Code or Table Nickname of the data table

S2 : the initial address number at the table to be read

D : the initial register ID number to store the read data

n : the length of the data to be read (Unit: Word)

- This instruction in the VS series PLC reads the specific data from the data table. And then, stores the data into given Data Registers to provide as the reference source for program operation.

- When X20 = "OFF"→"ON", it will execute to read a group of data at the addresses 0~99 in the "FORMULA_TABLE" and put the data in D1000~D1099. Due to the n = 100, it will read a group data that contents with 100 words.

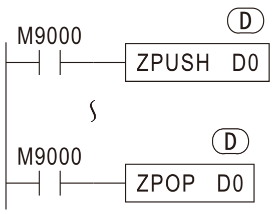
- At the previous VB series PLC, it has the File Register. Usually, the user can preset a huge number of data in the File Register and while the status of the PLC turns from STOP to RUN the system will duplicate the contents from a block of the File Register to the specific general registers automatically. That is very useful for the machine's parameters or reference data.

  Compared with that, the VS series substitute the File Register to the Data Table to store a huge of preset data for the parameters and the reference data. The new structure of Data Table can have many tables and be named for each purpose. It has the advantage about easy to manage and access.

- The Data Table is established by the programming tool Ladder Master S, it is a part of the project. Therefore, when the project is installed to the PLC, all the Data Tables will be copied together.

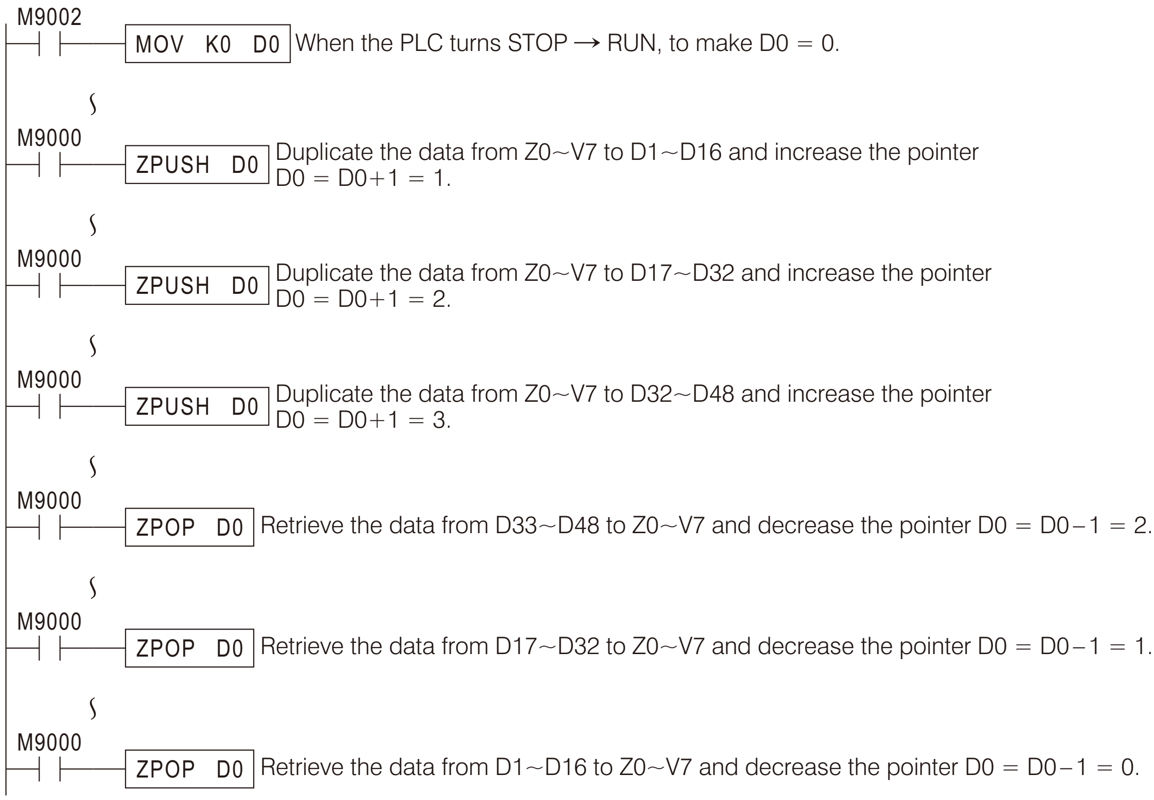
FNC 102	ZPUSH P (D)	Batch Store of All Index Register	1	2	M	3
						○
FNC 103	ZPOP P (D)	Batch Recover of All Index Register	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D													○					



D : the pointer to manage the packed location of the Index Registers

- The VS series PLC has 16 Index Registers (Z0~Z7 and V0~V7). If those are reused in the program and may disturb to the results of each other, it is necessary to manage the Index Registers.
  The purpose of the ZPUSH and ZPOP instructions are to manage the Index Registers conveniently.
  To use these two instructions must be paired, the ZPUSH is to store the value of registers and the ZPOP is to recover the data.
- Firstly, let the D0 = 0. When the ZPUSH instruction is executed, it will duplicate the content values of Z0~V7 to the D1~D16 and increase the value at D0 become 1 that means one set of Index Registers is stored.
  When the ZPOP instruction is executed, it will retrieve the content values of D1~D16 back to the Z0~V7 and decrease the value at D0 from 1 to 0 that means no preserved data for Index Registers.
- Below is an example to appoint the data storage by D0. The ZPUSH instruction will store all the data of Index Registers into the specified registers; the ZPOP instruction will restore the data from the specified registers.



The most common uses of the ZPUSH and ZPOP instructions are in the programming structure of the nest and call subroutine.

- Please note that occupied space of the instruction depends on the content value of (D) and the value is driven by the using times of call and nest levels.

6-12 Floating Point Arithmetic Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
110	DECMPP S1 S2 D	Compare Two BIN Floating Point Numbers	○	○	○	○
111	DEZCPP S1 S2 S D	Compare a BIN Float Number to BIN Float Zone	○	○	○	○
112	DEMOVP S D	Move Floating Point Data	○	○	○	○
116	DESTRP S1 S2 D	Convert BIN Floating Point to Character String				○
117	DEVALP S D	Convert Character String to BIN Floating Point				○
118	DEBCDP S D	Convert BIN to DEC Floating Point Format	○	○	○	○
119	DEBINP S D	Convert DEC to BIN Floating Point Format	○	○	○	○
120	DEADDP S1 S2 D	BIN Floating Point Addition	○	○	○	○
121	DESUBP S1 S2 D	BIN Floating Point Subtraction	○	○	○	○
122	DEMULP S1 S2 D	BIN Floating Point Multiplication	○	○	○	○
123	DEDIVP S1 S2 D	BIN Floating Point Division	○	○	○	○
124	DEXPP S D	BIN Floating Point Number Exponent	○	○	○	○
125	DLOGEP S D	BIN Floating Point Nature Logarithm	○	○	○	○
126	DLOG10P S D	BIN Floating Point Common Logarithm	○	○	○	○
127	DESQRP S D	BIN Floating Point Square Root	○	○	○	○
128	DENEGP D	BIN Floating Point Negation	○	○	○	○
129	DINTP S D	BIN Floating Point → BIN Integer Format	○	○	○	○
130	DSINP S D	Calculate Sine	○	○	○	○
131	DCOSP S D	Calculate Cosine	○	○	○	○
132	DTANP S D	Calculate Tangent	○	○	○	○
133	DASINP S D	Calculate Arc Sine	○	○	○	○
134	DACOSP S D	Calculate Arc Cosine	○	○	○	○
135	DATANP S D	Calculate Arc Tangent	○	○	○	○
136	DRADP S D	Convert Angle From Degrees to Radian	○	○	○	○
137	DDEGP S D	Convert Angle From Radian to Degrees	○	○	○	○

FNC 110	DECMP P S1 S2 D	Compare Two BIN Floating Point Numbers	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●		●	●	○	
S2													●		●	●	○	
D		●	●	●	○	○												

• D occupies 3 consecutive devices

```
        S1   S2   D
X0
|—| |——[DECMP  D0  D10  M0]
```

S1 : the compare value #1

S2 : the compare value #2

D : the compare result; occupying 3 consecutive points

- Compare the content value of S1 (compare value #1) with the value of S2 (compare value #2), and save the result in D (compare result).
- The DECMP instruction will be enabled when X0 = “ON”, and all the content values are used by the BIN floating point format.
  If S1 > S2 (D1, D0 > D11, D10), then M0 = “ON”;
  If S1 = S2 (D1, D0 = D11, D10), then M1 = “ON”;
  If S1 < S2 (D1, D0 < D11, D10), then M2 = “ON”.
- When X 0 = “OFF”, the instruction is disabled, the “ON”/“OFF” status of M0, M1 and M2 remains the same as the status before X0 = “OFF”.
- This instruction is a 32-bit instruction. Therefore, be sure to input DECMP or DECMPP in the program.
- Please use serial or parallel links of M0~M2 to generate the result as “≥”, “≤” or “≠”.
- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the comparison function.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 “Numerical System” for the reference of the floating point number.

FNC 111	D EZCP P S1 S2 S D	Compare a BIN Float Number to BIN Float Zone	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●		●	●	○	
S2													●		●	●	○	
S													●		●	●	○	
D		●	●	●	○	○												

• D occupies 3 consecutive devices • $S_1 \leqq S_2$

```
        S1  S2  S    D
X0
|—| |——DEZCP D0  D2  D10  M0
```

S1 : the lower limit of zone compare

S2 : the upper limit of zone compare

S : the compare value

D : the compare result; occupying 3 consecutive points

- Compare the content value of S (compare value) with the values of S1 (lower limit of zone compare) and S2 (upper limit of zone compare), then save the result in D (compare result).
- The DEZCP instruction will be enabled when X0 = "ON", and all the content values are used by the BIN floating point format.
  If S < S1 (D11, D10 < D1, D0, the compare value is less than the lower limit), then M0 = "ON";
  If S1 ≤ S ≤ S2 (D1, D0 ≤ D11, D10 ≤ D3, D2, the compare value is between the upper and lower limit), then M1 = "ON";
  If S > S2 (D11, D10 > D3, D2, the compare value is bigger than the upper limit), then M2 = "ON".
- When X0 = "OFF", the instruction is disabled, the "ON"/"OFF" status of M0, M1 and M2 remains the same as the status before X0 = "OFF".
- This instruction is a 32-bit instruction. Therefore, be sure to input DEZCP or DEZCPP in the program.
- When S1 > S2 , the content value of S1 will become both upper/lower limits to be compared with the S.
- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the comparison function.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 112	DEMOVP (S) (D)	Move Floating Point Data	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

```
 X0              (S)  (D)
|-| |----[DEMOV  D0   D10]
```

S : the source device to be transferred

D : the destination device

- To copy the designated value from (S) to (D) .
- The content value of the BIN floating point format in D1, D0 will be copied to D11, D10 when the X0 = "ON".
- The instruction is disabled and D11, D10 remain invariable when the X0 = "OFF".
- This instruction is a 32-bit instruction. Therefore, be sure to input DEMOV or DEMOVP in the program.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 116	DESTRP S1 S2 D	Convert BIN Floating Point to Character String	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●		●		○	
S2							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			

S1 : the source of BIN floating point number

S2 : the devices to appoint the conversion format

D : the head ID of the converted storage devices

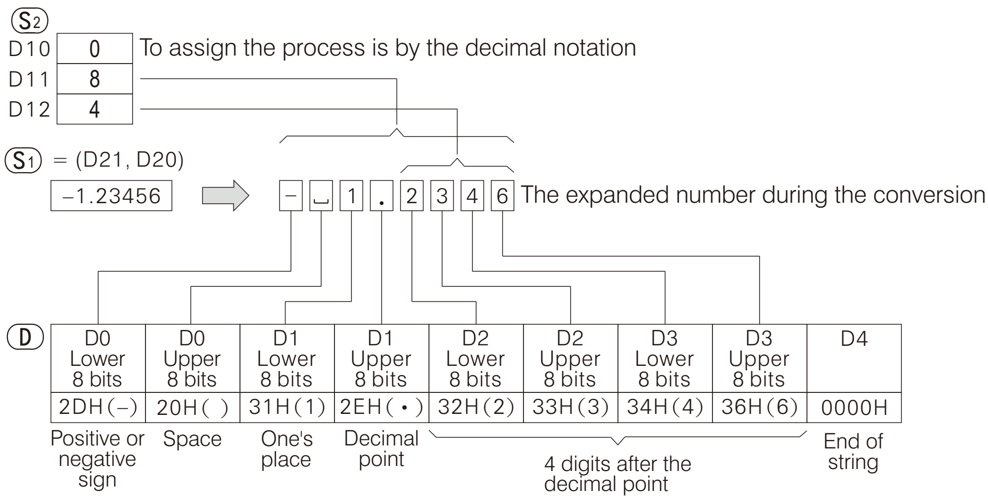
- This instruction uses a BIN floating point number at the S1 to perform the conversion. By the format parameters at the S2 to convert each digit of that number become an ASCII code and combine those codes to a string, then stores into the devices which are headed by D .

S2	Format of floating point number to display	0 : By the decimal notation; 1 : By the exponential notation
S2 + 1	Length of the converted string	2~24, the conversion could not exceed 24 characters.
S2 + 2	Number of digits after the decimal point at the string	0~7, input 0 if the conversion is used to get an integer.

- This instruction will use the following ASCII conversion table:

Sign & Number	SPACE	+	−	.	0	1	2	3	4	5	6	7	8	9	E
ASCII Cord	20H	2BH	2DH	2EH	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H	45H

- By the decimal notation:



The positive or negative sign is using the 20H to represent positive; the 2DH represents negative.

The byte next to the positive or negative sign is a blank space, that will fill in the code 20H.

If (S2+2) = 0, that indicates an integer will be shown, so there is no decimal point in the string; if (S2+2) > 0, that indicates the number has a fractional part, so the decimal point will be added to the string.

The expanded number 1.2346 is caused by to round off the content of next digit 6, hence that increases the last used digit from 5 to 6.

When the expanded number converts to the ASCII codes, the instruction adds the end of string at the tail automatically. If the set length of the string is an odd number, the end of string is 00H; in addition, the string with even number set length will add the end of string 0000H. Thus, the D4 at the converted string with the value 0000H that is the end of string.

• By the exponential notation:

S2

D10	1	To assign the process is by the exponential notation
D11	11	
D12	3	

S1 = (D21, D20)

123.4567 ⇨ ␣ ␣ 1 . 2 3 5 E + 0 2 The expanded number during the conversion

D

D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4 Lower 8 bits	D4 Upper 8 bits	D5 Lower 8 bits	D5 Upper 8 bits
20H( )	20H( )	31H(1)	2EH(•)	32H(2)	33H(3)	35H(5)	45H(E)	2BH(+)	30H(0)	32H(2)	00H
Positive or negative sign	Space	One's place	Decimal point	3 digits after the decimal point			Indicate the Exponential	Positive or negative sign for exponent	2 digits exponent		End of string

The positive or negative sign is using the 20H to represent positive; the 2DH represents negative.

It has one digit at the one's place only.

Also, the byte next to the positive or negative sign is a blank space, that will fill in the code 20H.

If ((S2)+2) = 0, that indicates an integer will be shown at the mantissa, so there is no decimal point in the string; if ((S2)+2) > 0, that indicates the number has a fractional part, so the decimal point will be added to the string. At the example above, ((S2)+2) = 3, thus the decimal point will be added automatically.

The expanded number 1.235 is caused by to round off the content of next digit 5, hence that increases the last used digit from 4 to 5.

For the part of the exponent:

The instruction adds a character “E” in the front of the exponent part automatically.

In this part, its positive or negative sign is using the 2BH to represent positive; the 2DH represents negative.

The length of the exponent number is fixed to 2 digits.

When the expanded number converts to the ASCII codes, the instruction adds the end of string at the tail automatically. If the set length of the string is an odd number, the end of string is 00H; in addition, the string with even number set length will add the end of string 0000H. Thus, the D4 at the converted string with the value 0000H that is the end of string.

• This instruction is a 32-bit instruction. Therefore, be sure to input DESTR or DESTRP in the program.

FNC 117	DEVALP S D	Convert Character String to BIN Floating Point	1	2	M	3
						○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D													●		●			

(The column heads X to "$" are all under the spanning head "Devices".)

S : the head ID of the character string to be converted

D : the storage device for the BIN floating point result

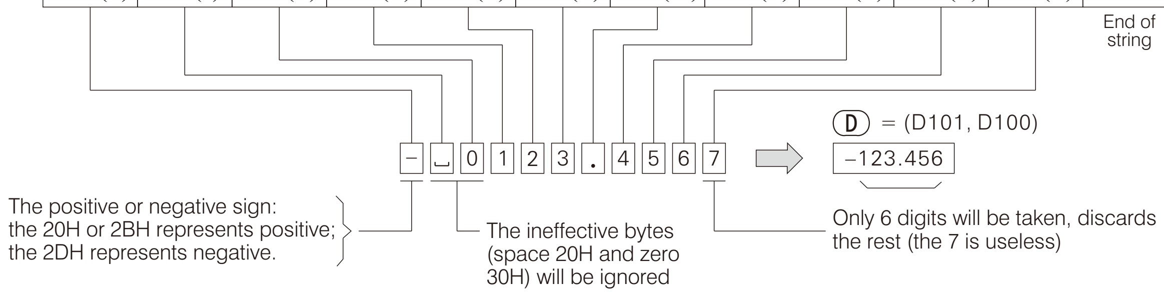
- This instruction uses a string started from the S to perform the BIN floating point number conversion, then stores the result into the device D. The format of the source string can either use the decimal notation or the exponential notation.

- This instruction will use the following ASCII conversion table:

ASCII Cord	20H	2BH	2DH	2EH	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H	45H
Sign & Number	SPACE	+	−	.	0	1	2	3	4	5	6	7	8	9	E

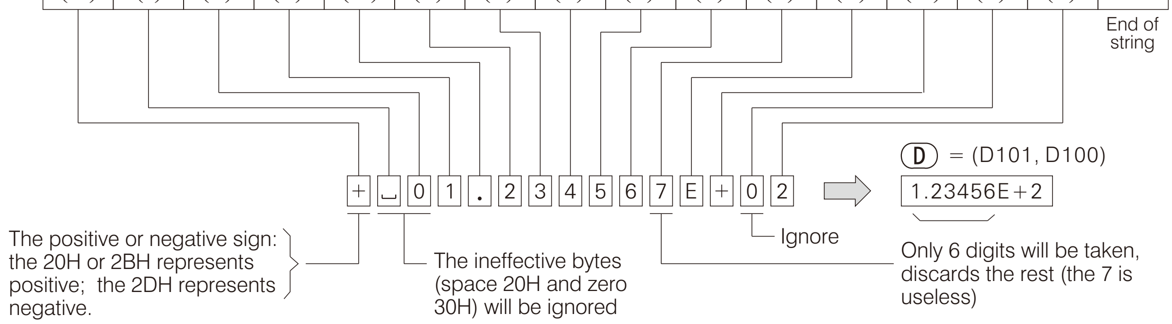
- By the decimal notation:

S	D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4 Lower 8 bits	D4 Upper 8 bits	D5 Lower 8 bits	D5 Upper 8 bits
	2DH(−)	20H( )	30H(0)	31H(1)	32H(2)	33H(3)	2EH(•)	34H(4)	35H(5)	36H(6)	37H(7)	00H



- By the exponential notation:

S	D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4 Lower 8 bits	D4 Upper 8 bits	D5 Lower 8 bits	D5 Upper 8 bits	D6 Lower 8 bits	D6 Upper 8 bits	D7 Lower 8 bits	D7 Upper 8 bits
	2BH (+)	20H ( )	30H (0)	31H (1)	2EH (•)	32H (2)	33H (3)	34H (4)	35H (5)	36H (6)	37H (7)	45H (E)	2BH (+)	30H (0)	32H (2)	00H



- The conversion could not process the string that exceeds 24 characters.

- This instruction will affect the special devices below:

Relay ID No.	Indication	Description: Cause	Description: Effect
M9020	Zero Flag	The number is equal to 0	M9020=ON
M9021	Borrow Flag	The absolute value is smaller then 2^{-126}	Make D = 2^{-126} and M9021=ON
M9022	Carry Flag	The absolute value is more then or equal to 2^{128}	Make D = 2^{128} and M9022=ON

- This instruction is a 32-bit instruction. Therefore, be sure to input DEVAL or DEVALP in the program.

FNC	Instruction	Function	1	2	M	3
FNC 118	DEBCDP (S) (D)	Convert BIN to DEC Floating Point Format	○	○	○	○
FNC 119	DEBINP (S) (D)	Convert DEC to BIN Floating Point Format	○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●			
D													●		●			

```
X0            (S) (D)
─┤ ├────── DEBCD  D0  D10
```

S : the source device to be converted

D : the destination device which the results to be stored

- When X0 = "ON", this instruction is activated. It will uses the BIN format floating number in (D1, D0) to convert the number to a DEC format floating number then moves the value into (D11, D10).
- This instruction is a 32-bit instruction. Therefore, be sure to input DEBCD or DEBCDP in the program.
- For example, if the content value in (D1, D0) is 1.234×10^{2}, then after the converted, (D10) = 1234 and (D11) = −1.

```
X1            (S) (D)
─┤ ├────── DEBIN  D2  D12
```

S : the source device to be converted

D : the destination device which the results to be stored

- When X1 = "ON", this instruction is activated. It will uses the DEC format floating number in (D3, D2) to convert the number to a BIN format floating number then moves the value into (D13, D12).
- This instruction is a 32-bit instruction. Therefore, be sure to input DEBIN or DEBINP in the program.
- For example, if the content value in (D2) = 2345 and (D3) = 5, then after the converted, (D13, D12) is 2.345×10^{8}.

- The format for storing a floating point number will occupy two Registers in the PLC system.
- Please refer to the section 2-13 "Numerical System" for the formats of BIN and DEC floating point numbers are stored in the Registers.

FNC 120	DEADDP S1 S2 D	BIN Floating Point Addition	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●		●	●	○	
S2													●		●	●	○	
D													●		●			

X0 — DEADDP D0 D2 D10 (S1 S2 D)

S1 : the summand
S2 : the addend
D : the total

- When X0 = "OFF"→"ON", the summand (D1, D0) will be added to the addend (D3, D2), and the total will be stored at the specified destination device (D11, D10). The components used in the instruction which are all stored in the BIN floating number format.

1.235×10^2 (D1, D0) BIN floating number
$+\ 3.2 \times 10^0$ (D3, D2) BIN floating number
1.267×10^2 (D11, D10) BIN floating number

- This instruction is a 32-bit instruction. Therefore, be sure to input DEADD or DEADDP in the program.

- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the addition function.

- The result of this instruction will affect the status of the following flags:
  When the operated result is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".
  When the operated result is exceeded the upper limit of the BIN floating point format (positive & negative included), the Carry Flag M9022 turns "ON".
  When the operated result is less than the insignificant limit of the BIN floating point format (positive & negative included), the Borrow Flag M9021 turns "ON".
  The available range of the BIN floating point format, please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 121	DESUBP S1 S2 D	BIN Floating Point Subtraction	1	2	M	3
			○	○	○	○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
	Devices																	
S1													●		●	●	○	
S2													●		●	●	○	
D													●		●			

X0 — DESUBP D0 D2 D10 (S1 S2 D)

S1 : the minuend
S2 : the subtrahend
D : the difference

- When X0 = "OFF"→"ON", the subtrahend (D3, D2) will be subtracted from the minuend (D1, D0), and the difference will be stored at the destination device (D11, D10). The components used in the instruction which are all stored in the BIN floating number format.

	1.235×10^2	(D1, D0) BIN floating number
−	3.2×10^0	(D3, D2) BIN floating number
	1.203×10^2	(D11, D10) BIN floating number

- This instruction is a 32-bit instruction. Therefore, be sure to input DESUB or DESUBP in the program.

- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the subtraction function.

- The result of this instruction will affect the status of the following flags:
  When the operated result is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".
  When the operated result is exceeded the upper limit of the BIN floating point format (positive & negative included), the Carry Flag M9022 turns "ON".
  When the operated result is less than the insignificant limit of the BIN floating point format (positive & negative included), the Borrow Flag M9021 turns "ON".
  The available range of the BIN floating point format, please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 122	DEMULP (S1) (S2) (D)	BIN Floating Point Multiplication	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1													●		●	●	○	
S_2													●		●	●	○	
D													●		●			

X0 —| |— DEMUL D0 D2 D10 ((S1) (S2) (D))

S_1 : the multiplicand

S_2 : the multiplier

D : the product (of a multiplication)

- When X0 = "ON", the multiplicand (D1, D0) will be multiplied by the multiplier (D3, D2), and the product will be stored at the destination device (D11, D10). The components used in the instruction which are all stored in the BIN floating number format.

$$\begin{array}{rl} 3.14\times10^{0} & \text{(D1, D0) BIN floating number} \\ \times\ 2.3\times10^{1} & \text{(D3, D2) BIN floating number} \\ \hline 7.222\times10^{1} & \text{(D11, D10) BIN floating number} \end{array}$$

- This instruction is a 32-bit instruction. Therefore, be sure to input DEMUL or DEMULP in the program.
- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the multiplication function.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 123	DEDIVP (S1) (S2) (D)	BIN Floating Point Division	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●		●	●	○	
S2													●		●	●	○	
D													●		●			

X0 — DEDIVP D0 D2 D10 ((S1) (S2) (D))

S_1 : the dividend
S_2 : the divisor
D : the quotient

- When X0 = "OFF"→"ON", the dividend (D1, D0) will be divided by the divisor (D3, D2), and the quotient will be stored at the destination device (D11, D10). The components used in the instruction which are all stored in the BIN floating number format.

	1.23×10^{4}	(D1, D0) BIN floating number
÷	3.0×10^{-1}	(D3, D2) BIN floating number
	4.1×10^{4}	(D11, D10) BIN floating number

- This instruction is a 32-bit instruction. Therefore, be sure to input DEDIV or DEDIVP in the program.
- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the division function.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- If the value of divisor (S2) is equal to "0", the PLC will regard that as an operational error.

FNC 124	DEXPP (S) (D)	BIN Floating Point Number Exponent	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

```
         X0              (S)  (D)
   ─┤ ├────────┤ DEXPP  D0  D10 │
```

S : the source device of the exponent at the function

D : the destination device to store the exponent result

- This instruction performs the exponential function on a BIN floating point number that is designated by (S) and stores the BIN floating point format result in the (D) designated device.
- When X0 = "OFF"→"ON", by way of the BIN floating point number in (D1, D0) as the exponent to perform the exponential function, and the result will be stored in (D11, D10).

$$e^{\boxed{2.1}} = \boxed{8.166167}$$

$\boxed{2.1}$ (D1, D0) BIN floating number

$\boxed{8.166167}$ (D10, D11) BIN floating number

- This instruction based on the constant base ***e*** = 2.71828 to perform the exponential function.
- This instruction is a 32-bit instruction. Therefore, be sure to input DEXP or DEXPP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- If the absolute value of the result (D) is larger than 2^{128} or less than 2^{-126}, the PLC will regard that as an operational error.

FNC 125	DLOGEP (S) (D)	BIN Floating Point Nature Logarithm	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			
• S>0																		

```
                  (S)  (D)
     X0
 |--| |----[DLOGEP  D0  D10]
```

S : the source device to be performed the natural logarithmic operation

D : the destination device to store the natural logarithmic result

- This instruction performs the natural logarithmic operation on a BIN floating point number that is designated by (S) and stores the BIN floating point format result in the (D) designated device.
- When X0 = "OFF"→"ON", this instruction performs the natural logarithmic operation on the BIN floating point number in (D1, D0) as the logarithm to the base of the mathematical constant ***e***, and the result will be stored in (D11, D10).

$\log_e \boxed{10} = \ln 10 = \boxed{2.302585}$ (D11, D10) BIN floating number

10: (D1, D0) BIN floating number

- This instruction is based on the constant base ***e*** = 2.71828 to perform the natural logarithm function.
- This instruction is a 32-bit instruction. Therefore, be sure to input DLOGEP or DLOGEP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- If the value of (S) is less than or equal to zero, the PLC will regard that as an operational error. Therefore, the (S) must designate to a positive number.

FNC 126	DLOG10P (S) (D)	BIN Floating Point Common Logarithm	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			
• S>0																		

S : the source device to be performed the common logarithmic operation

D : the destination device to store the common logarithmic result

- This instruction performs the common logarithmic operation on a BIN floating point number that is designated by (S) and stores the BIN floating point format result in the (D) designated device.
- When X0 = "OFF"→"ON", this instruction performs the common logarithmic operation on the BIN floating point number in (D1, D0) as the logarithm to the base 10, and the result will be stored in (D11, D10).

$\log_{10}$ [20] = [1.30103] (D11, D10) BIN floating number

└── (D1, D0) BIN floating number

- This instruction is based on the constant number 10 to perform the common logarithm function.
- This instruction is a 32-bit instruction. Therefore, be sure to input DLOG10 or DLOG10P in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- If the value of (S) is less than or equal to zero, the PLC will regard that as an operational error. Therefore, the (S) must designate to a positive number.

FNC 127	DESQRP (S) (D)	BIN Floating Point Square Root	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●	●	○	
D													●		●			

X0 —| |— DESQR D0 D10 (S) (D)

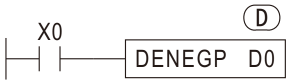
S : the source device to be performed the square root operation

D : the destination device to store the square root result

- This instruction performs the square root operation on a BIN floating point number that is designated by (S) and stores the BIN floating point format result in the (D) designated device.
- When X0 = "ON", this instruction performs the square root operation on the BIN floating point number in (D1, D0), and the result will be stored in (D11, D10).
- This instruction is a 32-bit instruction. Therefore, be sure to input DESQR or DESQRP in the program.
- If a source operand is assigned to a constant integer K or H, this instruction will automatically convert the number to BIN floating point number for the calculation. Therefore, could execute the square root function.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- When the operated result is equal to "0" , the Addition / Subtraction Zero Flag M9020 turns "ON".
- If the value of (S) is less than zero, the PLC will regard that as an operational error and turn the M9067 = "ON".

FNC 128	DENEGP (D)	BIN Floating Point Negation	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D													●		●			



D : the device to be positive/negative inverted

- This instruction inverts a BIN floating point format number which is designed by (D) into its negative value and stores the result back into (D) .
- When X0 = "OFF"→"ON", this instruction performs the positive/negative inversion operation on the BIN floating point number in (D1, D0), and then the result will be stored back into (D1, D0).

Before operation | −123.45 | (D1, D0) BIN floating number

↓ X0 = "OFF" → "ON"

After operation | 123.45 | (D1, D0) BIN floating number

- This instruction is a 32-bit instruction. Therefore, be sure to input DENEG or DENEGP in the program.
- The operand (D) must assign to a BIN floating point number for the calculation.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 129	DINTP (S) (D)	BIN Floating Point→BIN Integer Format	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●			
D													●		●			

- The 16-bit instruction, D occupies 1 component
- The 32-bit instruction, D occupies 2 components

```
 X0            (S)  (D)
-| |-------[ INT  D0  D10 ]
```

S : the source device to be performed the converting operation

D : the destination device to store the converted result

- This instruction performs the converting operation on a BIN floating point number that is designated by (S) and stores the converted BIN format integer result in the (D) designated device. The fractional part of the floating point number beyond the integer part will be ignored.

- When X0 = "ON", this instruction performs the converting operation on the BIN floating point number in (D1, D0), then the result of converted BIN format integer part will be stored in the D10 but the rest value (the fractional part) will be ignored.

Before operation [123.45] (D1, D0) BIN floating number

↓ X0 = "ON"

After operation [123] (D10) BIN integer, the value of fractional part will be ignored

- When this instruction is used as a 32-bit instruction, the (D) is a 32-bit device, too.

```
 X0              (S)  (D)
-| |-------[ DINTP  D0  D10 ]
```

- When X0 = "OFF"→"ON", this instruction performs the converting operation on the BIN floating point number in (D1, D0), then the result of converted BIN format integer part will be stored in the (D11, D10) but the rest value (the fractional part) will be ignored.

Before operation [123456.7] (D1, D0) BIN floating number

↓ X0 = "OFF" → "ON"

After operation [123456] (D11, D10) BIN integer, the value of fractional part will be ignored

- When the operated result is equal to "0", the Addition / Subtraction Zero Flag M9020 turns "ON".
  When the operated result has the fractional part and that is ignored, the Borrow Flag M9021 turns "ON".
  When the operated result is exceeded the available range of the BIN integer below, the Carry Flag M9022 turns "ON".
  For a 16 bit instruction: –32,768~32,767
  For a 32 bit instruction: –2,147,483,648~2,147,483,647

- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 130	DSINP (S) (D)	Calculate Sine	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

```
 X0          (S) (D)
─┤ ├────── DSIN  D0  D10
```

S : the source device for input the angle in radians

D : the calculated result

- This instruction performs the mathematical sine operation on the (S) designated BIN floating point radians of angle, and the result is stored in the (D) designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point radians at (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, be sure to input DSIN or DSINP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation, and the calculated result (D) is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- The example below is to calculate the 45 degree angle become a radian value of floating point, then use the radians to get the result value of SIN.

```
 M9000
─┤ ├──┬── DEDIV K314159 K18000000 D0    (π/180)→(D1,D0)
      ├── DEMUL K45 D0 D2               The angel of 45 degree×( π / 180 )→(D3, D2) [Unit: radian]
      └── DSIN D2 D10                   The result value of SIN is stored in (D11, D10).
```

FNC 131	DCOSP S D	Calculate Cosine	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

```
              S   D
 X0
─┤ ├───── DCOS  D0  D10
```

S : the source device for input the angle in radians

D : the calculated result

- This instruction performs the mathematical cosine operation on the S designated BIN floating point radians of angle, and the result is stored in the D designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point radians at (D1, D0) to calculate the cosine value, and the result will be stored at the specified destination devices (D11, D10).
- Radian = Degree × π / 180
- For to convert the angle unit between the radians and degrees, please refer to the DEG and RAD instructions.
- This instruction is a 32-bit instruction. Therefore, be sure to input DCOS or DCOSP in the program.
- The source operand S must assign to a BIN floating point number for the calculation, and the calculated result D is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 132	DTANP (S) (D)	Calculate Tangent	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

X0 — DTAN D0 D10 (S) (D)

S : the source device for input the angle in radians

D : the calculated result

- This instruction performs the mathematical tangent operation on the (S) designated BIN floating point radians of angle, and the result is stored in the (D) designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point radians at (D1, D0) to calculate the tangent value, and the result will be stored at the specified destination devices (D11, D10).
- Radian = Degree $\times \pi$ / 180
- For to convert the angle unit between the radians and degrees, please refer to the DEG and RAD instructions.
- This instruction is a 32-bit instruction. Therefore, be sure to input DTAN or DTANP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation, and the calculated result (D) is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 133	DASINP (S) (D)	Calculate Arc Sine	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

• $-1.0 \leq S \leq 1.0$

X0 —| |— [DASIN (S) D0 (D) D10]

S : the source device for input the value of sine

D : the device to store the calculated result (unit: radian)

- This instruction performs the mathematical Arcsine (inverse sine or $\sin^{-1}$) operation by the (S) designated BIN floating point sine value, and the result of radians is stored in the (D) designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point sine value at (D1, D0) to calculate the Arcsine value, and the result of radians will be stored at the specified destination devices (D11, D10).

  SIN^{-1} [0.8660254] = [1.047198] (D11, D10) BIN floating number

  └── (D1, D0) BIN floating number

  The calculated result at the (D11, D10) is by the value of radians and the range is between $-\pi/2$ and $\pi/2$.

- Degree = Radian $\times 180 / \pi$
- This instruction is a 32-bit instruction. Therefore, be sure to input DASIN or DASINP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation, and the calculated result (D) is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- For to convert the angle unit between the radians and degrees, please refer to the DEG and RAD instructions.
- The PLC will identify it as an operating error if the value of (S) is exceeded the available range $-1.0 \sim 1.0$.

FNC 134	DACOSP S D	Calculate Arc Cosine	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			
• $-1.0 \leqq S \leqq 1.0$																		

```
X0                 (S)  (D)
─┤ ├──────[DACOS  D0   D10]
```

S : the source device for input the value of cosine

D : the device to store the calculated result (unit: radian)

- This instruction performs the mathematical Arccosine (inverse cosine or $\cos^{-1}$) operation by the S designated BIN floating point cosine value, and the result of radians is stored in the D designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point cosine value at (D1, D0) to calculate the Arccosine value, and the result of radians will be stored at the specified destination devices (D11, D10).

  $\cos^{-1}$ [0.5] = [1.047198] (D11, D10) BIN floating number

  └── (D1, D0) BIN floating number

  The calculated result at the (D11, D10) is by the value of radians and the range is between 0 and π .

- Degree = Radian ×180 / π
- For to convert the angle unit between the radians and degrees, please refer to the DEG and RAD instructions.
- This instruction is a 32-bit instruction. Therefore, be sure to input DACOS or DACOSP in the program.
- The source operand S must assign to a BIN floating point number for the calculation, and the calculated result D is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.
- The PLC will identify it as an operating error if the value of S is exceeded the available range −1.0 ~ 1.0.

FNC 135	DATANP (S) (D)	Calculate Arc Tangent	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

```
         X0           (S) (D)
    |----| |------[ DATAN  D0  D10 ]
```

S : the source device for input the value of tangent

D : the device to store the calculated result (unit: radian)

- This instruction performs the mathematical Arctan (inverse tangent or $\tan^{-1}$) operation by the (S) designated BIN floating point tangent value, and the result of radians is stored in the (D) designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point tangent value at (D1, D0) to calculate the Arctangent value, and the result of radians will be stored at the specified destination devices (D11, D10).

  TAN^{-1} [1.732051] = [1.047198] (D11, D10) BIN floating number

  └── (D1, D0) BIN floating number

  The calculated result at the (D11, D10) is by the value of radians and the range is between $-\pi/2$ and $\pi/2$.

- Degree = Radian $\times 180 / \pi$
- For to convert the angle unit between the radians and degrees, please refer to the DEG and RAD instructions.
- This instruction is a 32-bit instruction. Therefore, be sure to input DATAN or DATANP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation, and the calculated result (D) is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 136	DRADP (S) (D)	Convert Angle From Degrees to Radian	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

```
         X0        (S)  (D)
 |--| |-------[DRAD  D0  D10]
```

S : the source device for input the angle in degrees

D : the calculated result in radians

- This instruction performs the mathematical converting operation on the (S) designated BIN floating point degrees of angle, and the result of radians is stored in the (D) designated device.
- When X0 = "ON", the instruction is activated, it uses the BIN floating point degrees at (D1, D0) to convert, and the result of radians will be stored at the specified destination devices (D11, D10).

Angle by degrees		Angle by radians
60	X0 = ON →	1.047198
(D1, D0) BIN floating number		(D11, D10) BIN floating number

- Radian = Degree × π / 180
- This instruction is a 32-bit instruction. Therefore, be sure to input DRAD or DRADP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation, and the calculated result (D) is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 "Numerical System" for the reference of the floating point number.

FNC 137	DDEGP (S) (D)	Convert Angle From Radian to Degrees	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●		●		○	
D													●		●			

X0 — DDEG D0 D10 (S) (D)

S : the source device for input the angle in degrees

D : the calculated result in radians

- This instruction performs the mathematical converting operation on the (S) designated BIN floating point radians of angle, and the result of degrees is stored in the (D) designated device.
- When X0 = “ON”, the instruction is activated, it uses the BIN floating point radians at (D1, D0) to convert, and the result of degrees will be stored at the specified destination devices (D11, D10).

Angle by radians		Angle by degrees
0.5235988	X0 = ON →	30
(D1, D0) BIN floating number		(D11, D10) BIN floating number

- Degree = Radian $\times 180 / \pi$
- This instruction is a 32-bit instruction. Therefore, be sure to input DDEG or DDEGP in the program.
- The source operand (S) must assign to a BIN floating point number for the calculation, and the calculated result (D) is by the BIN floating point format, too.
- The format for storing a floating point number will occupy two Registers in the PLC system. Please refer to the section 2-13 “Numerical System” for the reference of the floating point number.

6-13 Advanced Data Processing and MBUS Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
140	D WSUM P (S) (D) (n)	Sum of Word Data				○
141	WTOB P (S) (D) (n)	Split Word to Byte				○
142	BTOW P (S) (D) (n)	Combine Byte to Word				○
143	UNI P (S) (D) (n)	Combine 4-bit Nibble to Word				○
144	DIS P (S) (D) (n)	Separate Word to 4-bit Nibble				○
147	D SWAP P (D)	Swap High / Low Byte	○	○	○	○
148	D SORT2 (S) (m1) (m2) (D) (n)	Sort Tabulated Data 2	○	○	○	○
149	MBUS (S1) (S2) (n)	MODBUS Communication Instruction	○	○	○	○

FNC 140	DWSUMP (S) (D) (n)	Sum of Word Data	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●		●			
D											●	●	●		●			
n													○			○		

- For a 16-bit instruction, S occupies n components, D occupies 2 components
- For a 32-bit instruction, S occupies (2×n) components, D occupies 4 components • n>0

```
        X0              (S)   (D)   (n)
 ─┤ ├───────[ WSUM   D10   D100   K5 ]
```

S : the head ID of source devices to be added up

D : the destination device to store the amount

n : the number of consecutive devices to be accumulated

- This 16-bit instruction adds up the consecutive (n) 16-bit word devices which are started from (S), then store the result to the 32-bit double word device (D) .
- When the X0 = "ON", this instruction accumulates the contents of D10~D14 (5 16-bit devices), then stores the amount to the (D101, D100).

(n) = 5

(S)	
D10	101
D11	55
D12	234
D13	186
D14	79

X0 = ON → 655 (D) (D101, D100)

- The result device (D) for this 16-bit instruction is a 32-bit device, the positive or negative sign is located at its the Most Significant Bit (MSB = 0 = positive; MSB = 1 = negative).

```
        X0              (S)   (D)   (n)
 ─┤ ├───────[ DWSUM   D0   D100   K4 ]
```

- This 32-bit instruction adds up the consecutive (n) 32-bit word devices which are started from (S), then store the result to the 64-bit double word device (D) .
- When the X0 = "ON", this instruction accumulates the contents of (D1, D0) ~ (D7, D6) (4 32-bit devices), then stores the amount to the (D103, D102, D101, D100).

(n) = 4

(S)	
(D1, D0)	100000
(D3, D2)	200000
(D5, D4)	300000
(D7, D6)	400000

X0 = ON → 1000000 (D) $\underbrace{\text{(D103, D102, D101, D100)}}_{\text{64 bits}}$

- The result device (D) for this 32-bit instruction is a 64-bit device, the positive or negative sign is located at its the Most Significant Bit (MSB = 0 = positive; MSB = 1 = negative).

FNC 141	WTOBP S D n	Split Word to Byte	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D											●	●	●					
n													○			○		

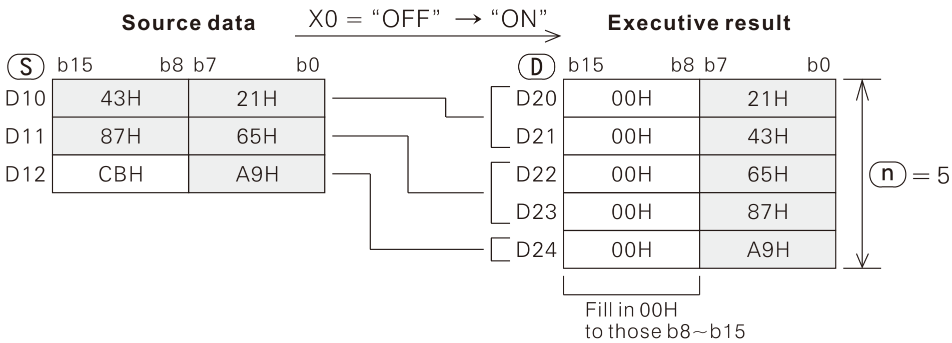
- S occupies (n/2) components • D occupies n components • n ≧ 0

S : the head ID of source devices to be split

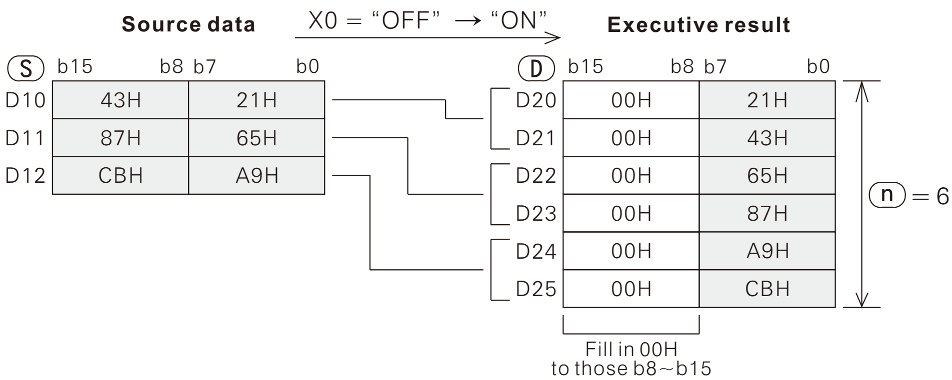
D : the head ID of storage devices for the split result

n : the number of bytes to be split

- This instruction splits the consecutive n byte devices which are started from the lower byte of S, then store the split result to the lower bytes of devices those are started from D.
- When X0 = "OFF" → "ON", the instruction separates the contents of D10~D11 and D12's lower byte into 5 bytes of data, then stores those bytes to 5 16-bit word components D20~D24. Since it is no data for all the upper bytes of D20~D23, these bytes will be filled in 00H.



- If n = 6, the execute result is shown below.



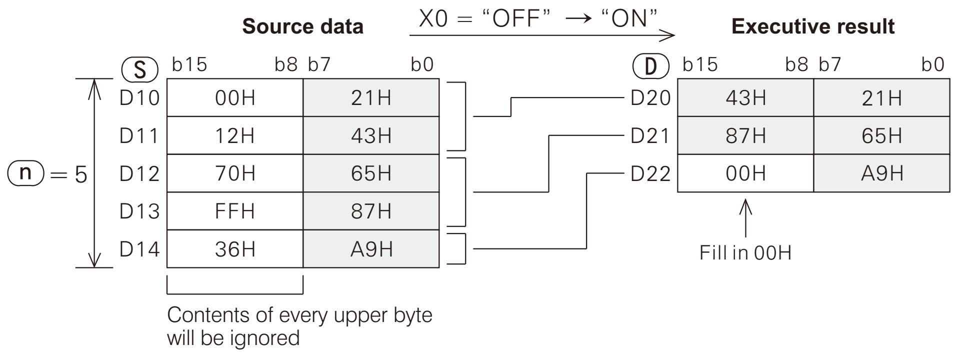
- If n = 0, the instruction will not execute.

FNC 142	BTOWP (S) (D) (n)	Combine Byte to Word	1	2	M	3
						○

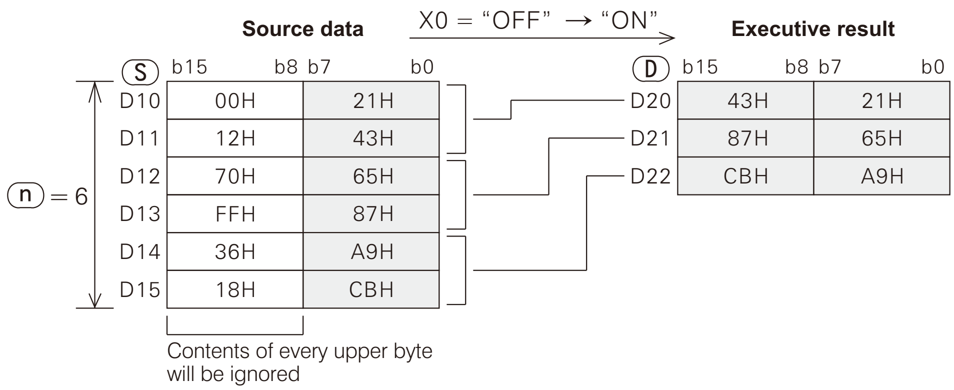
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D											●	●	●					
n													○			○		
• S occupies n components					• D occupies (n/2) components						• n ≧ 0							

S : the head ID of source devices to be combined
D : the head ID of storage devices for the combined result
n : the number of bytes to be combined

- This instruction takes all the lower bytes of the devices that started from the (S) to get (n) bytes of data, then restructures and stores the data to continuous 16-bit words that started from (D) .
- When X0 = "OFF" → "ON", the instruction takes 5 bytes of data from those lower bytes within D10~D14, to merge and store the data into 16-bit word components D20~D22. Since it is no data for the upper byte of D22, this byte will be filled in 00H.



- If (n) = 6, the execute result is shown below.



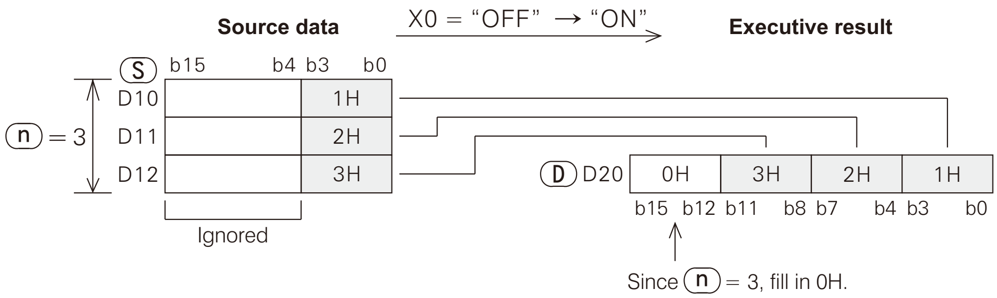
- If (n) = 0, the instruction will not execute.

FNC 143	UNIP S D n	Combine 4-bit Nibble to Word	1	2	M	3
						○

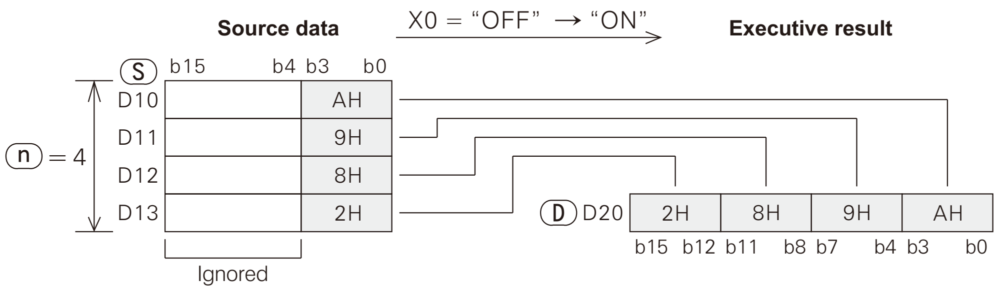
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D											●	●	●					
n													○			○		
• S occupies n components	• n = 0~4																	

S : the head ID of source devices to be combined
D : the head ID of storage devices for the combined result
n : the number of 4-bit nibbles to be combined

- This instruction takes all the lowest nibbles of the devices that started from the S to get n nibbles of data, then restructures and stores the data to the 16-bit word device D . If n = 0, the instruction will not execute. When the n < 4, the vacant nibbles will be filled in 0H.
- When X0 = "OFF" → "ON", the instruction takes 3 nibbles of data from those lowest nibbles within D10~D12, to merge and store the data into a 16-bit word component D20. Since it is no data for the highest nibble of D20, this nibble will be filled in 0H.



- If n = 4, the execute result is shown below.



FNC 144	DISP P (S) (D) (n)	Separate Word to 4-bit Nibble	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D											●	●	●					
n													○			○		
• D occupies n components • n = 0~4																		

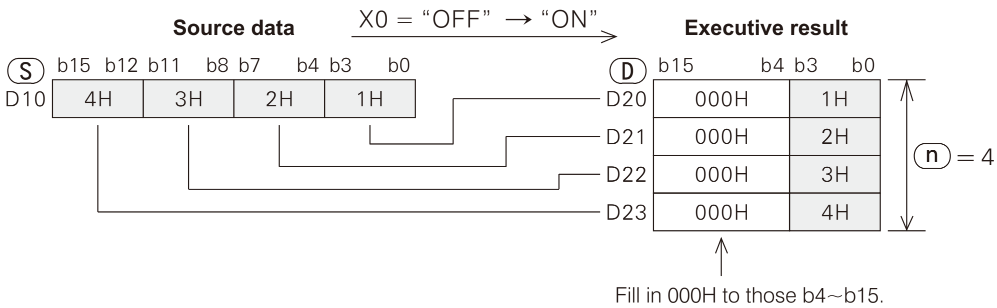
S : the source device to be split

D : the head ID of storage devices for the split result

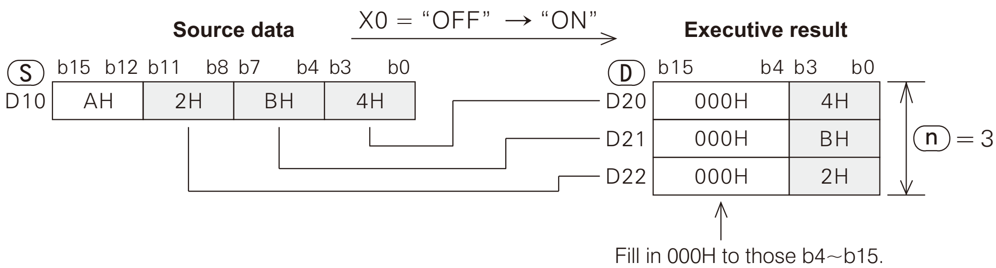
n : the number of 4-bit nibbles to be split

- This instruction separates the source device (S) to get (n) nibbles of data, then stores the nibble form data to (n) 16-bit word devices that started from the (D). For all the devices related to (D), those vacant upper 3 nibbles (b4~b15) will be filled in 0. If (n) = 0, the instruction will not execute.

- When X0 = "OFF" → "ON", the instruction separates the content of D10 into 4 nibbles, then stores those nibbles to 4 16-bit word components D20~D23. Since it is no data for all the b4~b15 of D20~D23, these 3 nibbles at every result register will be filled in 000H.



- If (n) = 3, the execute result is shown below.



FNC 147	DSWAPP (D)	Swap High / Low Byte	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S								●	●	●	●	●	●	○	●			

X20 — SWAPP D0 (D)

D : the Upper / Lower 8 bits of the device to be swapped

- When X20 = "OFF"→"ON", the Upper 8 bits and Lower 8 bits in the (D0) will be swapped.

Upper 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 = "OFF"→"ON", the Upper / Lower 8 bits in the (D10) will be swapped also the Upper / Lower 8 bits in the (D11) will be swapped, too.

Upper 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 148	D SORT2 S m1 m2 D n	Sort Tabulated Data 2	1 2 M 3

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													○					
m1													○			○		
m2																○		
D													○					
n													○			○		

- For a 16-bit instruction, each of the S and D respectively occupy ($m_1 \times m_2$) components
- For a 32-bit instruction, each of the S and D respectively occupy ($2 \times m_1 \times m_2$) components

- $m_1 = 1 \sim 32$
- $m_2 = 1 \sim 6$
- $n = 1 \sim m_2$

```
      X0
  ─┤ ├────[SORT2  D0  K5  K4  D100  D200]
               S   m1  m2   D     n
```

S : the head register ID of the original data array

m_1 : the number of data record sets to be sorted

m_2 : the number of data at an arrangement for each set

D : the head register ID number of the data array where to store the sorted results

n : to assign the nth. data in an arrangement as the sort basis n^{th}

Original Data Array
(Starting from the appointed register S)

Data Arrangement			
1	2	3	4
Student ID	Philology	Mathematics	History
(D0) 1	(D1) 80	(D2) 70	(D3) 75
(D4) 2	(D5) 65	(D6) 70	(D7) 90
(D8) 3	(D9) 90	(D10) 65	(D11) 80
(D12) 4	(D13) 75	(D14) 90	(D15) 65
(D16) 5	(D17) 80	(D18) 85	(D19) 95

m1 (rows) × m2 (columns)

- The SORT2 instruction is used to sort an data array that has a serial of data record sets (the number of sets is assigned by m1). Each set has a grouped arrangement (the number of grouped data at each arrangement set is assigned by m2). While that n is used to assign the n^{th} arrangement as the routing basis to sort the data array. The S is to designate the head register ID of the original data array to be sorted, and the D is to designate the head register ID of the result data array.
- Both of the arrays S and D will separately occupy m1 × m2 consecutive registers.
- The M9165 is to assign the instruction's operation mode. If the M9165 = "OFF", the data sorting is in ascending order; conversely, "ON" is in descending order.
- When X0 = "ON", the SORT2 instruction is starting to perform. This instruction needs take 4 (the content of m1) scan cycles to complete the sort action. When it is completed, the Execution Completed Flag M9029 = "ON" for a Scan Time and the sort action will be stopped.

- When M9165 = "OFF", the data sorting is in ascending order.

Sorted Data Result Array if the D200 = 2
(Starting from the appointed register D)

Data Arrangement			
1	2	3	4
Student ID	Philology	Mathematics	History
(D100) 2	(D101) 65	(D102) 70	(D103) 90
(D104) 4	(D105) 75	(D106) 90	(D107) 65
(D108) 1	(D109) 80	(D110) 70	(D111) 75
(D112) 5	(D113) 80	(D114) 85	(D115) 95
(D116) 3	(D117) 90	(D118) 65	(D119) 80

m1 (rows) × m2 (columns)

Sorted Data Result Array if the D200 = 4
(Starting from the appointed register D)

Data Arrangement			
1	2	3	4
Student ID	Philology	Mathematics	History
(D100) 4	(D101) 75	(D102) 90	(D103) 65
(D104) 1	(D105) 80	(D106) 70	(D107) 75
(D108) 3	(D109) 90	(D110) 65	(D111) 80
(D112) 2	(D113) 65	(D114) 70	(D115) 90
(D116) 5	(D117) 80	(D118) 85	(D119) 95

m1 (rows) × m2 (columns)

• When M9165 = "ON", the data sorting is in descending order.

Sorted Data Result Array if the D200 = 3
(Starting from the appointed register D)

Data Arrangement			
1	2	3	4
Student ID	Philology	Mathematics	History
(D100) 4	(D101) 75	(D102) 90	(D103) 65
(D104) 5	(D105) 80	(D106) 85	(D107) 95
(D108) 1	(D109) 80	(D110) 70	(D111) 75
(D112) 2	(D113) 65	(D114) 70	(D115) 90
(D116) 3	(D117) 90	(D118) 65	(D119) 80

m1 (rows), m2 (columns)

Sorted Data Result Array if the D200 = 1
(Starting from the appointed register D)

Data Arrangement			
1	2	3	4
Student ID	Philology	Mathematics	History
(D100) 5	(D101) 80	(D102) 85	(D103) 95
(D104) 4	(D105) 75	(D106) 90	(D107) 65
(D108) 3	(D109) 90	(D110) 65	(D111) 80
(D112) 2	(D113) 65	(D114) 70	(D115) 90
(D116) 1	(D117) 80	(D118) 70	(D119) 75

m1 (rows), m2 (columns)

• In case of the 32-bit instruction is used, all of the components at S , m1 , m2 , D and n must by the format of 32-bit.

• The related special devices are summarized below:

Relay ID No.	Description
M9029	Instruction execution completed flag. M9029="ON" when the executions of some applied instructions are completed (please refer to the relevant instructions).
M9165	Assigned the SORT2 instruction sort process mode. When M9165="OFF" by ascending order; and M9165="ON" by descending order.

• Notes

(1) During the instruction is in operation, all parameters and the contents of data are not allowed to change.

(2) There is no limit on the used times of the SORT2 instruction in the program. However, only two instructions can be executed at the same time.

(3) After the operation of this instruction is completed and if that is required to sort again, should switch the condition contact "OFF" and then "ON".

(4) In this instruction, the components of the data arrays for the S and D are not allowed overlap, except for S = D .

FNC 149	MBUS (S1) (S2) (n)	MODBUS Communication Instruction	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
n																○		
• S2 occupies 4 components • For the VS1, VS2 or VSM series, n = 1~3; for the VS3 series, n = 1~5																		

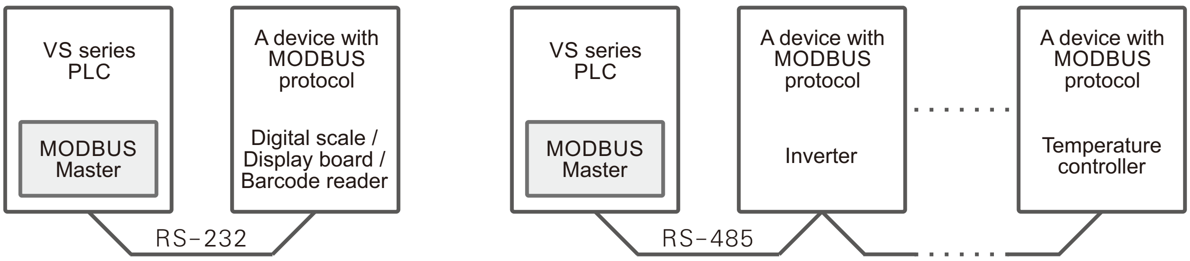
```
            (S1)              (S2)  (n)
X20
─┤ ├──[MBUS ACCESS_MBS_EQUIP D100 K1]
```

S1 : the communication table that describes the data receiving and transmitting

S2 : the working area for the instruction that occupies 4 registers

n : to assign the communication port, 1~5 = CP1~CP5

- The VS series PLC uses this instruction to transmit or get the designated data via its Communication Port CP1~CP5 with other peripherals which use the MODBUS ASCII or RTU protocol.
- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the "Application type:" become the "MODBUS Master". Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the "Project" -- "COM Port Setting".
- As shown in the figure below, connect the PLC with other device(s) by the RS-232 or RS-485 wiring. Furthermore, to set a unique station number (1~247) for every device and make the other parameters identical. Besides, use the Ladder Master S to open the project of the VS series PLC and write this MBUS instruction in the Master PLC's program, then set the port's "Application type:" as the "MODBUS Master" also to set the communication parameters in that window. After that, edit the MODBUS communication table to designate data transferring actions. Moreover, to load this project to the Master PLC. Following these procedures, the purpose of data delivery between the PLC and peripherals can be reached.



- When X20 = "ON", the MBUS instruction starts execution. According to the contents described in the "ACCESS_MBS_EQUIP" communication table, it proceeds data write or read actions to the appointed peripherals, while the D100~D103 will be occupied to store the state of instruction execution.
- When the contents of the communication table specified by the (S1) are execution completed from beginning to end, the M9104 will be "ON" for a Scan Time then the procedure will start over again from the first item set of the table.
- When the X20 turns from "ON" to "OFF", this instruction stops and the data sharing immediately stop but data which has transferred previously will still remain.
- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
■M9104	CP1 LINK / MBUS instruction on execution table complete once flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
■M9114	CP2 LINK / MBUS instruction on execution table complete once flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
■M9124	CP3 LINK / MBUS instruction on execution table complete once flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
■M9134	CP4 LINK / MBUS instruction on execution table complete once flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.
■M9144	CP5 LINK / MBUS instruction on execution table complete once flag.

• The communication table is assigned by the S1 of the instruction.

An example of the "MBUS" provided by programming tool the Ladder Master S is shown below.

Item No.	Command	Device at Master	Direction	Slave St. No.	Device at Slave	Length	Word/Bit	Disable Contact
1	H03 Read (4x)	D0	<--	1	1000	10	—	M1
2	H10 Write (4x)	D10	-->	1	1100	5	—	M1
3	H03 Read (4x)	D20	<--	2	1000	10	—	M2
4	H10 Write (4x)	D30	-->	2	1100	5	—	—
5	H10 Write (4x)	D40	-->	0	100	20	—	M0

The first item set in the table means that the Master reads the data 41000~41009 from the Slave #1 and stores them at the D0~D9 of the Master.

The second item set in the table means that the Master writes its data at D10~D14 to the 41100~41104 of the Slave #1.

The third item set in the table means that the Master reads the data 41000~41009 from the Slave #2 and stores them at the D20~D29 of the Master.

The fourth item set in the table means that the Master writes its data at D30~D34 to the 41100~41104 of the Slave #2.

The fifth item set in the table means that the Master writes its data at D40~D59 to the 40100~40119 of all the Slaves. To set the Slave station #0 means that the Master writes the data to all of the Slaves. Note that Read Command cannot be used in this application.

The last column of the table is for the Disable Contact. If that specified contact is "ON", the communication item set will be ignored. For example, if M1 = "ON", the first and second item sets in the table will skip. This is the new function of the VS series, which can help designers effectively manage the operation of the communication table. A communication item set does not need to specify a Disabled Contact (such as the fourth item set), so that the command of the set does not have the disable control function.

• The working area of the executive instruction is starting from the S2 (using D100~D103 as the example).

S2		**Description**
D100	Lower 8 bits	The record of the Slave's station number when the first communication error is occurred
	Upper 8 bits	Instruction working status 0: Normal data transmitting / receiving 2: The length of the received data is incorrect 4: The error is caused by the assigned device is inappropriate 7: The error is caused by the communication command is incorrect A: The communication setting is normal but no response from the Slave station (Time-out occurs) B: Abnormal communication
D101 ~ D103	The working area is required for the system when this instruction is performed	

The MBUS instruction will reset the D100 to be "0" before the first item set at the table is executed.

If any communication error occurs during the execution of the instruction, a code will be recorded in D100. Only when the content value of D100 is "0", the recording action can be executed. Therefor, when there are possibilities of several errors, users can use the program to move out the content value of D100 then reset it to "0". This way allows the D100 to record the next error.

• To edit a communication table

Use the Ladder Master S to set up a MBUS communication table and through its interactive window can set up and edit a communication table easily.

In the structure of VS series PLC, the communication tables are a part of the project. When the programmer to copy or access the project, those tables will be duplicated automatically with the program.

MEMO

6-14 Real Time Clock Related Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
160	TCMP P S1 S2 S3 S D	Time Compare	○	○	○	○
161	TZCP P S1 S2 S D	Time Zone Compare	○	○	○	○
162	TADD P S1 S2 D	Time Addition	○	○	○	○
163	TSUB P S1 S2 D	Time Subtraction	○	○	○	○
164	D HTOS P S D	Convert Hour to Second	○	○	○	○
165	D STOH P S D	Convert Second to Hour	○	○	○	○
166	TRD P D	Read RTC Data	○	○	○	○
167	TWR P S	Set RTC Data	○	○	○	○

FNC 160	TCMP P (S1) (S2) (S3) (S) (D)	Time Compare	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1							●	●	●	●	●	●	●	○	●	●		
S_2							●	●	●	●	●	●	●	○	●	●		
S_3							●	●	●	●	●	●	●	○	●	●		
S											●	●	●		●			
D		●	●	●	○	○												

• S_1 = 0~23 • S_2 = 0~59 • S_3 = 0~59 • S occupies 3 components • D occupies 3 components

X20 — TCMP K8 K30 K20 D0 M0 (S1 = K8, S2 = K30, S3 = K20, S = D0, D = M0)

$\underbrace{\text{K8 K30 K20}}_{08^h\,30^m\,20^s}$

S_1 : the "Hour" for comparison

S_2 : the "Minute" for comparison

S_3 : the "Second" for comparison

S : the registers of the time combination to do the comparison

D : the compare result; occupying 3 consecutive points

- Compare the setting value (Hours, Minutes and Seconds which are designated by S_1 ~ S_3) to the time combination (specified by the head ID S of 3 consecutive data devices), and the result of the comparison is stored to D .

- When X20 = "ON", the instruction will be performed.

If $08^h\ 30^m\ 20^s$ > [D0 (Hour) / D1 (Minute) / D2 (Second)], then M0 = "ON"

If $08^h\ 30^m\ 20^s$ = [D0 (Hour) / D1 (Minute) / D2 (Second)], then M1 = "ON"

If $08^h\ 30^m\ 20^s$ < [D0 (Hour) / D1 (Minute) / D2 (Second)], then M2 = "ON"

- The current time of the real time clock is stored in Special Registers D9013~D9015.
  D9015 (Hours), D9014 (Minutes), D9013 (Seconds)

- When X20 = "OFF", the instruction is disabled, the "ON"/"OFF" status of M0, M1 and M2 remains the same as the status before X20 = "OFF".

- Please use serial or parallel links of M0~M2 to generate the result as "≥", "≤" or "≠".

- If the content value of the register designated by S exceeding the time value required, the PLC will regard that as an operational error.

FNC 161	TZCPP (S1) (S2) (S) (D)	Time Zone Compare	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●		●			
S2											●	●	●		●			
S											●	●	●		●			
D		●	●	●	○	○												

- All the S1, S2, S and D occupies 3 components respectively
- $S_1 \leqq S_2$

```
X20          (S1) (S2) (S)  (D)
-| |------[ TZCP  D0   D10  D20  M0 ]
```

S_1 : the lower setting of the time period combination

S_2 : the upper setting of the time period combination

S : the registers of the time combination to do the comparison

D : the compare result; occupying 3 consecutive points

- The time value is defined by (S), it will be compared to the lower setting of the time period defined by (S1) and the upper setting of the 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)			, then M0 = "ON"
D2 (Second)		D22 (Second)			

				(S2)	
D0 (Hour)		D20 (Hour)		D10 (Hour)	
D1 (Minute)	≦	D21 (Minute)	≦	D11 (Minute)	, then M1 = "ON"
D2 (Second)		D22 (Second)		D12 (Second)	

D20 (Hour)		D10 (Hour)	
D21 (Minute)	>	D11 (Minute)	, then M2 = "ON"
D22 (Second)		D12 (Second)	

- The current time of the real time clock is stored in Special Registers D9013~D9015. D9015 (Hours), D9014 (Minutes), D9013 (Seconds)
- When X20 = "OFF", the instruction is disabled, the "ON"/"OFF" status of M0, M1 and M2 remains the same as the status before X20 = "OFF".
- When (S1) > (S2), the content value of (S1) will become both upper/lower settings to be compared with the (S).
- If the content value of the register designated by (S1) , (S2) or (S) exceeding the time value required, the PLC will regard that as an operational error.

FNC 162	TADDP (S1) (S2) (D)	Time Addition	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●		●			
S2											●	●	●		●			
D											●	●	●		●			

• All the S1, S2 and D occupies 3 components respectively

```
                (S1) (S2) (D)
 X20
 |—| |——————[TADD  D0  D10  D20]
```

S1 : the summand of the time combination
S2 : the addend of the time combination
D : the sum of the time combination

- The designated time combination of the summand (S1) will be added to the designated time combination (S2) and the combination of the sum will be stored at the assigned registers (D).
- When X20 = "ON", the instruction will be 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)
$08^h\ 30^m\ 0^s$		$06^h\ 35^m\ 30^s$		$15^h\ 05^m\ 30^s$

- If the result of the time addition is exceeded 24 hours, the Carry Flag M9022 will be set "ON" and the (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)	, also the M9022 = "ON".
D2 = 30 (Second)		D12 = 20 (Second)		D22 = 50 (Second)	
$08^h\ 25^m\ 30^s$		$20^h\ 10^m\ 20^s$		$04^h\ 35^m\ 50^s$	
				28-24 = 4	

- If the result of the time addition equals to "0" (0 hour 0 min 0 sec), then the Zero Flag M9020 will be set "ON".
- If the content value of the registers that designated by (S1) or (S2) exceeding the time value required, the PLC will regard that as an operational error.

FNC 163	TSUBP (S1) (S2) (D)	Time Subtraction	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●		●			
S2											●	●	●		●			
D											●	●	●		●			

• All the S1, S2 and D occupies 3 components respectively

```
                (S1)  (S2)  (D)
 X20
 ─┤ ├──── TSUB  D0   D10   D20
```

S1 : the minuend of the time combination
S2 : the subtrahend of the time combination
D : the difference of the time combination

- The designated time combination of the minuend (S1) will be subtracted to the designated time combination (S2) and the combination of the difference will be stored at the assigned registers (D).
- When X20 = "ON", the instruction will be 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)
$18^h\ 28^m\ 50^s$		$08^h\ 40^m\ 20^s$		$09^h\ 48^m\ 30^s$

- If the result of the time subtract is a negative value, the Borrow Flag M9021 will be set "ON" and the (D) will display the value where 24 hours is added to the difference.

(S1)		(S2)		(D)	
D0 = 6 (Hour)		D10 = 20 (Hour)		D20 = 10 (Hour)	
D1 = 30 (Minute)	−	D11 = 20 (Minute)	=	D21 = 10 (Minute)	, also the M9021 = "ON".
D2 = 20 (Second)		D12 = 10 (Second)		D22 = 10 (Second)	
$06^h\ 30^m\ 20^s$		$20^h\ 20^m\ 10^s$		$10^h\ 10^m\ 10^s$	
				(−14)+24 = 10	

- If the result of the time subtraction equals to "0" (0 hour 0 min 0 sec), then the Zero Flag M9020 will be set "ON".
- If the content value of the registers that designated by (S1) or (S2) exceeding the time value required, the PLC will regard that as an operational error.

FNC 164	DHTOSP (S) (D)	Convert Hour to Second	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
• S occupies 3 components																		

```
 X0          (S)  (D)
─┤ ├───── HTOS  D0   D10
```

S : the head register of the time combination to be transferred

D : the transferred result

- The designated time combination (S) uses three registers to store the Hours, Minutes and Seconds to be transferred, this instruction converts those into decimal seconds and store the result to the assigned register (D) .

- When X0 = “ON”, the instruction will be performed. The time combination at D0~D2 (Hours, Minutes and Seconds) will convert to equal decimal seconds and the result stores at the D10.

(S)

D0 = 03 (Hour)
D1 = 15 (Minute)
D2 = 48 (Second)

X0＝ON → (D) D10 | 11748 (Seconds) |

- In case of the 32-bit instruction is used, the component (D) is by the format of 32-bit and that occupies 2 registers.

```
 X0           (S)  (D)
─┤ ├───── DHTOSP  D0   D10
```

(S)

D0 = 40 (Hour)
D1 = 21 (Minute)
D2 = 33 (Second)

X0 ＝ OFF→ON → (D) (D11, D10) | 145293 (Seconds) |

- For the 16-bit instruction, (S) = 0~4, (S)+1 = 0~59, (S)+2 = 0~59. If the content value of the registers exceeding the range, the PLC will be regarded as an operating error.

- For the 32-bit instruction, (S) = 0~32767, (S)+1 = 0~59, (S)+2 = 0~59. If the content value of the registers exceeding the range, the PLC will regard that as an operational error.

FNC 165	DSTOHP (S) (D)	Convert Second to Hour	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
• D occupies 3 components																		

X0 — STOH D10 D20 (S) (D)

S : the decimal seconds to be transferred

D : the transferred head register of the time combination result

- The decimal seconds at (S) transfers to the Hours, Minutes and Seconds then respectively store those into three registers (D), (D)+1 and (D)+2.
- When X0 = "ON", the instruction will be performed. The decimal seconds at D10 will convert to equal time combination by Hours, Minutes and Seconds and store those results into the D20, D21 and D22.

(S) D10: 11748 (Seconds) —X0 = ON→ (D)

D20 = 03 (Hour)
D21 = 15 (Minute)
D22 = 48 (Second)

- In case of the 32-bit instruction is used, the component (S) is by the format of 32-bit and that occupies 2 registers.

X0 — DSTOHP D10 D20 (S) (D)

(S) (D11, D10): 145293 (Seconds) —X0 = OFF→ON→ (D)

D20 = 40 (Hour)
D21 = 21 (Minute)
D22 = 33 (Second)

- For the 32-bit instruction, (S) = 0~117,964,799. If the content value of the register exceeding the range, the PLC will regard that as an operational error.
- For the 16-bit instruction, (S) = 0~32,767. If the content value of the register exceeding the range, the PLC will be regarded as an operating error.

FNC 166	TRDP (D)	Read RTC Data	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D											●	●	●		●			

• D occupies 7 components

```
       (D)
X20
─┤ ├───[TRD  D0]
```

D : the head register that stores the Present Value of the RTC

- When the Main Unit has installed a VS-MCR multi-function memory card, the PLC will provide with the real time clock (RTC) function. The real time clock has seven sets of data, such as year, month, date, hour, minute, second and day of the week, the data will be stored in the Special Registers D9013~D9019 sequentially.
- Programmers do not need to memorize the Special Registers of the RTC, they can use this instruction to read the current time and store the time data to contiguous 7 registers which are specified by (D).
- When X20 = "ON", the instruction will be performed to read the current time from the RTC.

Item	Special Register	Content Value		(D)
Year (C.E.)	D9018	2000～2099	Read ⇨	D0
Month	D9017	1～12		D1
Date	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 meaning of the content value at the D9019:
  0 = Sunday,
  1 = Monday,
  2 = Tuesday,
  3 = Wednesday,
  4 = Thursday,
  5 = Friday,
  6 = Saturday.

FNC 167	TWRP (S)	Set RTC Data	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●		●			

• S occupies 7 components

```
          (S)
X20
─┤ ├────[TWR  D0]
```

S : the head register that stores the set time data of the RTC

- When the Main Unit has installed a VS-MCR multi-function memory card, the PLC will provide with the real time clock (RTC) function. The real time clock has seven sets of data, such as year, month, date, hour, minute, second and day of the week, the data will be stored in the Special Registers D9013~D9019 sequentially.
- When X20 = "ON", the instruction will be performed to write the designated data at (S) ~ (S)+6 to the RTC.

(S)	Content Value		Special Register		RTC
D0	2000～2099		D9018		Year (C.E.)
D1	1～12		D9017		Month
D2	1～31	Write	D9016	Write	Date
D3	0～23		D9015		Hour
D4	0～59		D9014		Minute
D5	0～59		D9013		Second
D6	0～6		D9019		Week

- The meaning of the content value 0~6 at the D9019 represents Sunday~Saturday.
- If one of the content value in the source register block (S) is exceeding its available range above, the PLC will regard that as an operational error.
- Alternatively, can use the programming software Ladder Master S to set up the current time of the RTC.

MEMO

6-15 Code Conversion and Timer Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS 1	2	M	3
169	DHOUR S D1 D2	Hour Meter	○	○	○	○
170	DGRYP S D	Convert BIN to Gray Code	○	○	○	○
171	DGBINP S D	Convert Gray Code to BIN	○	○	○	○
176	TFT D1 S D2	Timer (10 ms.)	○	○	○	○
177	TFH D1 S D2	Timer (100 ms.)	○	○	○	○
178	TFK D1 S D2	Timer (1 sec.)	○	○	○	○

FNC 169	DHOUR (S) (D1) (D2)	Hour Meter	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D_1													●					
D_2		●	●	●	○	○												

• For a 16-bit instruction, D_1 occupies 2 components • For a 32-bit instruction, D_1 occupies 3 components

```
                 (S)     (D1)   (D2)
   X0
|--| |-----[HOUR  K1000  D7000  Y0]
```

S : the set value of the timer (Unit: hour)

D_1 : the Present Value of the timer (Unit: hour)

D_2 : the output contact of the timer

- This instruction creates a timer which is using the hour as the counting unit.
- When the condition contact is "ON", this timer increases its Present Value at (D1). After a while, the Present Value of the timer will equal to or larger than the set value (S), that causes the timer's output contact (D2) = "ON".
- The real setting period of the timer = 1 hour × the set value (S) .
- The (D1) stores the integer number of the Present Value (in hours). Besides, the register next to the (D1) is to store the remnant Present Value which is less than 1 hour (in seconds).
- As the example diagram above:
  When X0 = "ON", the Present Value of the register (D1) will begin to do the increase counting (hourly). If the Present Value of D7000 reaches the set value K1000 (1000 hours), the output contact of the timer Y0 = "ON". When X0 = "OFF", the Present Value of the timer at the registers will be retained. Therefrom, the characteristic of this instruction is similar to a retentive timer.
- Mostly, this instruction is used to monitor the lifespan of a component or to remind the regular maintenance.
  For to retain the Present Value of the timer during the power cut off, please assign the (D1) to a latched register.
  If assign the (D1) to an unlatched general register, when the power cut off or the PLC is "STOP", the content value of the (D1) will be reset to "0".
- After the timer's output contact (D2) = "ON", the timer's Present Value at the (D1) will continuously execute the increase counting.
- When the timer's Present Value (D1) reaches the maximum value of a 16-bit or 32-bit register, the counting will be stopped.
- Even though, the driving condition contact of this instruction (X0 in this example) is "OFF", its output contact (D2) will turn "ON" as long as the Present Value (D1) >= the set value (S) ; on the other hand, if (D1) < (S) , that will cause the (D2) = "OFF".

FNC 170	DGRYP (S) (D)	Convert BIN to Gray Code	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

• For a 16-bit instruction, S = 0~32,767 • For a 32-bit instruction, S = 0~2,147,483,647

```
X20           (S)  (D)
─┤ ├──────[ GRY  D0  K4Y20 ]
```

S : the source device (by the decimal BIN value) to be converted

D : the destination device where the converted Gary Code is stored

- When the instruction is performed, it converts the content decimal BIN value of the (S) designated source device into the Gary Code and stores the result to the designated device (D) .
- When X20 = "ON", the content value of (D0) will be converted to the Gary Code, and then the 16 continuous contacts (Y20~Y37) will be exported to the output points of the PLC.

Decimal 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 available range of the (S) is shown below. If the content value of the register designated by (S) exceeding the range, the PLC will regard that as an operational error.
  For a 16-bit instruction: 0~32,767
  For a 32-bit instruction: 0~2,147,483,647

FNC 171	DGBINP (S) (D)	Convert Gray Code to BIN	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D								●	●	●	●	●	●	○	●			

• For a 16-bit instruction, S = 0~32,767 • For a 32-bit instruction, S = 0~2,147,483,647

S : the source device (by the Gary Code) to be converted

D : the destination device where the converted decimal BIN value is stored

- When the instruction is performed, it converts the content Gary Code of the (S) designated source device into the decimal BIN value and stores the result to the designated device (D) .

- There is a kind of position detecting device called the Absolute Rotary Encoder, usually this kind of encoder generates the Gary Code signal to the PLC's input terminals. So, need to use this instruction to convert the Gray Code format to a decimal BIN value and store the result to the register in the PLC.

- When X0 = "ON", the status of those X20~X37 (16 inputs) receives the Gary Code from an Absolute Rotary Encoder, and then this instruction converts the code to a decimal BIN format value and stores it to the 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

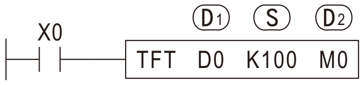
Decimal BIN Value 6,513

- The available range of the (S) is shown below. If the content value of the register designated by (S) exceeding the range, the PLC will regard that as an operational error.
  For a 16-bit instruction: 0~32,767
  For a 32-bit instruction: 0~2,147,483,647

FNC 176	TFT D1 S D2	Timer (10 ms.)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D1													●					
S													●		●	●		
D2		●	●	●	○	○												

- S = 0~32,767, will be treated as 0 if it is exceeding this range.



D1 : the Present Value of the timer (Unit: 10 ms.)

S : the set value of the timer (Unit: 10 ms.)

D2 : the output contact of the timer

- This instruction creates a timer which is using 10 ms. as the basic unit for the counting.
- When the condition contact is “ON”, this timer increases its Present Value D1 . After a while, the Present Value of the timer will equal to the set value S, that causes the timer's output contact D2 = “ON”.
- The real setting period of the timer = 10 ms. × the set value S .
- As the example diagram above:
  When X0 = “ON”, the Present Value of the register D1 will begin to do the increase counting (by every 10 ms.).
  If the Present Value of D0 reaches the set value K100 (1 second), the output contact of the timer M0 = “ON”.
  When X0 = “OFF”, the Present Value of the timer in the register D0 will be reset to “0”. And also, the output contact M0 will be reset to “OFF”.

FNC 177	TFH (D1) (S) (D2)	Timer (100 ms.)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D_1													●					
S													●		●	●		
D_2		●	●	●	○	○												
• S = 0~32,767, will be treated as 0 if it is exceeding this range.																		

```
            (D1)  (S)   (D2)
    X0
|--| |----[TFH  D0  K100  M0]
```

D_1 : the Present Value of the timer (Unit: 100 ms.)

S : the set value of the timer (Unit: 100 ms.)

D_2 : the output contact of the timer

- This instruction creates a timer which is using 100 ms. as the basic unit for the counting.
- When the condition contact is "ON", this timer increases its Present Value (D1). After a while, the Present Value of the timer will equal to the set value (S), that causes the timer's output contact (D2) = "ON".
- The real setting period of the timer = 100 ms. × the set value (S).
- As the example diagram above:
  When X0 = "ON", the Present Value of the register (D1) will begin to do the increase counting (by every 100 ms.). If the Present Value of D0 reaches the set value K100 (10 second), the output contact of the timer M0 = "ON". When X0 = "OFF", the Present Value of the timer in the register D0 will be reset to "0". And also, the output contact M0 will be reset to "OFF".

FNC 178	TFK D1 S D2	Timer (1 sec.)	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D1													●					
S													●		●	●		
D2		●	●	●	○	○												

- S = 0~32,767, will be treated as 0 if it is exceeding this range.

```
        D1   S    D2
X0
TFK  D0  K100  M0
```

D1 : the Present Value of the timer (Unit: second)
S : the set value of the timer (Unit: second)
D2 : the output contact of the timer

- This instruction creates a timer which is using a second as the basic unit for the counting.
- When the condition contact is "ON", this timer increases its Present Value D1. After a while, the Present Value of the timer will equal to the set value S, that causes the timer's output contact D2 = "ON".
- The real setting period of the timer = 1 sec. × the set value S .
- As the example diagram above:
  When X0 = "ON", the Present Value of the register D1 will begin to do the increase counting (by every second). If the Present Value of D0 reaches the set value K100 (100 second), the output contact of the timer M0 = "ON". When X0 = "OFF", the Present Value of the timer in the register D0 will be reset to "0". And also, the output contact M0 will be reset to "OFF".

MEMO

6-16 RND, DUTY, CRC and HHCMV Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS 1	2	M	3
184	RNDP (D)	Generate Random Number	○	○	○	○
186	DUTY (n1) (n2) (D)	Timing Pulse Generation				○
188	CRCP (S) (D) (n)	Cyclic Redundancy Check - 16	○	○	○	○
189	DHHCMVP (S) (D)	Hardware High-Speed Counter Data Move	○	○	○	○

FNC 184	RNDP (D)	Generate Random Number	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D								●	●	●	●	●	●		●			

```
    X0             (D)
|--| |-------[RNDP  D0]
```

D : the destination device where the generated random number is stored

- This instruction uses the content value at the (D9161, D9160) and by its calculation to generate a pseudo-random number that the range of the number is from 0 to 32,767, then the instruction stores the result to the designated device (D) .

- When X0 = “OFF” → “ON”, this instruction follows the formula below to generate a random number and stores the random number to the D0.

  Step 1: (D9161, D9160) = (D9161, D9160)×1103515245 + 12345.

  Step 2: D0 = [(D9161, D9160) > 16] ∧ (00007FFFH)

  ※ When the PLC's power is turned from “OFF” to “ON”, the value in the (D9161, D9160) will be set to K1 automatically. Also, the value in the (D9161, D9160) can not be a negative number.

- Program example:

```
  M9000
|--| |----[DINC  D7000]
  M9002
|--| |----[DBON  D7000  M0  K31]
  M0
|--| |----[DNEGP  D7000]
  M9002
|--| |----[DMOV  D7000  D9160]
  X10
|--| |----[RNDP  D10]
```

DINC D7000: The latched register (D7001, D7000) accumulates the times of scan cycle.

DBON D7000 M0 K31 / DNEGP D7000: When PLC from “STOP” → “RUN”, that takes the absolute value of the content in the (D7001, D7000), for ensuring it is a positive value.

DMOV D7000 D9160: When PLC from “STOP” → “RUN”, that moves the value in (D7001, D7000) to the (D9161, D9160) as the initial value for the RND instruction.

RNDP D10: When X10 = “OFF” → “ON”, this instruction will generate a random number and store the result at D10.

FNC 186	DUTY (n1) (n2) (D)	Timing Pulse Generation	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
n1											○	○	○			○		
n2											○	○	○			○		
D			●															

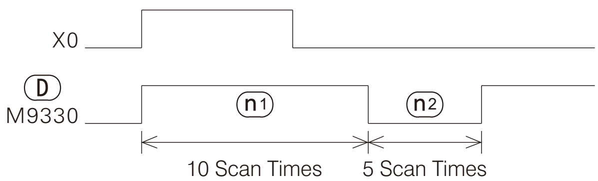
• $n_1 > 0$ • $n_2 > 0$ • D = M9330～M9334

n1 : the number of Scan Times to output "ON" in a pulse cycle

n2 : the number of Scan Times to output "OFF" in a pulse cycle

D : the timing pulse output destination device

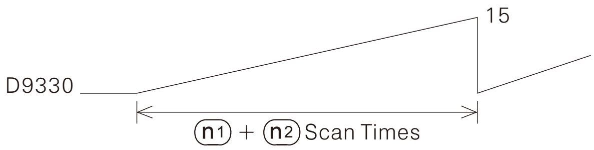
- At the destination point (D), it will get a pulse string that is using the PLC's Scan Time to switch ON/OFF. The (n1) appoints the number of Scan Times that turns the (D) to "ON". The (n2) appoints the number of Scan Times that turns the (D) to "OFF".
- When the X0 = "ON", the instruction drives the M9330 "ON" 10 Scan Times then "OFF" 5 Scan Times, this timing sequence pulse will be alternately again and again.



- When the (n1) or (n2) is equal to 0, the result of (D) is below:

  If (n1) = 0 and (n2) >= 0, the (D) is kept "OFF";

  If (n1) > 0 and (n2) = 0, the (D) is kept "ON".



- The (D) of this instruction can assign to M9330~M9334 and those 5 destination components can not repeat. Thus, the instruction can use 5 times only.
- Each the D9330~D9334 is the cooperated timing sequence cycle counter for the M9330~M9334. When the content value of the D9330~D9334 reaches the amount of (n1) + (n2) or the condition contact of this instruction turns "OFF" → "ON", the value will be reset to "0".
- When the condition contact is set to "ON" once, the operation is started. Even if the condition is turned to "OFF", the operation would not end. Also, the operation is suspended when the PLC is set to STOP. Cut off the power is the only way to stop the operation.
- If the content value of the (n1) or (n2) is less than 0, or the (D) is not assigned to the M9330~M9334, the PLC will be regarded as an operating error.
- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9330	Destination #1 for the timing sequence pulse generative instruction DUTY (FNC186).
M9331	Destination #2 for the timing sequence pulse generative instruction DUTY (FNC186).
M9332	Destination #3 for the timing sequence pulse generative instruction DUTY (FNC186).
M9333	Destination #4 for the timing sequence pulse generative instruction DUTY (FNC186).
M9334	Destination #5 for the timing sequence pulse generative instruction DUTY (FNC186).

Register ID No.	Description
■D9330	Cycle counter #1 for the timing sequence pulse generative instruction DUTY (FNC186).
■D9331	Cycle counter #2 for the timing sequence pulse generative instruction DUTY (FNC186).
■D9332	Cycle counter #3 for the timing sequence pulse generative instruction DUTY (FNC186).
■D9333	Cycle counter #4 for the timing sequence pulse generative instruction DUTY (FNC186).
■D9334	Cycle counter #5 for the timing sequence pulse generative instruction DUTY (FNC186).

FNC 188	CRCP (S) (D) (n)	Cyclic Redundancy Check - 16	1 ○	2 ○	M ○	3 ○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
n													○			○		

• When S or D is designated to the K*n*X, K*n*Y, K*n*M or K*n*S, that K*n* has to be K4 • n = 1～256

```
        X20          (S)  (D)   (n)
  |—| |————[ CRC  D0  D100  K7 ]
```

S : the head ID of continuous data source
D : the device where the result of CRC is stored
n : the number of source data to be checked

• This instruction calculates the CRC-16 (Cyclic Redundancy Check) code from the content of continuous (n) bytes (by the unit of 8-bit) data headed by (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 uses 7 consecutive 8-bit of data that headed by the D0 to calculate the CRC-16 code, and the result is stored in the 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 16-bit device as two 8-bit data. The instruction uses (n) (= K7) 8-bit data (headed by (S)) to calculate the CRC-16 code and stores the result 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 the Upper 8 bits). The instruction uses (n) (= K7) 8-bit data (headed by (S)) to calculate the CRC-16 code and stores the result 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	

FNC 189	DHHCMVP (S) (D)	Hardware High-Speed Counter Data Move	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●			●		
D													●					

• When the S or D is assigned to a HHSC related Special Register, not has the V, Z index function.

```
X20                   (S)    (D)
─┤ ├──────[DHHCMVP  D9226  D0]
```

S : the source device to be transferred
D : the destination device

- This DHHCMV is a special instruction for to instantly read the content value of the Hardware High Speed Counter (HHSC), or to instantly write a new content value or set-point value onto the HHSC.
- When the (S) is assigned to a HHSC's related Special Register, the PLC's CPU will process the following procedure. First, reads the data from the related hardware circuit. Second, stores the data to the specific Special Register (S). Last, transmits the data to the designated (D). Thus, the value at the HHSC can be read instantly.
- When the (D) is assigned to a HHSC's related Special Register, the PLC's CPU will process the following procedure. It copies the data from the designated (S) to the specific Special Register (D), then duplicates the data from the Special Register (D) to the related hardware circuit.
  The writable object of the process is the Set Value or Present Value of the HHSC. Thus, the Set Value or Present Value at the HHSC can be write instantly.
- When X20 = "OFF"→"ON", the Present Value at the hardware circuit of HHSC1 will be read and store to the Special Register (D9227, D9226) and the destination register (D1, D0).
- This instruction is a 32-bit instruction. Therefore, be sure to input DHHCMV or DHHCMVP in the program.

```
X21
─┤ ├──────[DHHCMVP  K1000  D9230]
```

- When X21 = "OFF"→"ON", the Special Register (D9231, D9230) and the Set Value at the hardware circuit of HHSC1 will be written by the constant K1000.

- The related special devices are summarized below:

Register ID No.	Description	
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 Set Value of HHSC1.
D9231	Upper 16 bits	
D9232	Lower 16 bits	The Set Value of HHSC2.
D9233	Upper 16 bits	

MEMO

6-17 Block Data Handling Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
192	D BK+ P S1 S2 D n	Block Data Addition				○
193	D BK− P S1 S2 D n	Block Data Subtraction				○
194	D BKCMP= P S1 S2 D n	Block Data Compare (S1) = (S2)				○
195	D BKCMP> P S1 S2 D n	Block Data Compare (S1) > (S2)				○
196	D BKCMP< P S1 S2 D n	Block Data Compare (S1) < (S2)				○
197	D BKCMP<> P S1 S2 D n	Block Data Compare (S1) ≠ (S2)				○
198	D BKCMP<= P S1 S2 D n	Block Data Compare (S1) ≤ (S2)				○
199	D BKCMP>= P S1 S2 D n	Block Data Compare (S1) ≥ (S2)				○

FNC 192	DBK+P S1 S2 D n	Block Data Addition	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1											●	●	●					
S_2											●	●	●			●		
D											●	●	●					
n													○			○		

- For a 16-bit instruction, S_1, S_2 and D occupies n components individually (except that S_2 is using K or H)
- For a 32-bit instruction, S_1, S_2 and D occupies (2×n) components individually (except that S_2 is using K or H)

```
X0        S1  S2   D    n
─┤ ├──── BK+P D0  D10  D20  K5
```

S_1 : the head ID of the summand block
S_2 : the addend or the head ID of the addend block
D : the head ID of the sum block
n : the length of the data block to be calculated

- When X0 = "OFF" → "ON", each component in the summand block (D0~D4) will be added to the corresponding component at the addend block (D10~D14), and store the result into the sum block (D20~D24) one by one.

S1		n	+	S2		n	=	D		n
D0	K100			D10	K10			D20	K110	
D1	K200			D11	K50			D21	K250	
D2	K234			D12	K–10			D22	K224	
D3	K400			D13	K150			D23	K550	
D4	K–1			D14	K8			D24	K7	

- If the S2 is appointed by a constant number (e.g. K100), the execution method is shown below.

S1		n	+	S2	=	D		n
D0	K100					D20	K200	
D1	K200					D21	K300	
D2	K234			K100		D22	K334	
D3	K400					D23	K500	
D4	K–1					D24	K99	

- For a 32-bit instruction, the components at the S1, S2, D and n are all organized by the 32-bit format.

```
X0        S1  S2   D    n
─┤ ├──── DBK+ D0  D10  D20  D100
```

Assume the content value of the 32-bit instruction's n at (D101, D100) is equal to K3, the treatment is as the following:

S1		n	+	S2		n	=	D		n
(D1, D0)	K123456			(D11, D10)	K123456			(D21, D20)	K246912	
(D3, D2)	K200000			(D13, D12)	K–2			(D23, D22)	K199998	
(D5, D4)	K5			(D15, D14)	K100			(D25, D24)	K105	

- The calculation may have the "Borrow" or "Carry" action, but that will not affect the corresponding flag.
- The components which used by the instruction could not be overlapped. Otherwise, the PLC will regard that as an operational error.

FNC 193	DBK – P S1 S2 D n	Block Data Subtraction	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●					
S2											●	●	●			●		
D											●	●	●					
n													○			○		

- For a 16-bit instruction, S1, S2 and D occupies n components individually (except that S2 is using K or H)
- For a 32-bit instruction, S1, S2 and D occupies (2×n) components individually (except that S2 is using K or H)

```
                 S1   S2   D    n
X0
─┤├───┤ BK–P  D0  D10  D20  K5 ├
```

S1 : the head ID of the minuend block
S2 : the subtrahend or the head ID of the subtrahend block
D : the head ID of the difference block
n : the length of the data block to be calculated

- When X0 = "OFF"→"ON", each component in the minuend block (D0~D4) will be subtracted to the corresponding component of the subtrahend block (D10~D14), and store the result into the difference block (D20~D24) one by one.

S1			S2			D	
D0	K100		D10	K10		D20	K90
D1	K200		D11	K50		D21	K150
D2	K234	—	D12	K–10	=	D22	K244
D3	K400		D13	K150		D23	K250
D4	K–1		D14	K8		D24	K–9

- If the S2 is appointed by a constant number (e.g. K100), the execution method is shown below.

S1					D	
D0	K100				D20	K0
D1	K200				D21	K100
D2	K234	—	S2 K100	=	D22	K134
D3	K400				D23	K300
D4	K–1				D24	K–101

- For a 32-bit instruction, the components at the S1, S2, D and n are all organized by the 32-bit format.

```
                 S1   S2   D    n
X0
─┤├───┤ DBK–  D0  D10  D20  D100 ├
```

Assume the content value of the 32-bit instruction's n at (D101, D100) is equal to K3, the treatment is as the following:

S1			S2			D	
(D1, D0)	K123456		(D11, D10)	K123456		(D21, D20)	K0
(D3, D2)	K200000	—	(D13, D12)	K–2	=	(D23, D22)	K200002
(D5, D4)	K5		(D15, D14)	K10		(D25, D24)	K–5

- The calculation may have the "Borrow" or "Carry" action, but that will not affect the corresponding flag.
- The components which used by the instruction could not be overlapped. Otherwise, the PLC will regard that as an operational error.

FNC	Instruction	Description	1	2	M	3
FNC 194	D BKCMP = P (S1) (S2) (D) (n)	Block Data Compare (S1) = (S2)				○
FNC 195	D BKCMP > P (S1) (S2) (D) (n)	Block Data Compare (S1) > (S2)				○
FNC 196	D BKCMP < P (S1) (S2) (D) (n)	Block Data Compare (S1) < (S2)				○
FNC 197	D BKCMP < > P (S1) (S2) (D) (n)	Block Data Compare (S1) ≠ (S2)				○
FNC 198	D BKCMP < = P (S1) (S2) (D) (n)	Block Data Compare (S1) ≤ (S2)				○
FNC 199	D BKCMP > = P (S1) (S2) (D) (n)	Block Data Compare (S1) ≥ (S2)				○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●			●		
S2											●	●	●					
D		●	●	●	○	○												
n													○			○		

- For a 16-bit instruction, S_1, S_2 and D occupies n components individually (except that S_1 is using K or H)
- For a 32-bit instruction, S_1 and S_2 occupies (2×n) components individually (except that S_1 is using K or H), also D occupies n components

```
              S1  S2  D   n
 X0
─┤├──── BKCMP=P D0 D10 M0 K5
```

S1 : the comparison value or the head ID of the comparison data block

S2 : the head ID of another comparison data block

D : the head ID of the comparison result block

n : the length of the data block to be compared

- When X0 = "OFF" → "ON", each component in the first block (D0~D4) will compare with the relevant component of the second block (D10~D14), and store the compare result to the destination block (M0~M4) one by one.

S1		"=" CMP	S2		⇨	D	
D0	K100		D10	K10		M0	0(OFF)
D1	K200		D11	K200		M1	1(ON)
D2	K234		D12	K–10		M2	0(OFF)
D3	K400		D13	K400		M3	1(ON)
D4	K–1		D14	K8		M4	0(OFF)

(n) components in each block.

- The FNC194~FNC199 are designed for to execute the =, >, <, <>, ≤ and ≥ comparisons respectively.
- The (S1) can directly use a constant number.
- When all the comparative results in the (D) block are "ON", the M9090 = "ON".
- For a 32-bit instruction, the components at the (S1), (S2) and (n) are all organized by the 32-bit format.

```
                 S1  S2  D   n
 X0
─┤├──── DBKCMP<> D0 D10 M0 D100
```

Assume the content value of the 32-bit instruction's (n) at (D101, D100) is equal to K3, the treatment is as the following:

S1		"<>" CMP	S2		⇨	D	
(D1, D0)	K123456		(D11, D10)	K123456		M0	0(OFF)
(D3, D2)	K200000		(D13, D12)	K–2		M1	1(ON)
(D5, D4)	K5		(D15, D14)	K10		M2	1(ON)

- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
■M9090	All bits "ON" flag at the result of a block data comparison BKCMP (FNC194~FNC199) instruction.

6-18 Character String Handling Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
200	DSTRP S1 S2 D	BIN to Character String Conversion				○
201	DVALP S D1 D2	Character String to BIN Conversion				○
202	$+P S1 S2 D	Join Up Two Character Strings				○
203	LENP S D	Character String Length Detection				○
204	RIGHTP S D n	Read Character from the Right of String				○
205	LEFTP S D n	Read Character from the Left of String				○
206	MIDRP S1 D S2	Extract Character from Specific Place of String				○
207	MIDWP S1 D S2	Rewrite Characters to Specific Place of String				○
208	INSTRP S1 S2 D n	Search Character String from another String				○
209	$MOVP S D	Transfer Character String				○

FNC 200	DSTRP (S1) (S2) (D)	BIN to Character String Conversion	1	2	M	3
						○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●					
S2							●	●	●	●	●	●	●	○	●	●		
D											●	●	●					

S1 : the devices to appoint the conversion format

S2 : the source of a BIN number

D : the head ID of the devices where conversed character string is stored

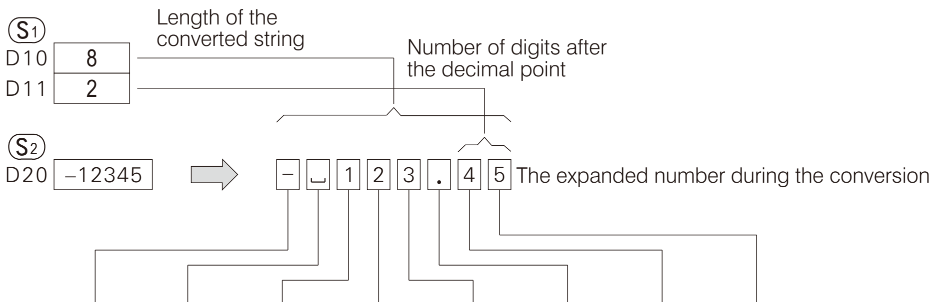
- This instruction uses the format parameters at the (S1) to interpret a BIN number at the (S2). By the format to convert each digit of that number become an ASCII code and combine those codes to a string, then stores into the devices which are headed by (D) .

(S1)	Length of the converted string	16-bit instruction, the length = 2~8; 32-bit instruction, the length = 2~13.
(S1)+1	Number of digits after the decimal point at the string	16-bit instruction, the digits = 0~5; 32-bit instruction, the digits = 0~10.

- This instruction will use the following ASCII conversion table:

Sign & Number	SPACE	–	.	0	1	2	3	4	5	6	7	8	9
ASCII Cord	20H	2DH	2EH	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H

- Assume the D10 = 8, D11 = 2, D20 = −12345 and X0 = "ON", the instruction will be executed as follows:



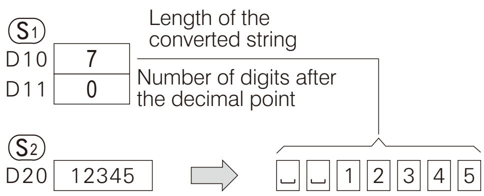
D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4
2DH(–)	20H( )	31H(1)	32H(2)	33H(3)	2EH(•)	34H(4)	35H(5)	0000H
Positive or negative sign	Space	Integer part			Decimal point	2 digits after the decimal point		End of string

The positive or negative sign is using the 20H to represent positive; the 2DH represents negative.

The byte next to the positive or negative sign is a blank space, that will fill in the code 20H.

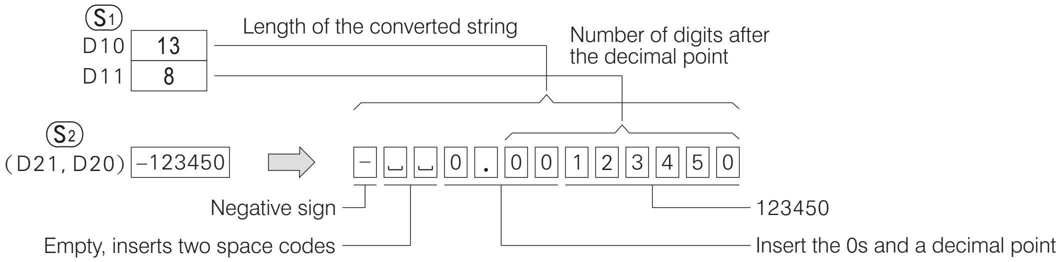
When the expanded number converts to the ASCII codes, the instruction adds the end of string at the tail automatically. If the set length of the string is an odd number, the end of string is 00H; in addition, the string with even number set length will add the end of string 0000H. Thus, the D4 at the converted string with the value 0000H that is the end of string.

- Assume the content value of the 16-bit instruction's D10 = 7, D11 = 0 and D20 = 12345, the instruction will be executed as follows:



Since the D11 = 0, that indicates an integer will be shown, so there is no decimal point in the string.

- Assume the content value of the 32-bit instruction's D10 = 13, D11 = 8 and (D21, D20) = −123450, the instruction will be executed as follows:



- For example, the S_1 = 3 and S_2 = 123, that will be regarded as an operating error. Since, the S_2 = 123 is a 3 digits number, but to convert that will need a digit for the positive or negative sign in the front. So, the S_1 should greater than or equal to 4.

FNC 201	DVALP (S) (D1) (D2)	Character String to BIN Conversion	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D1											●	●	●					
D2								●	●	●	●	●	●		●			

S : the head ID of the devices where character string is stored

D1 : the format of the character string

D2 : the converted BIN number result

- This instruction uses a string started from the (S) to perform the BIN number conversion, then stores that format into the devices (D1) and the results of BIN pure number into the device (D2).

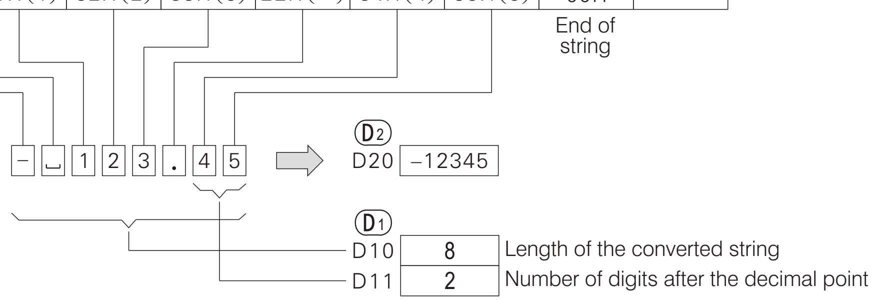
(D1)	Length of the converted string	16-bit instruction, the length = 2~8; 32-bit instruction, the length = 2~13.
(D1) + 1	Number of digits after the decimal point at the string	16-bit instruction, the digits = 0~5; 32-bit instruction, the digits = 0~10.

- This instruction will use the following ASCII conversion table:

ASCII Cord	20H	2DH	2EH	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H
Sign & Number	SPACE	−	.	0	1	2	3	4	5	6	7	8	9

- When X0 = "ON", the 16-bit instruction will be executed as follows:

(S)	D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4 Lower 8 bits	D4 Upper 8 bits
	2DH(−)	20H( )	31H(1)	32H(2)	33H(3)	2EH(•)	34H(4)	35H(5)	00H	
	Positive or negative sign	Space							End of string	

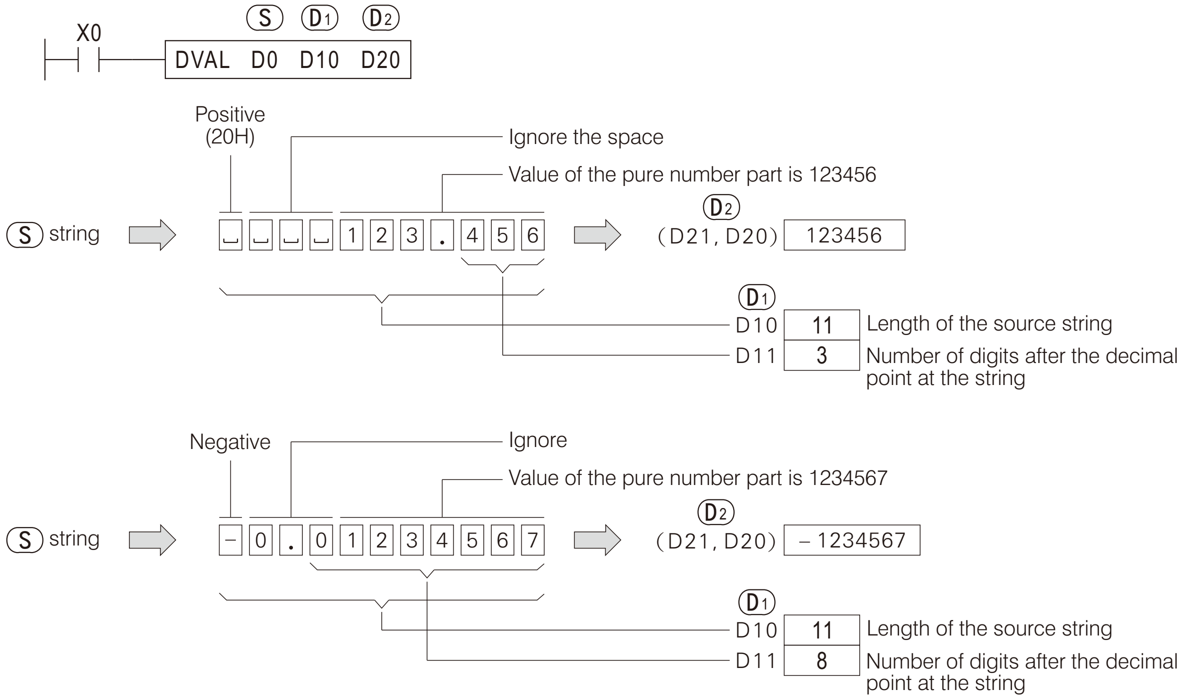


The positive or negative sign: the 20H represents positive; the 2DH represents negative.

The byte next to the positive or negative sign is a blank space, that should fill in the code 20H.

After the conversion, will get the BIN pure number part at (D2) (−12345) and the string format at (D1).

• When to execute the 32-bit instruction, the results will as follows:



• If the operation has one of the following situation, the PLC will regard that as an operational error.

1. Length of the converted string for the 16-bit instruction is not between 2~8; 32-bit instruction is not between 2~13.
2. Number of digits after the decimal point at the string for the 16-bit instruction is not between 0~5; 32-bit instruction is not between 0~10.
3. The beginning (lowest byte) of the source string S is not the ASCII code 20H (positive) or 2DH (negative).
4. The source string S contains a code which is improper.
5. The result of the converted BIN number for the 16-bit instruction is over the available range −32,768~32,767; 32-bit instruction is over −2,147,483,648~2,147,483,647.

FNC 202	$+P S1 S2 D	Join Up Two Character Strings	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●			○
S2							●	●	●	●	●	●	●		●			○
D								●	●	●	●	●	●		●			

```
X0          S1   S2   D
|-| |------[$+  D0  D10  D20]
```

S1 : the head ID of the character string #1

S2 : the head ID of the character string #2

D : the head ID of the combined string result

- The instruction will join up the string S2 after the string S1 and store the combined result string into D .
- When X0 = "ON", the instruction will be executed as follows:

S1	Upper 8 bits	Lower 8 bits
D0	20H( )	49H(I)
D1	49H(I)	4CH(L)
D2	45H(E)	4BH(K)
D3	00H	20H( )

"I ␣ LIKE ␣ "

+

S2	Upper 8 bits	Lower 8 bits
D10	49H(I)	56H(V)
D11	4FH(O)	47H(G)
D12	00H	52H(R)

"VIGOR"

=

D	Upper 8 bits	Lower 8 bits	
D20	20H( )	49H(I)	S1
D21	49H(I)	4CH(L)	S1
D22	45H(E)	4BH(K)	S1
D23	56H(V)	20H( )	
D24	47H(G)	49H(I)	S2
D25	52H(R)	4FH(O)	S2
D26	0000H		

The code for end of string will be added.

"I ␣ LIKE ␣ VIGOR"

- The instruction will add the end of string at the tail of D automatically. If the length of the result string is an odd number, the end of string is "00H"; in addition, the string with even number length will add the end of string "0000H".
- As the example instruction below, the S1 or S2 can be substituted by a key-in string between two quotation marks. Only the length of the input string should less than 32 characters.

S1	Upper 8 bits	Lower 8 bits
D0	20H( )	49H(I)
D1	49H(I)	4CH(L)
D2	45H(E)	4BH(K)
D3	00H	20H( )

"I ␣ LIKE ␣ "

+

S2 "VIGOR"

=

D	Upper 8 bits	Lower 8 bits	
D20	20H( )	49H(I)	S1
D21	49H(I)	4CH(L)	S1
D22	45H(E)	4BH(K)	S1
D23	56H(V)	20H( )	
D24	47H(G)	49H(I)	S2
D25	52H(R)	4FH(O)	S2
D26	0000H		

The code for end of string will be added.

"I ␣ LIKE ␣ VIGOR"

- If both of the S1 and S2 are empty strings (their lower bytes are "00H"), the instruction will input the result "0000H" to D .
- If two or three of the operands for the S1, S2 and D are allocated to a same component, the PLC will regard that as an operational error.

FNC 203	LENP (S) (D)	Character String Length Detection	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			

X0 ─┤ ├── LEN D0 (S) D100 (D)

S : the head ID of the character string to be measured

D : the device to store the length of the string

- To count the length of the string that is from the head component (S) to the last character (before the end of string "00H" appears) and store the result number into (D) .
- When X0 = "ON", the instruction will be executed as follows:

	Upper 8 bits	Lower 8 bits
(S) D0	20H( )	49H(I)
D1	49H(I)	4CH(L)
D2	45H(E)	4BH(K)
D3	56H(V)	20H( )
D4	47H(G)	49H(I)
D5	52H(R)	4FH(O)
D6		00H

"I ␣ LIKE ␣ VIGOR"

X0 = "ON" → (D) D100 | 12 |

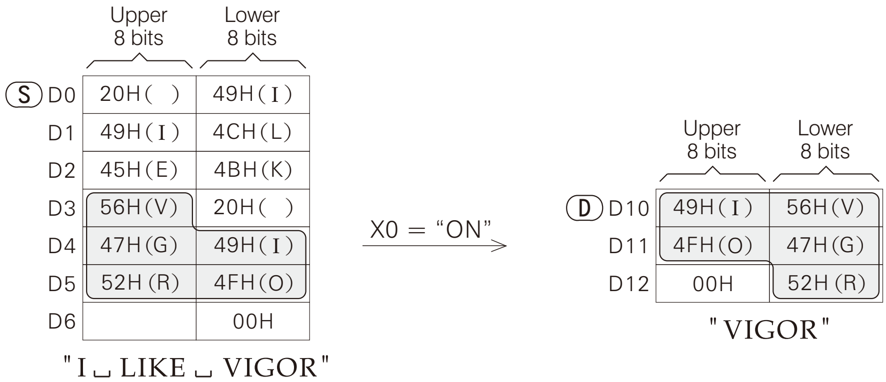
- (S) If the length of the string is more than 32,767 characters, the PLC will regard that as an operational error.

FNC 204	RIGHTP S D n	Read Character from the Right of String	1	2	M	3
						○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
n													○			○		

S : the head ID of the source character string
D : the head ID of the storage block for the read string
n : the length of the characters to be read

- To read the last of n characters from the string S and store the extracted result into D.
- When X0 = "ON", the instruction will be executed as follows:



The instruction will add the end of string at the tail of result D automatically. If the n is equal to 0 or an even number, the end of string "0000H" will be added; if the n is an odd number, the end of string is "00H".

- When the string from S which does not have the end of string code "00H" or the n is negative or the n is exceeding the string from S could provide with, the PLC will regard that as an operational error.

FNC 205	LEFTP (S) (D) (n)	Read Character from the Left of String	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
n													○			○		

S : the head ID of the source character string
D : the head ID of the storage block for the read string
n : the length of the characters to be read

- To read the beginning of (n) characters from the string (S) and store the extracted result into (D).
- When X0 = "ON", the instruction will be executed as follows:

(S)	Upper 8 bits	Lower 8 bits
D0	20H( )	49H(I)
D1	49H(I)	4CH(L)
D2	45H(E)	4BH(K)
D3	56H(V)	20H( )
D4	47H(G)	49H(I)
D5	52H (R)	4FH(O)
D6		00H

"I ␣ LIKE ␣ VIGOR"

X0 = "ON" →

(D)	Upper 8 bits	Lower 8 bits
D10	20H( )	49H(I)
D11	49H(I)	4CH(L)
D12	45H(E)	4BH(K)
D13	0000H	

"I ␣ LIKE"

The instruction will add the end of string at the tail of result (D) automatically. If the (n) is equal to 0 or an even number, the end of string "0000H" will be added; if the (n) is an odd number, the end of string is "00H".

- When the string from (S) which does not have the end of string code "00H" or the (n) is negative or the (n) is exceeding the string from (S) could provide with, the PLC will regard that as an operational error.

FNC 206	MIDRP S1 D S2	Extract Character from Specific Place of String	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
S2							●	●	●	●	●	●	●		●			

S1 : the head ID of the source character string

D : the head ID of the storage block for the read string

S2 : to appoint the patterns of the string reading

- To read S2+1 characters by the S2 designated starting position at the source string S, and store the extracted result into D .
- When X0 = "ON", the instruction will be executed as follows:

	Upper 8 bits	Lower 8 bits
S1 D0	20H( )	49H(I)
D1	49H(I)	4CH(L)
D2	45H(E)	4BH(K)
D3	56H(V)	20H( )
D4	47H(G)	49H(I)
D5	52H(R)	4FH(O)
D6		00H

"I ␣ LIKE ␣ VIGOR"

X0 = "ON" →

	Upper 8 bits	Lower 8 bits
D D10	49H(I)	4CH(L)
D11	45H(E)	4BH(K)
D12	0000H	

"LIKE"

S2 D20	3
D21	4

If the D21 = 0, this instruction will not execute.
If the D21 = –1, this instruction will extract a part of source string which is starting from the third position to the last character. That is shown in the example below.

	Upper 8 bits	Lower 8 bits
S1 D0	20H( )	49H(I)
D1	49H(I)	4CH(L)
D2	45H(E)	4BH(K)
D3	56H(V)	20H( )
D4	47H(G)	49H(I)
D5	52H(R)	4FH(O)
D6		00H

"I ␣ LIKE ␣ VIGOR"

X0 = "ON" →

	Upper 8 bits	Lower 8 bits
D D10	49H(I)	4CH(L)
D11	45H(E)	4BH(K)
D12	56H(V)	20H( )
D13	47H(G)	49H(I)
D14	52H(R)	4FH(O)
D15	0000H	

"LIKE ␣ VIGOR"

S2 D20	3
D21	–1

- If the operation has one of the following situation, the PLC will regard that as an operational error.
  1. The string from S1 which does not have the end of string code "00H".
  2. The S2 is negative.
  3. The value of S2+1 is exceeding the string from S1 could provide with, or the S2+1 is less than –1.

FNC 207	MIDWP (S1) (D) (S2)	Rewrite Characters to Specific Place of String	1	2	M	3
						○

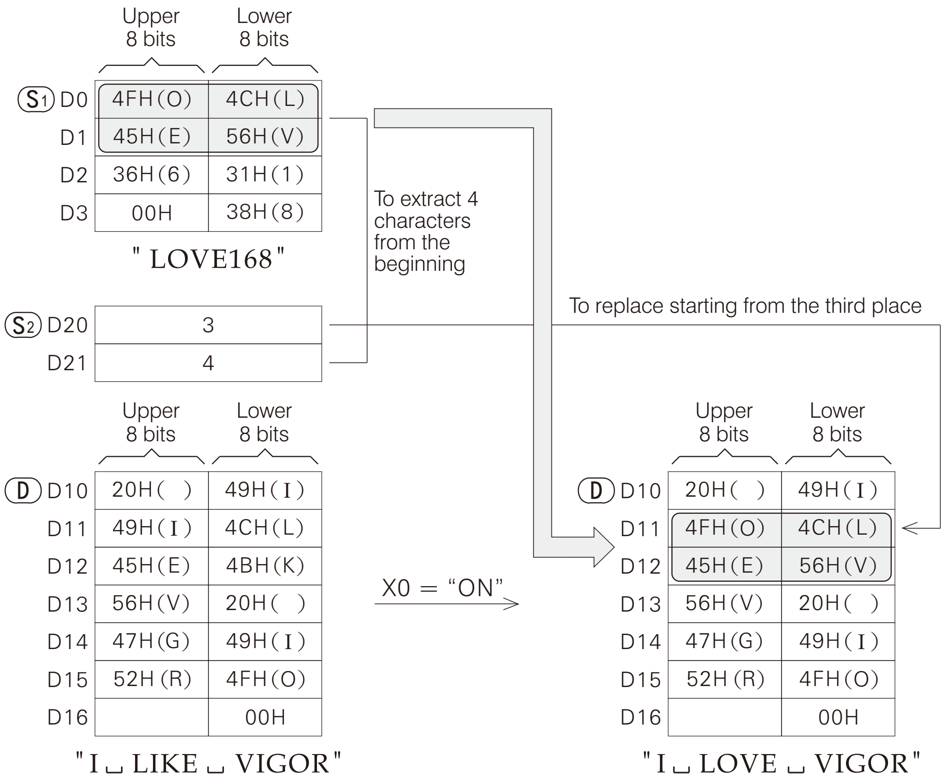
Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
S2							●	●	●	●	●	●	●		●			

S1 : the head ID of the source character string

D : the head ID of the string for the partial replacement using

S2 : to appoint the replacement patterns

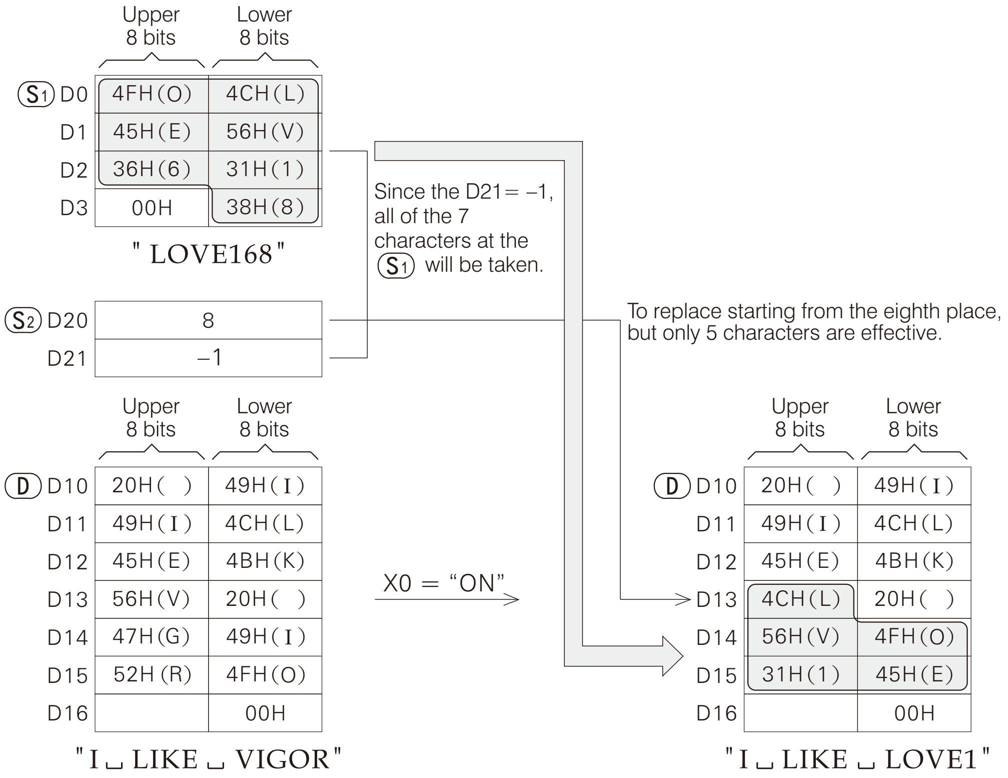
- To read the beginning (S2)+1 characters from the string (S1). Then use those to replace a part of the string (D), the replacement starting position is designated by the (S2).
- When X0 = "ON", the instruction will be executed as follows:



If the D21 ((S2)+1) is equal to 0, the instruction will not execute.

- If the D21 = –1, this instruction will extract all the characters at the source string S1. As the example below, the characters “LOVE168” are caught.

  If use the extracted characters to replace a part of string D and that range is exceeded the string D have, the replacement will only affect to the original occupied space, as the example below.



- If the operation has one of the following situation, the PLC will regard that as an operational error.
  1. The string from S1 or D which does not have the end of string code “00H”.
  2. The S2 designated starting location is not in the string D or the S2 is negative.
  3. The value of S2+1 is exceeding the string from S1 could provide with or the S2+1 is less than –1.

FNC 208	INSTR P (S1) (S2) (D) (n)	Search Character String from another String	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●					○
S2											●	●	●					
D											●	●	●					
n													○			○		

X0 —| |— INSTR D0 D10 D20 K1 (S1 = D0, S2 = D10, D = D20, n = K1)

S1 : the head ID of the searching source character string

S2 : the head ID of the string to be searched

D : the search result

n : to appoint the starting location of the search

- Use the character string S1 to execute a search that is starting from the designated n place at the source string S2 , and the result of the matched location is stored into D .

- When X0 = "ON", the instruction will be executed as follows:

S1	Upper 8 bits	Lower 8 bits
D0	49H(I)	4CH(L)
D1	45H(E)	4BH(K)
D2		00H

"LIKE"

S2	Upper 8 bits	Lower 8 bits
D10	20H( )	49H(I)
D11	49H(I)	4CH(L)
D12	45H(E)	4BH(K)
D13	56H(V)	20H( )
D14	47H(G)	49H(I)
D15	52H (R)	4FH(O)
D16		00H

"I ␣ LIKE ␣ VIGOR"

Since the n = 1, the search starts from the first one. If the n is less than 1, the instruction will not execute.

X0 = "ON" →

D	
D20	3

If the n = 5, the search will be starting from the fifth position of the string S2 . Therefore, the search cannot find a section who is equal to the S1 , so the result at the D20 will get 0.

As the example instruction below, the S1 can be substituted by a key-in string between two quotation marks.

The instruction will get the result D = D20 = 8.

X0 —| |— INSTR "VIGOR" D10 D20 K3 (S1 = "VIGOR", S2 = D10, D = D20, n = K3)

- If the operation has one of the following situation, the PLC will regard that as an operational error.
  1. The string from S1 or S2 which does not have the end of string code "00H".
  2. The n designated starting location is not in the string S2 .

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●		●			○
D								●	●	●	●	●	●		●			

X0 — $MOV D0 D10 (S = D0, D = D10)

S : the head ID of the source character string

D : the head ID of the moving target

- Move the character string from the (S) to the (D) .
- When X0 = "ON", the instruction will be executed as follows:

	Upper 8 bits	Lower 8 bits
(S) D0	49H(I)	56H(V)
D1	4FH(O)	47H(G)
D2	00H	52H (R)

" VIGOR"

X0 = "ON" →

	Upper 8 bits	Lower 8 bits
(D) D10	49H(I)	56H(V)
D11	4FH(O)	47H(G)
D12	00H	52H (R)

The code of the end of string

In the example above, the source string (S) has 5 characters that length is an odd number, thus the instruction will add the end of string "00H" at the tail of result (D) automatically. Else, if the string length is an even number, will add the end of string "0000H".

	Upper 8 bits	Lower 8 bits
(S) D0	49H(I)	4CH(L)
D1	45H(E)	4BH(K)
D2		00H

"LIKE"

X0 = "ON" →

	Upper 8 bits	Lower 8 bits
(D) D10	49H(I)	4CH(L)
D11	45H(E)	4BH(K)
D12	0000H	

The code of the end of string

- If the string from (S) which does not have the end of string code "00H", the PLC will regard that as an operational error.

6-19 Data Table Handling and Shift Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS 1	2	M	3
210	FDELP S D n	Delete Data from Specific Location of Table				○
211	FINSP S D n	Insert Data into Specific Location of Table				○
212	POPP S D n	Shift the Last Register Read (FILO Last Read)				○
213	SFRP D n	Shift n Bit Right in 16-bit Word Data with Carry				○
214	SFLP D n	Shift n Bit Left in 16-bit Word Data with Carry				○

FNC 210	FDELP (P) (S) (D) (n)	Delete Data from Specific Location of Table	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D											●	●	●					
n													○			○		
• n>0																		

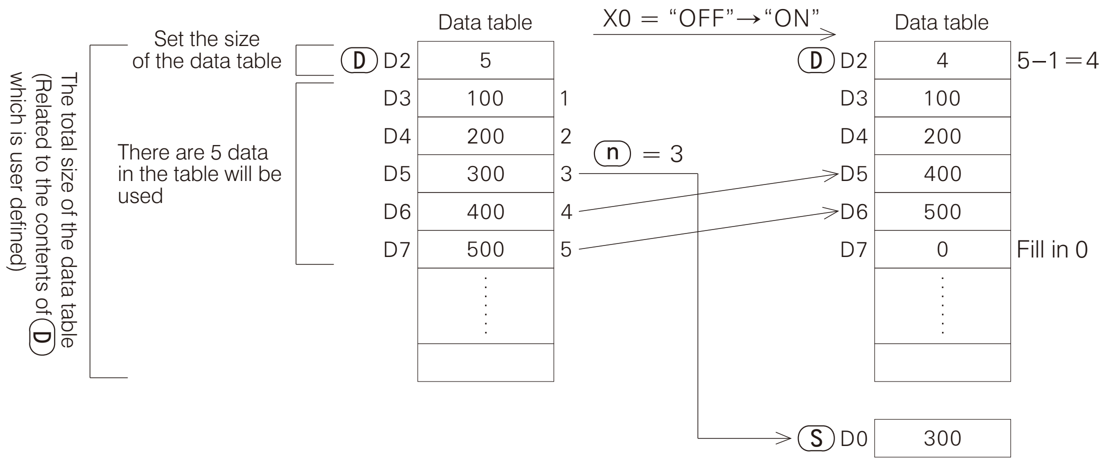
S : the register is to store the split data from the data table

D : the head ID of the data table to be shortened

n : to appoint the point at the table to be cut

- This instruction uses a series of word components starting with the (D) to form a data table, and the component (D) is used to define the size of the data table.
  When the instruction is executed, the (n) th data in the table will extract out and store to the component (S), and all the data behind will shift one position backward, also the size of the data table at the (D) will decrease one.

- When X0 = “OFF” → “ON” and (n) = (D1) = K3, the process result is below.



- Please pay attention to the total size of the data table, the programmer should manage it to avoid interferes with other data is used.

- When the instruction executes, the content value of (n) should bigger than 0, and also it should equal or less than the size of the data table at (D); otherwise, the PLC will regard that as an operational error.

FNC 211	FINSP (S) (D) (n)	Insert Data into Specific Location of Table	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●			●		
D											●	●	●					
n													○			○		
• n>0																		

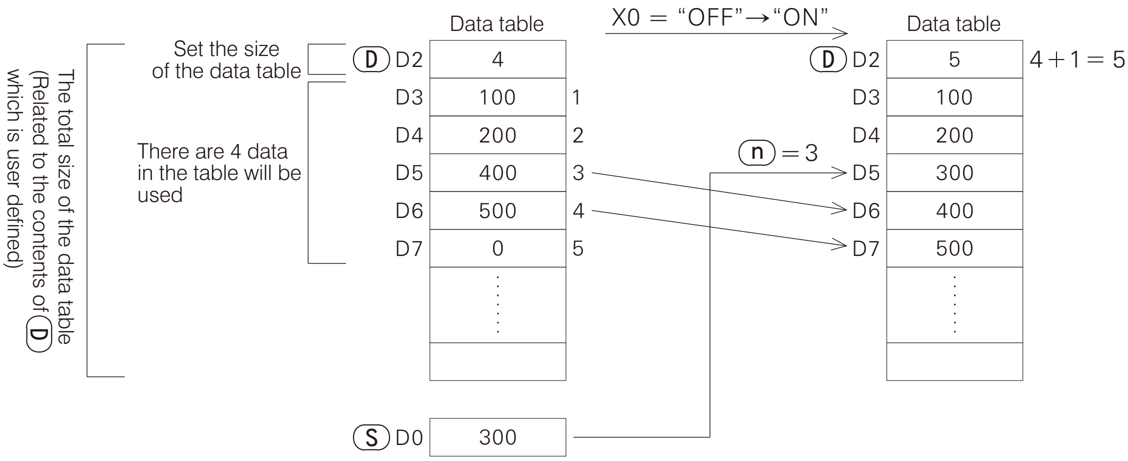
S : the register is to store the split data from the data table

D : the head ID of the data table to be shortened

n : to appoint the point at the table to be cut

- This instruction uses a series of word components starting with the (D) to form a data table, and the component (D) is used to define the size of the data table.
  When the instruction is executed, all the data starting from the (n) th component at the table will shift one position forward and insert the content of (S) into the place of (n) th in the table, also the size of the data table at the (D) will increase one.

- When X0 = "OFF" → "ON" and (n) = (D1) = K3, the process result is below.



- Please pay attention to the total size of the data table, the programmer should manage it to avoid interferes with other data is used.

- When the instruction executes, the content value of (n) should bigger than 0, and also it should equal or less than the size of the data table at (D) plus one; otherwise, the PLC will regard that as an operational error.

FNC 212	POPP (S) (D) (n)	Shift the Last Register Read (FILO Last Read)	1	2	M	3
						○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S								●	●	●	●	●	●		●			
D								●	●	●	●	●	●	○	●			
n																○		

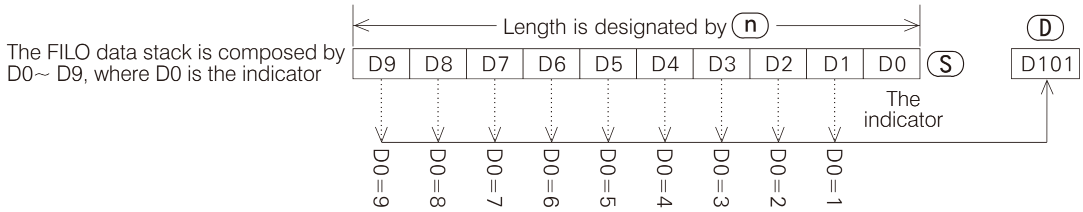
• S occupies n components • n = 2~512

S : the header ID of the FILO data stack

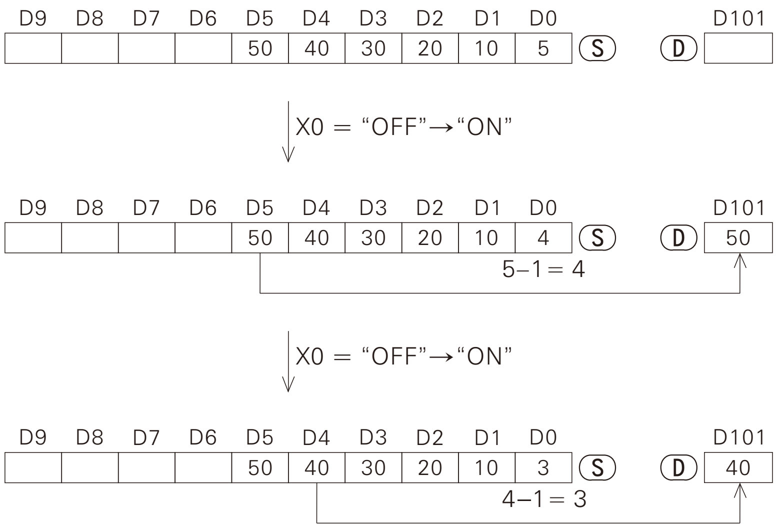
D : the destination device for storing read data from a FILO data stack

n : the length of the FILO data stack

- A series of word data can be used for the FILO (First In, Last Out) data stack access. The FILO data stack is headed by S, the size is assigned by n and the stack's first device is designated to be the indicator. When each time this instruction is enabled, it firstly reads out the content value of the device which is designated by the indicator in the FILO data stack, and then duplicates the value to the destination device D, after that the content value of the indicator decreases 1.



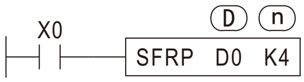
- When X0 = "OFF" → "ON" and S = D0 = 5, the content value at D5 will duplicate to D101 also the content value of D0 decreases 1 become 4. After that, if turn the X0 through "OFF" → "ON" one more time, the content value at D4 will duplicate to D101 then the content value of D0 decreases 1 become 3, and so on.



- When the content value of D0 is equal to 0 and the instruction is active again, it will not execute to duplicate data or decrease the indicator any more. So the value of D101 will remain unchanged, and furthermore the Zero Flag M9020 = "ON".
- When the content value of D0 is equal to 1 and the instruction is active again, it will decrease the indicator become 0 and furthermore the Zero Flag M9020 = "ON".
- This POP instruction can be used jointly with the FNC38 SFWR instruction to achieve the write / read control of the FILO (First In, Last Out) data stack.

FNC 213	SFRP (D) (n)	Shift n Bit Right in 16-bit Word Data with Carry	1 2 M 3 ○
FNC 214	SFLP (D) (n)	Shift n Bit Left in 16-bit Word Data with Carry	1 2 M 3 ○

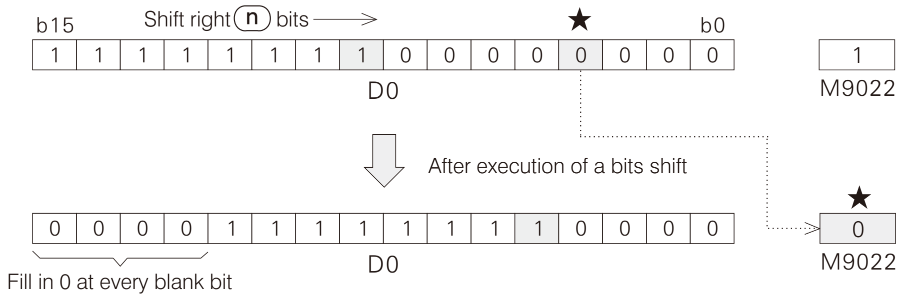
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
D								●	●	●	●	●	●	○	●			
n							○	○	○	○	○	○	○	○	○	○		
• n = 0 ~ 15																		

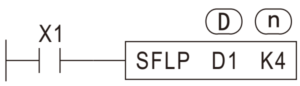


D : the word data device to be shifted

n : the number of the bits to be shifted

- The bit format of the designated device (D) executes to shift to the right by the number of (n) bits.
- When X0 = "OFF"→"ON", the 16-bit data of (D0) will be shifted by 4 bits to the right, and the status of the highest bit that shifted out will be copied to the Carry Flag M9022. All blank bits will be filled with 0 at every blank bit.

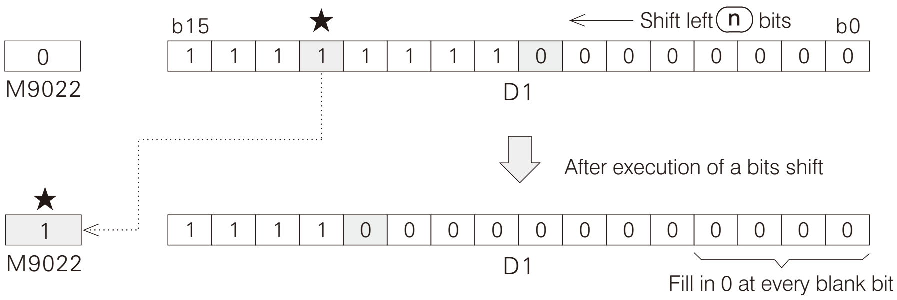




D : the word data device to be shifted

n : the number of the bits to be shifted

- The bit format of the designated device (D) executes to shift to the left by the number of (n) bits.
- When X1 = "OFF"→"ON", the 16-bit data of (D1) will be shifted by 4 bits to the left, and the status of the lowest bit that shifted out will be copied to the Carry Flag M9022. All blank bits will be filled with 0 at every blank bit.



MEMO

6-20 In-Line Comparison Instructions

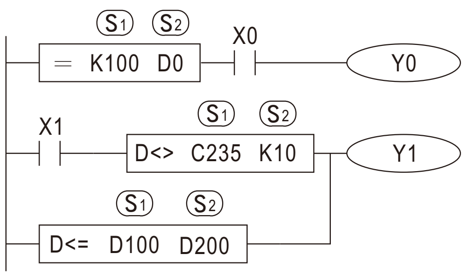
FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS 1	2	M	3
224	D= (S1) (S2)	Initial In-line Compare, Connect Up if (S1) = (S2)	○	○	○	○
225	D> (S1) (S2)	Initial In-line Compare, Connect Up if (S1) > (S2)	○	○	○	○
226	D< (S1) (S2)	Initial In-line Compare, Connect Up if (S1) < (S2)	○	○	○	○
228	D< > (S1) (S2)	Initial In-line Compare, Connect Up if (S1) ≠ (S2)	○	○	○	○
229	D< = (S1) (S2)	Initial In-line Compare, Connect Up if (S1) ≤ (S2)	○	○	○	○
230	D> = (S1) (S2)	Initial In-line Compare, Connect Up if (S1) ≥ (S2)	○	○	○	○
232	D= (S1) (S2)	Serial In-line Compare, Connect Up if (S1) = (S2)	○	○	○	○
233	D> (S1) (S2)	Serial In-line Compare, Connect Up if (S1) > (S2)	○	○	○	○
234	D< (S1) (S2)	Serial In-line Compare, Connect Up if (S1) < (S2)	○	○	○	○
236	D< > (S1) (S2)	Serial In-line Compare, Connect Up if (S1) ≠ (S2)	○	○	○	○
237	D< = (S1) (S2)	Serial In-line Compare, Connect Up if (S1) ≤ (S2)	○	○	○	○
238	D> = (S1) (S2)	Serial In-line Compare, Connect Up if (S1) ≥ (S2)	○	○	○	○
240	D= (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) = (S2)	○	○	○	○
241	D> (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) > (S2)	○	○	○	○
242	D< (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) < (S2)	○	○	○	○
244	D< > (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) ≠ (S2)	○	○	○	○
245	D< = (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) ≤ (S2)	○	○	○	○
246	D> = (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) ≥ (S2)	○	○	○	○

FNC	Ladder	Description	1	2	M	3
FNC 224	D= (S1) (S2)	Initial In-line Compare, Connect Up if (S1) = (S2)	○	○	○	○
FNC 225	D> (S1) (S2)	Initial In-line Compare, Connect Up if (S1) > (S2)	○	○	○	○
FNC 226	D< (S1) (S2)	Initial In-line Compare, Connect Up if (S1) < (S2)	○	○	○	○
FNC 228	D<> (S1) (S2)	Initial In-line Compare, Connect Up if (S1) ≠ (S2)	○	○	○	○
FNC 229	D<= (S1) (S2)	Initial In-line Compare, Connect Up if (S1) ≤ (S2)	○	○	○	○
FNC 230	D>= (S1) (S2)	Initial In-line Compare, Connect Up if (S1) ≥ (S2)	○	○	○	○
FNC 232	D= (S1) (S2)	Serial In-line Compare, Connect Up if (S1) = (S2)	○	○	○	○
FNC 233	D> (S1) (S2)	Serial In-line Compare, Connect Up if (S1) > (S2)	○	○	○	○
FNC 234	D< (S1) (S2)	Serial In-line Compare, Connect Up if (S1) < (S2)	○	○	○	○
FNC 236	D<> (S1) (S2)	Serial In-line Compare, Connect Up if (S1) ≠ (S2)	○	○	○	○
FNC 237	D<= (S1) (S2)	Serial In-line Compare, Connect Up if (S1) ≤ (S2)	○	○	○	○
FNC 238	D>= (S1) (S2)	Serial In-line Compare, Connect Up if (S1) ≥ (S2)	○	○	○	○
FNC 240	D= (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) = (S2)	○	○	○	○
FNC 241	D> (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) > (S2)	○	○	○	○
FNC 242	D< (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) < (S2)	○	○	○	○
FNC 244	D<> (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) ≠ (S2)	○	○	○	○
FNC 245	D<= (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) ≤ (S2)	○	○	○	○
FNC 246	D>= (S1) (S2)	Parallel In-line Compare, Connect Up if (S1) ≥ (S2)	○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		

S1 : the compare value #1

S2 : the compare value #2



When the content value of D0 = K100 and X0 = “ON”, Y0 = “ON”.

When X1 = “ON” and the Present Value of C235 ≠ K10, Y1 = “ON”.

Besides, when the content value of (D101, D100) ≤ the content value of (D201, D200), Y0 = “ON” too.

- Since the programming software Ladder Master S at the monitoring mode could not to read the status at the special module’s BFM directly, the operand in the instruction which is appointed to the UnG will not have the correct value to display. Furthermore, that in-line comparison could not display the result. However, the logical processing at the PLC is correct.

6-21 Handy Instructions and DABIN, BINDA, HSCT Instructions

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
256	D LIMIT P S1 S2 S3 D	Limit Control				○
257	D BAND P S1 S2 S3 D	Dead Band Control				○
258	D ZONE P S1 S2 S3 D	Zone Shift Control				○
259	D SCL P S1 S2 D	Scaling (Coordinate by Point Data)	○	○	○	○
269	D SCL2 P S1 S2 D	Scaling 2 (Coordinate by X/Y Data)	○	○	○	○
260	D DABIN P S D	Convert Decimal ASCII String to BIN Number				○
261	D BINDA P S D	Convert BIN Number to Decimal ASCII String				○
280	D HSCT S1 m S2 D n	Software High Speed Counter Table Compare	○	○	○	○

FNC 256	DLIMITP S1 S2 S3 D	Limit Control	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1							●	●	●	●	●	●	●		●	●		
S_2							●	●	●	●	●	●	●		●	●		
S_3							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			

• $S_2 > S_1$

```
                      S1    S2     S3  D
 X0
─┤ ├────┤LIMIT  K500  K1000  D0  D1│
```

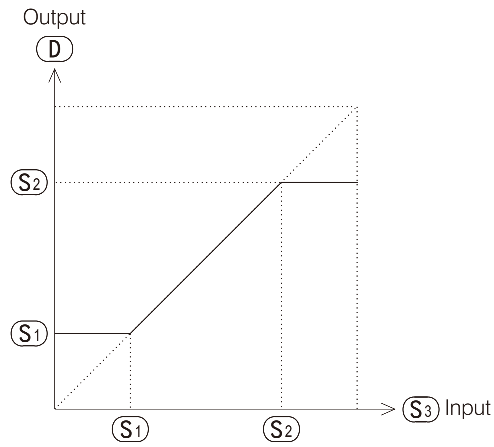
S_1 : the lower limit value
S_2 : the upper limit value
S_3 : the input value
D : the output value

- Use the lower limit S_1 and the upper limit S_2 to create a conversion curve as shown in the right diagram. Through that curve, an input value from the S_3 can get a restrictive output value at the D .

  If the S_3 < S_1 , the output D is equal to the lower limit S_1 .

  If the S_1 <= S_3 <= S_2 , the output D is equal to the input S_3 .

  If the S_3 > S_2 , the output D is equal to the upper limit S_2 .



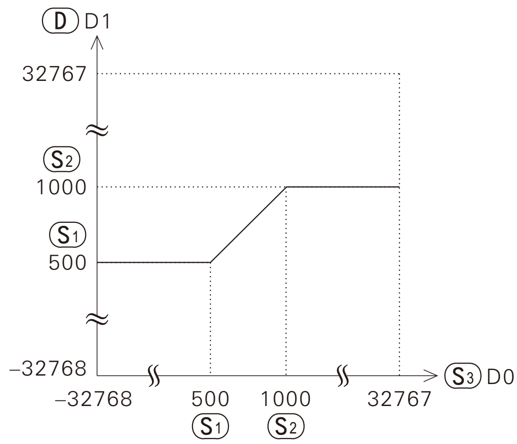
- When X0 = "ON", the instruction and the input value D0 use the two limits at the right diagram to generate a restricted result D1.

  By the setting at the right diagram, the S_1 = 500 and S_2 = 1000:

  If the D0 < 500, the output D1 is equal to 500.

  If the 500 <= D0 <= 1000, the output D1 is equal to the input D0.

  If the D0 > 1000, the output D1 is equal to 1000.



- For a 16-bit instruction, the available range of the S_3 and D are –32,768~+32,767.
  For a 32-bit instruction, the available range of the S_3 and D are –2,147,483,648~+2,147,483,647.

FNC 257	D BAND P (S1) (S2) (S3) (D)	Dead Band Control	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1							●	●	●	●	●	●	●		●	●		
S_2							●	●	●	●	●	●	●		●	●		
S_3							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			
• $S_1 < S_2$																		

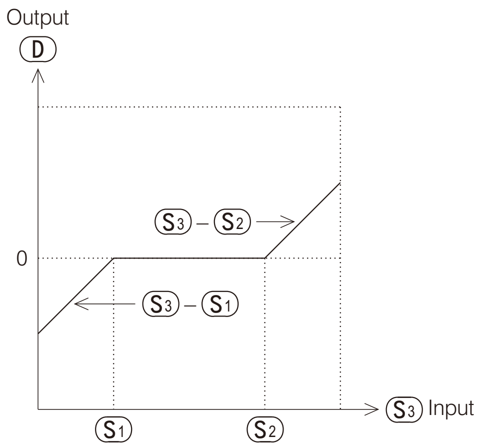
```
       X0          (S1)    (S2)   (S3) (D)
--| |------[ BAND  K-5000  K5000  D0   D1 ]
```

S_1 : the lower unresponsive value of the dead band

S_2 : the upper unresponsive value of the dead band

S_3 : the input value of the dead band control

D : the output value of the dead band control

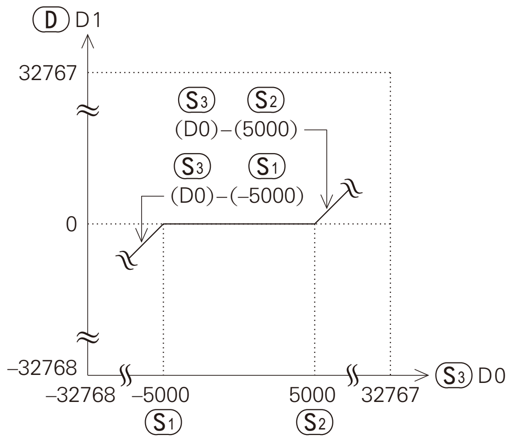


- Use the values of lower unresponsive (S1) and upper unresponsive (S2) to create a dead band conversion curve as shown in the right diagram. Through that curve, an input value from the (S3) can get an adjusted output value at the (D) .

  If the (S3) < (S1) , the output (D) is equal to the arithmetical difference of (S3) − (S1) .

  If the (S1) <= (S3) <= (S2) , the output (D) is equal to 0.

  If the (S3) > (S2), the output (D) is equal to the arithmetical difference of (S3) − (S2) .



- When X0 = "ON", the instruction and the input value D0 use the dead band between two unresponsive points at the right diagram to generate an adjusted result D1.

  By the setting at the right diagram, the (S1) =−5000 and (S2) = 5000:

  If the D0 < –5000, the output D1 is equal to D0 − (−5000).

  If the −5000 <= D0 <= 5000, the output D1 is equal to 0.

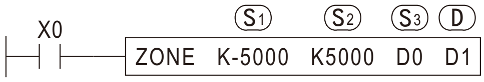
  If the D0 > 5000, the output D1 is equal to D0 − (5000).

- For a 16-bit instruction, the available range of the (S3) and (D) are −32,768∼+32,767.
  For a 32-bit instruction, the available range of the (S3) and (D) are −2,147,483,648∼+2,147,483,647.

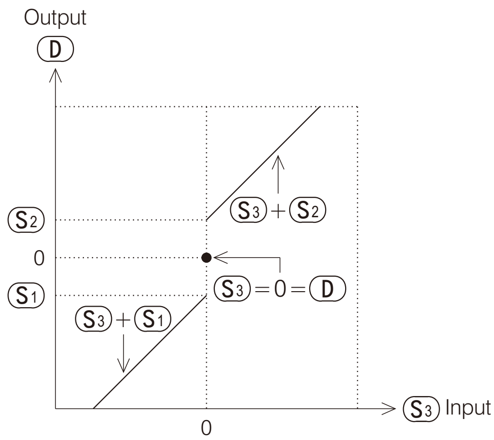
FNC 258	DZONEP S1 S2 S3 D	Zone Shift Control	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●	●		
S2							●	●	●	●	●	●	●		●	●		
S3							●	●	●	●	●	●	●		●			
D								●	●	●	●	●	●		●			

- S1 < S2



S1 : the negative bias value of the shift control
S2 : the positive bias value of the shift control
S3 : the input value of the zone shift control
D : the output value of the zone shift control

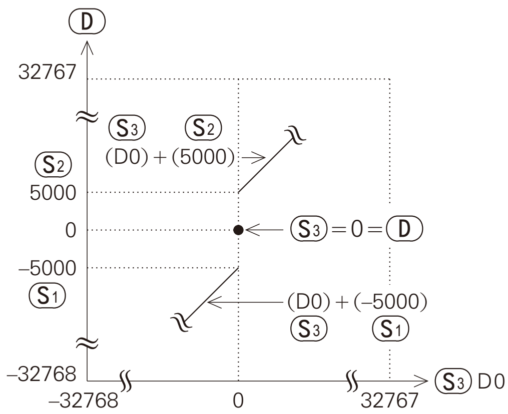


- Use the values of negative bias S1 and positive bias S2 to create two shift control conversion curves as shown in the right diagram. Through the curves, an input value from the S3 can get an adjusted output value at the D .

  If the S3 < 0, the output D is equal to the amount of S3+S1.

  If the S3 = 0, the output D is equal to 0.

  If the S3 > 0, the output D is equal to the amount of S3+S2.



- When X0 = "ON", the instruction and the input value D0 use the two bias values and curves at the right diagram to generate a shifted result D1.

  By the setting at the right diagram, the S1 = −5000 and S2 = 5000:

  If the D0 < 0, the output D1 is equal to D0 + (–5000).

  If the D0 = 0, the output D1 is equal to 0.

  If the D0 > 0, the output D1 is equal to D0 + (5000).

- For a 16-bit instruction, the available range of the S3 and D are –32,768~+32,767.
  For a 32-bit instruction, the available range of the S3 and D are –2,147,483,648~+2,147,483,647.

FNC 259	DSCLP S1 S2 D	Scaling (Coordinate by Point Data)	1 ○	2 ○	M ○	3 ○
FNC 269	DSCL2P S1 S2 D	Scaling 2 (Coordinate by X/Y Data)	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●		●	●		
S2													●					
D								●	●	●	●	●	●		●			

```
      X0          S1   S2    D
 |-| |----[ SCL  D0  D1000  D1 ]
```

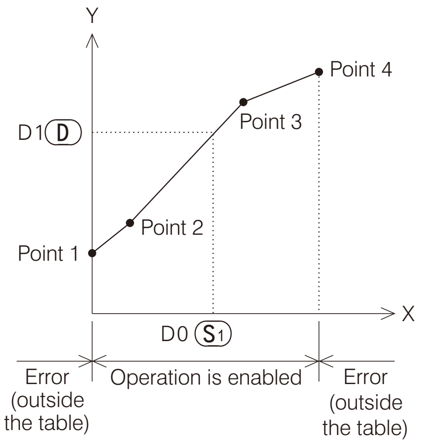
S1 : the source device to store the input value of scaling (X coordinate)

S2 : the head register ID of the conversion data table is used

D : the device to store the result value of scaling (Y coordinate)

- When this SCL instruction is active, it will base on the specified input value at S1 (X coordinate) and the conversion data table started from the S2, then by the comparison and proportion to produce a result (Y coordinate) afterwards stores that to the specified device D.

- When X0 = "ON", the content value at the source device D0 and the conversion data table (that is started from the D1000) will be used to calculate the scaled result, then the result will be stored at the destination device D1.



For 16-bit SCL instruction:
Conversion data table S2

Number of Coordinate Points		S2	D1000
Point 1	X	S2+1	D1001
	Y	S2+2	D1002
Point 2	X	S2+3	D1003
	Y	S2+4	D1004
Point 3	X	S2+5	D1005
	Y	S2+6	D1006
Point 4	X	S2+7	D1007
	Y	S2+8	D1008

For 32-bit SCL instruction:
Conversion data table S2

Number of Coordinate Points		S2+1, S2
Point 1	X	S2+3, S2+2
	Y	S2+5, S2+4
Point 2	X	S2+7, S2+6
	Y	S2+9, S2+8
Point 3	X	S2+11, S2+10
	Y	S2+13, S2+12
Point 4	X	S2+15, S2+14
	Y	S2+17, S2+16

- It is an easy and effective solution for to transform the values between analog I/O and external physical quantity (such as weight, length, temperature....).

- If the result data (Y coordinate) is not an integer, round the number to the nearest integer.

- If two or more points in the conversion data table are specified their X coordinate values to a same number, moreover the content value of the source S1 is just equal to this number, the result value at D will be the Y coordinate value of the second point.

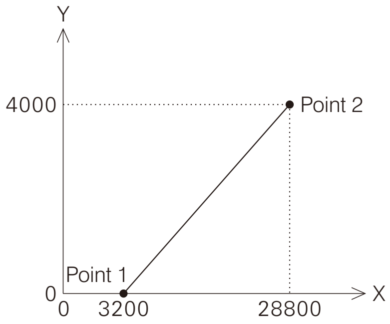
- Any one of the following mistake may cause an operating error: (the Operation Error flag M9067 = "ON" and the error code in D9067 = K9067 .)
  1. At the conversion data table, those Xn are not edited by the ascending order. (do not let Xn > Xn+1)
  2. When the input source value S1 exceeds the range of the conversion data table.
  3. When the difference between two adjacent points (Xn and Xn+1 or Yn and Yn+1) is more than "65,535".

- Application example: To transform the position from a linear potentiometer

  Use the 1st. channel of a VS-4AD module to connect with a 500 mm stroke linear potentiometer for measuring the position. This application in the machine just needs to use a part of the potentiometer (50 mm to 450 mm). The request is to get a value which is between 0 to 4,000 (unit: 0.1 mm).

  Input a standard 10 VDC to the 0~500 mm potentiometer, it should return a relative 0~10 V analog signal to the AIN1+/AIN− of the VS-4AD's AI1 channel, then at the module's BFM#5 will have a value between 0~32,000. Since this application only uses the segment between 50~450 mm in the meter, so at the BFM#5 will get 3,200~28,800.

  The chart below is to show the conversion curve of this application.



X axis represents analog input value.
Y axis represents the current position (Unit : 0.1mm)

Conversion data table for SCL instruction

	Item		Register ID	Content value
S2	Number of Coordinate Points		D1000	2
S2+1	Point 1	X	D1001	3200
S2+2		Y	D1002	0
S2+3	Point 2	X	D1003	28800
S2+4		Y	D1004	4000

```
 M9000
--| |----+--[FROM K1 K5 D0 K1]   To read the VS-4AD's analog value (BFM#5) and put it to the D0.
         |
         +--[SCL D0 D1000 D1]    By the SCL instruction, to transform at the location of the potentiometer and
                                 put the position data into D1. (unit: 0.1 mm)
```

- Functionally, both the SCL2 and SCL instructions are the same. The difference between these instructions are the arrangement of data in the conversion table. The tables below are the way to organize the data tables of the SCL2 instruction. (assume the total number of the points is 4)

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

FNC 260	D DABIN P (S) (D)	Convert Decimal ASCII String to BIN Number	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S											●	●	●					
D								●	●	●	●	●	●	○	●			

• For a 16-bit instruction, S occupies 3 components • For a 32-bit instruction, S occupies 6 components

```
 X0              (S) (D)
-| |----[DABIN  D0  D10]
```

S : the head ID of the devices where the decimal ASCII string is stored

D : the converted BIN number result

- This instruction uses a decimal format ASCII string number started from the (S) to perform the BIN number conversion, then stores the result into the device (D) .

- This instruction will use the following ASCII conversion table:

ASCII Cord	00H	20H	2DH	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H
Sign & Number	NULL	SPACE	–	0	1	2	3	4	5	6	7	8	9

- When the X0 = "ON", this 16-bit instruction converts the content of an ASCII code string at D0~D2, then stores the result of a BIN number to the D10.

(S)	D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits
	Sign code	10^4	10^3	10^2	10^1	10^0
	2DH(–)	31H(1)	32H(2)	35H(5)	38H(8)	39H(9)

(D)
D10 | –12589 |

The lowest byte in the string is the sign code for the conversion. If the sign is 2DH, the number is negative; otherwise, 20H means positive.
The remaining bytes are to store the data of a value. If the ASCII codes in this segment of value have the 20H, 00H or 30H, that will be treated as 30H.
For a 16-bit instruction, the available range is –32,768~+32,767.

- For a 32-bit instruction, the structure of the source string starting from the (S) is shown below. (assume the (S) is assigned to the D0)

(S)	D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4 Lower 8 bits	D4 Upper 8 bits	D5 Lower 8 bits	D5 Upper 8 bits
	Sign code	10^9	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	Ignore

For a 32-bit instruction, the available range is –2,147,483,648~+2,147,483,647. Therefore, with a sign code, that will occupy 11 bytes.

- If the source string (S) contains a code which is improper or the result is exceeded to the available range, the PLC will regard that as an operational error.

FNC 261	DBINDAP (S) (D)	Convert BIN Number to Decimal ASCII String	1	2	M	3
						○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D											●	●	●					

• For a 16-bit instruction, D occupies 4 components • For a 32-bit instruction, D occupies 6 components

```
 X0                  (S)  (D)
─┤ ├──────────[ BINDA  D10  D0 ]
```

S : the source BIN number to be converted

D : the head ID of the converted decimal ASCII string

- This instruction uses a BIN number at the (S) to perform the conversion and the result is using a decimal format ASCII string to explain the number, then stores the string into the devices started from (D).
- This instruction will use the following ASCII conversion table:

Sign & Number	NULL	SPACE	–	0	1	2	3	4	5	6	7	8	9
ASCII Cord	00H	20H	2DH	30H	31H	32H	33H	34H	35H	36H	37H	38H	39H

- When the X0 = "ON", this 16-bit instruction converts the content of a BIN number in the D10, then stores the result using an ASCII code string to D0~D3.

(S) D10 | –12589 |

(D)

D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3
Sign code	10^4	10^3	10^2	10^1	10^0	End of string
2DH(–)	31H(1)	32H(2)	35H(5)	38H(8)	39H(9)	0H or retain

If the M9091 = "OFF", will fill in 0000H.
If the M9091 = "ON", will retain the original data at D3.

If the D10 is positive, the sign code will fill in 20H;
else the D10 is negative, the sign code will fill in 2DH.
Assume the content value at the D10 is 125, the result is below:

(D)

D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3
20H( )	20H( )	20H( )	31H(1)	32H(2)	35H(5)	0H
It is positive.	Empty, fill in 20H.		1	2	5	Assume the M9091 = "OFF".

- For a 32-bit instruction, the structure of the result string starting from the (D) is shown below. (assume the (D) is assigned to the D0)

(D)

D0 Lower 8 bits	D0 Upper 8 bits	D1 Lower 8 bits	D1 Upper 8 bits	D2 Lower 8 bits	D2 Upper 8 bits	D3 Lower 8 bits	D3 Upper 8 bits	D4 Lower 8 bits	D4 Upper 8 bits	D5 Lower 8 bits	D5 Upper 8 bits
Sign code	10^9	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	0H or 20H

For a 32-bit instruction, the available range is
–2,147,483,648 ~ +2,147,483,647.
Therefore, with a sign code, that will occupy 11 bytes.

If the M9091 = "OFF", will fill in 00H.
If the M9091 = "ON", will fill in 20H.

- The related special device for this instruction:

Relay ID No.	Description
M9091	Assign the BINDA operating mode. 16-bit instruction: When M9091="OFF", will add the end of string 0000H after the result. When M9091="ON", will only convert the data without to add the end of string. 32-bit instruction: When M9091="OFF", will add the end of string 00H at the result's last upper 8 bits. When M9091="ON", will add the end of string 20H at the result's last upper 8 bits.

FNC 280	DHSCT S1 m S2 D n	Software High Speed Counter Table Compare	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1													●					
m																○		
S2												●						
D		●	●	●														
n																○		

• 1≤m≤16 • S2 = C235~C255 • When D is assigned to Y, only Y0~Y7 have instant response ability • 1≤n≤16

```
              S1  m  S2   D   n
X20
|-| |----[DHSCT D0 K6 C235 Y4 K3]
```

S1 : the head register ID of the comparison table

m : the number of data sets to be compared

S2 : the ID of the designated Software High Speed Counter

D : the head ID of output components for the comparison results

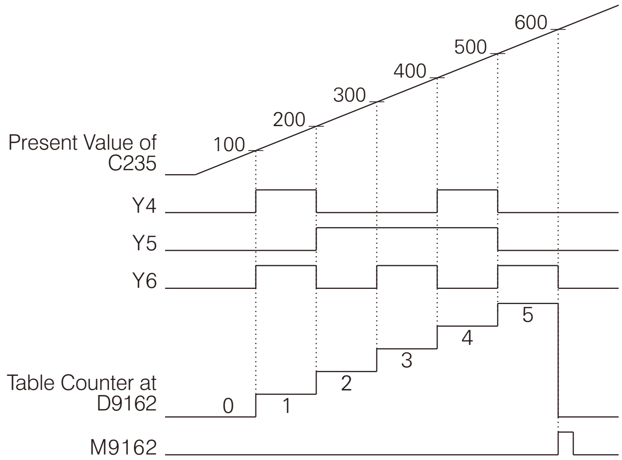
n : the number of output components for the comparison results

- This instruction uses a comparison table which is headed by the S1 and continuously occupies m × 3 registers, then takes the Present Value of the S2 designated Software High Speed Counter to compare to that table, furthermore the comparison result presents through the D headed continuously n components. Both the values comparison and result outputting at this instruction are operated by the interruption of system.

S1 Comparison Table

Data Set to be Compared	Output Appointer	D9162 Table Counter
(D1, D0) K100	D2 H05	0
(D4, D3) K200	D5 H02	1
(D7, D6) K300	D8 H06	2
(D10, D9) K400	D11 H03	3
(D13, D12) K500	D14 H04	4
(D16, D15) K600	D17 H00	5

m



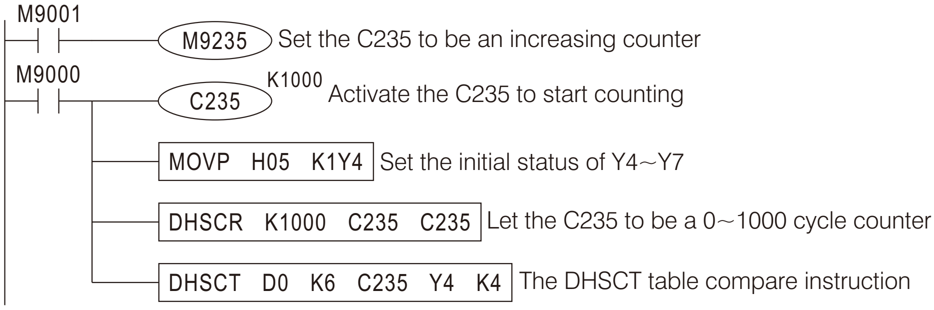
- When X20 = "ON", this instruction starts operation. The Present Value of C235 SHSC assigned by S2 will compare with the content value of the 1st data set (D1, D0) in the comparison table. If the input makes those values equal, the content value of the 1st output appointer (D2) in the comparison table will be instantly output through the Y4~Y6 (assigned by D and n), also it increases the value in the table counter D9162 by 1 (become 1 from 0). After that, the Present Value of C235 will compare with the content value of the 2nd data set (D4, D3) in the table. When the values are equal, the content value of the 2nd output appointer (D5) in the table will be instantly output through the Y4~Y6, also it increases the value in the table counter D9162 by 1 (become 2 from 1). The comparison action is continuous, until the value of the last data set in the table is equal to the Present Value of C235. That causes the operation completion flag M9162 for one Scan Time and the D9162 will be reset to 0, furthermore the comparison operation should start over again from the 1st data set.
- When the X20 turns from "ON" to "OFF", this instruction stops and the content value of D9162 will be reset to 0 but the result status which has output previously will still remain.
- This instruction is a 32-bit instruction. Therefore, be sure to input DHSCT in the program.
- This instruction can only be used once in the program.
- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
■M9162	To indicate the completion of HSCT instruction.

Register ID No.	Description
■D9162	The number of count is activated at the table of HSCT instruction.

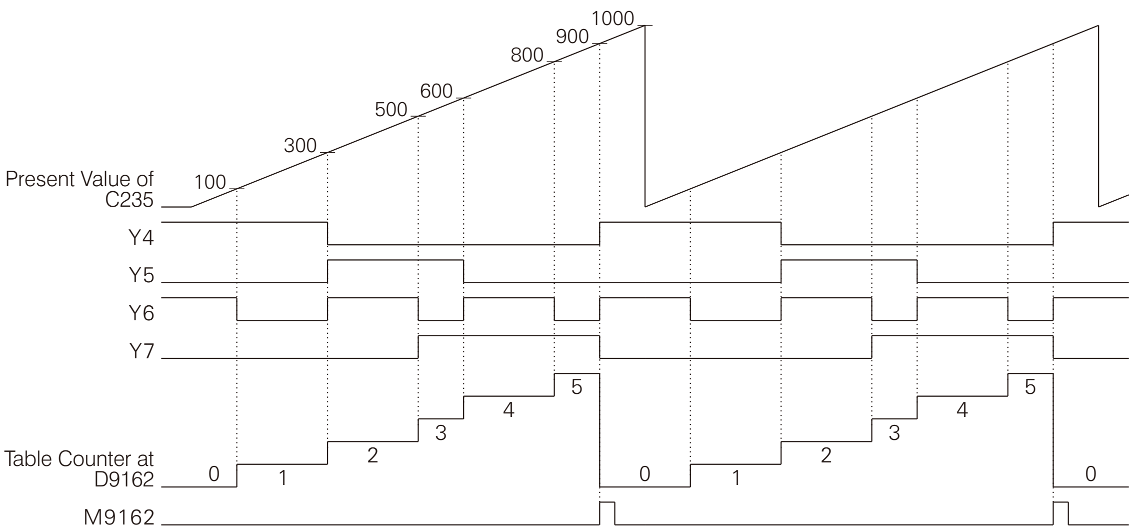
Program Example:

This example uses the C235, it will count the number of pulses from the input point X0. The program arranges the C235 become an increasing cycle counter and limits its Present Value between 0 to 1000. During the counting process, the DHSCT instruction generates output signals according to the contents in the comparison table.



Comparison Table

Data Set to be Compared	Output Appointer	Table Counter (D9162)
(D1, D0) K100	D2 H1	0
(D4, D3) K300	D5 H6	1
(D7, D6) K500	D8 HA	2
(D10, D9) K600	D11 HC	3
(D13, D12) K800	D14 H8	4
(D16, D15) K900	D17 H5	5



6-22 Positioning Control Instructions

The VS series PLCs are provided with 4 axes control function. Given that to use positing control function is more than understanding positioning control instructions, more professional description is needed. Thus, this manual provides a special chapter 8 “Statement of Positioning Control Functions” for detailed description about the function.
Hereby provides the table of positioning control instructions for reference.

FNC No.	Mnemonic in Ladder Diagram	Function Description	Applicable VS			
			1	2	M	3
300	DZRN S1 S2 D1 D2	Zero (Home Position) Return	○	○	○	○
301	DJOGF S1 S2 D1 D2	Jog Forward	○	○	○	○
302	DJOGR S1 S2 D1 D2	Jog Reverse	○	○	○	○
303	DDRVR S1 S2 D1 D2	Drive to Relative Position	○	○	○	○
304	DDRVA S1 S2 D1 D2	Drive to Absolute Position	○	○	○	○
305	DDV2R S1 S2 D1 D2	Drive to Relative Position by 2 Stages	○	○	○	○
306	DDV2A S1 S2 D1 D2	Drive to Absolute Position by 2 Stages	○	○	○	○
307	DDVIT S1 S2 D1 D2	Interrupt Constant Quantity Positioning	○	○	○	○
308	DDV2I S1 S2 D1 D2	2 Stages Interrupt Constant Quantity Positioning	○	○	○	○
309	DDVSR S1 S2 D1 D2	Interrupt to Stop or Drive to Relative Position	○	○	○	○
310	DDVSA S1 S2 D1 D2	Interrupt to Stop or Drive to Absolute Position	○	○	○	○
311	DPLSV S D1 D2	Variable Speed Pulse Output	○	○	○	○
312	DTBL S1 S2 D1 D2	Data Table Positioning	○	○	○	○
313	DABS S D1 D2	Absolute Current Value Read	○	○	○	○
314	MPG S1 S2 D1 D2	Handwheel Positioning	○	○	○	○
315	DLIR S D	Relatively Linear Interpolation	○	○	○	○
316	DLIA S D	Absolutely Linear Interpolation	○	○	○	○

MEMO

7. Statement of Communication Functions

In the field of industrial automatic control, communication between equipment has become an extremely important part. The abilities to transmit data mutually, collaborate and communicate with each other to complete the control work together are highly valued. PLCs are the most basic and most common control equipment in industrial automatic control, and thus naturally it is also very important to enhance their communication capacity.
According to the different model of the VS series PLC, could have 3 to 6 communication ports.
Those are not only equipped to execute the programming, but also vertically uplink with the Human-Machine Interface (HMI) or Supervisory Control And Data Acquisition system (SCADA) to construct a monitoring network; horizontally, it can be connected to multiple VS PLCs to operate the distributed control. In addition, can be the master to connect with peripheral equipments (such as thermostat, inverter, etc.). Therefore, these create a complete control system.
Compared to the company's previous products, the VS series PLC has made great progress in increasing the number of communication ports, communication speeds, the flexibility of application types and user-friendliness.
This chapter is specifically for the VS series PLC communication function to provide a detailed description, and provides useful help when users operate these communication functions.

7-1 Key Points of Communication Functions

7-1-1 Fundamental Object about the Communication Function

● Communication Interface

The communication interface which can go with the VS series PLC is including the USB, RS-232 and RS-485.

- USB interface — the most popular interface used on computers. The VS PLC's main units are all equipped with a built-in USB port to be the programming interface of the PLC and that with a speed rate up to 12 Mbps.
- RS-232 interface — often used in point-to-point short distance (within 15 meters) communication.
- RS-485 interface — often used for multi-points long distance communication. Since it provides with multi-points data exchange capability and long distance communication performance; therefore, now is widely used in industrial control field.

● Communication Parameters
To transfer data through communication interface, the settings such as “data bit length”, “parity”, “stop bit” and “baud rate” must be configured first. These settings are collectively referred to as communication parameters and can be regarded as communication protocols at the hardware level. The communication parameters must be consistent for all communication devices in a system.

● Communication Protocol
All the device which provided with communication function should have a communication protocol to support data transmitting with other equipment.
The communication protocol is a protocol at the software level. Through the same communication protocol, can reach the purpose of data exchange between different devices.
A communication protocol usually consists of starting character, station number, communication command, data content, end character, check code, and so on. Of course, each device will define the appropriate communication protocol according to its communication needs. Meanwhile, some equipment will follow the commonly used communication agreement on the market and the MODBUS is undoubtedly the most common one.

● Communication Principles
When two or more devices exchange data between each other, these devices must be linked together to form a communication circuit. Moreover, this communication circuit needs to follow the basic principles below to start working:

- Have consistent communication interface.
- Have consistent communication parameters.
- Have consistent communication protocol.
- The communication circuit must have a main leader, and only one at a time.

7-1-2 Notes for Constructing Communication Systems

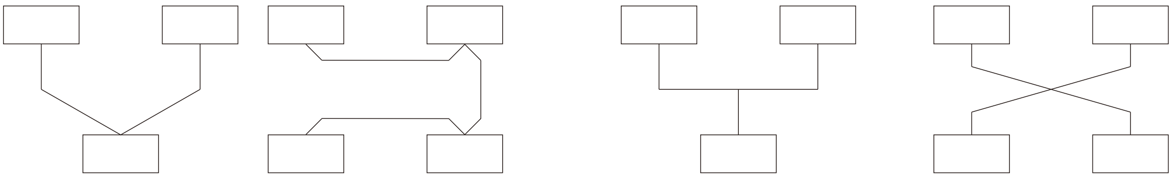
- Keep away from high noise sources as can as possible while wiring. In the distribution box, do not use the same groove with the power wire. For the external environment, keep away from the equipment which produces electromagnetic radiation.

- Pay attention to the communication distance and select the appropriate communication interface. As the specifications of the RS-485 interface are much better than the RS-232, should use the RS-485 interface as far as possible in industrial control systems. Even so, in the use of RS-485 interface, many things must be noted in accordance with its specifications.

- Guidelines for the use of RS-485

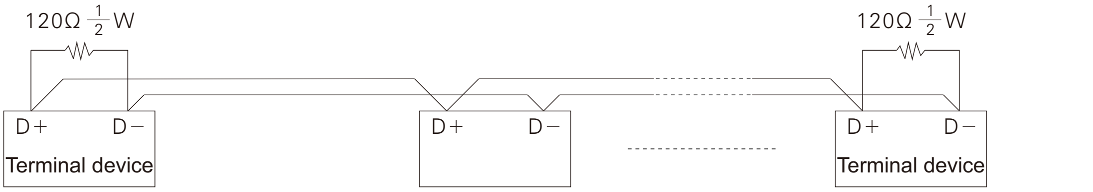
A. The transfer wire need to use shielded twisted pair wire. Normal twisted pair wire can be used when conducting short distance communication in low noise environment to cut down cost. But in high noise environment, long distance communication or in occasions where high communication quality is required, the dedicated transfer wire for RS-485 (like the Belden 9841) is recommended. It may make higher budget, but the communication quality will be improved magnificently.

B. Make sure the principle of connect in sequence is followed when do hardware wiring. Do not use the T-type, the radial distribution type or any other wiring method for the purpose of convenience.



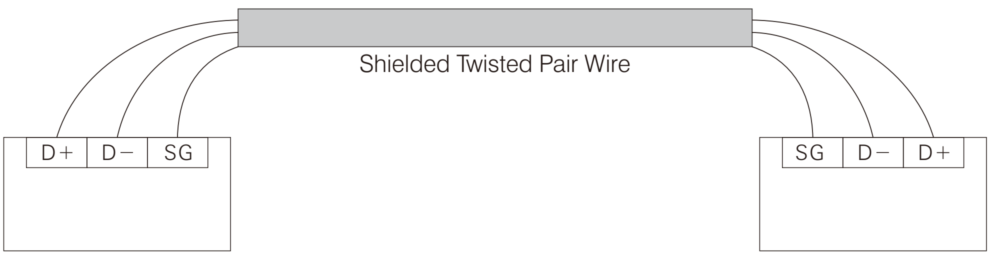
Connect in sequence wiring T-type wiring Radial distribution wiring

C. Terminal resistances must be parallel connected to the two terminal devices 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 resistors.



All of the VS series communication cards have built-in terminal resistances, user can be enabled the function by short connect the TR terminals. For those communication devices which have no built-in terminal resistances, take special note during wiring to ensure that the external terminal resistance is well connected.

D. 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 to achieve the purpose of normal communication.



E. When the number of serial connections at the RS-485 circuit has exceeded a certain amount (depends on the specification of the devices connected, usually 32), an RS-485 amplifier has to be added to the circuit.

F. According to the standard specifications of the 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 to connect with different devices at the same time, thus when the communication fails, carefully inspect whether all wirings are correct and stable, furthermore check the configuration values of each device are correct. On the other hand, usually a complicated structure is more difficult to find out the problem. Therefore, can separate a device to do individual checking before connecting it with many other devices, that for to make sure every device can work well.

- Misconception about communication speed.
The communication systems are built for various purposes and applications. Most people feel that faster communication means better. In fact, this concept is not necessarily correct. The reason is that faster communication speed relies on the higher communication quality to support, which also means that higher communication construction costs may be required.
Therefore, the correct approach is to, in response to the needs, choose the appropriate communication speed, consider moderate construction costs and pursue stable communication quality.

- For the situations when the constructed system can communicate acceptably, but sometimes breaks or error occurs. That results the unsmooth and delayed data transmission, there are several suggestions listed below:
  A. Check whether the communication software is working properly, including whether the communication parameters (such as the time-out period setting) are correct.

  B. Reduce and improve on the interference from the surrounding environment. Specific practices include reducing the carrier frequency of the inverter, setting appropriately the grounding system of inverters and a variety of power equipment. Besides, can even add noise suppress devices to the power wire.

  C. It is recommended to use RS-485 dedicated transmission cable if the general transmission line is currently used.

  D. Rewire the communication cables that following the principle of keeping away from sources of interference.

- Overall consideration for the communication quality:
When building a communication system, it is often necessary to connect a variety of devices with different brands and which is the difficult part. Additionally, the impact of environmental factors makes it more complicated.
When constructing communication systems, users often ask questions such as how long can this communication distance be, then how long will the communication distance be? Nothing is wrong to ask such questions, since only the excellent communication quality can ensure smooth communication. However, there is no standard answer.
Too many factors impact the communication quality of a project, such as the quality of equipment, wiring quality, the quality of the environment at the site, even software quality and so on.
The suggestions to improve the quality of communication are made as follows.
  A. Equipment quality:
  The isolated communication interface is definitely better than non-isolated communication interface. However, for many cheap communication interface converters on the market, even their specifications do not meet the standard. They may work in a simple testing, but the communication quality is poor, so should avoid using them.

  B. Wiring quality:
  Each communication interface has its standard transmission cable. The user often uses inappropriate cables in order to reduce costs. When the communication system begins to appear to be unstable, please replace the standard transmission cable to enhance the communication quality.

  C. The quality of the environment at the site:
  When a high noise source at the work site affects the communication, this situation should be improved by reducing noise interferences. If this cannot be changed, should consider changing the routes of wiring to keep away from the noise source.

  D. The quality of the software program:
  The communication system is generally used as monitor of equipments. There are monitoring computer and monitoring software at the monitoring center. There are possibilities for errors to occur during communication. During the communication, there may have the possibility of error. Therefore, the ability of monitoring software to handle errors is related to the stability of the entire system.

  At the hardware part, the communication quality should be improved as much as possible to avoid communication errors. When the occasional error occurs, the quality of the software can offer a correct treatment. Only by the two-pronged approach can build a stable and reliable communication system.

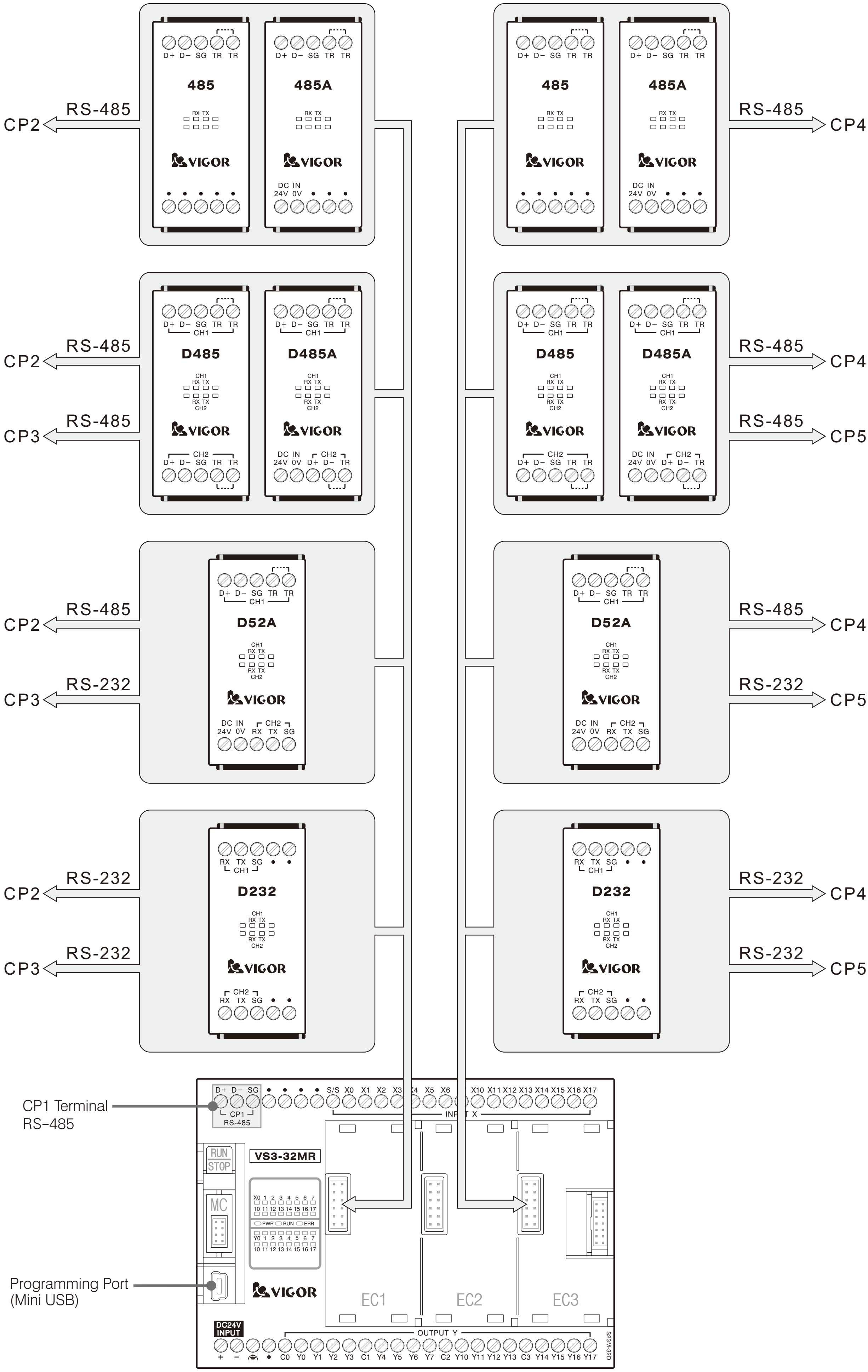
7-2 Structure of Communication System

Every VS PLC Main Unit is equipped with a built-in mini USB interface programming port for to connect and communicate with the programming device.
Moreover, each VS series PLC Main Unit is also equipped with the built-in CP1 which is a RS-485 multi-functional communication port. This CP1 can select the correct application type for fit with the external peripherals from a various of prepared modes, and then to achieve the control objectives cooperatively.
If more communication ports are needed, can be achieved by installing communication expansion cards to expand the number of communication ports. Moreover, all of the expanded communication ports are multi-functional and can be applied to a variety of purposes.

The VS series PLC Main Unit not only has a built-in CP1 communication port, but also the EC1 Expansion Card Socket is available to install a communication expansion card. Thus, by the EC1 socket, the PLC could have the additional CP2 and CP3. Moreover, the VS3 can use its EC1 and EC3 to expand CP2~CP5. However, if the CP5 at a VS3 PLC is required, its EC2 socket may install a VS-3AV-EC card or not to use; any DIO or SF Card at EC2 will cause the CP5 at EC3 ineffective.

A communication port at CP card is not related to the working area of DIO expansion or SF card, to operate the port is directly by the installed setting and program.

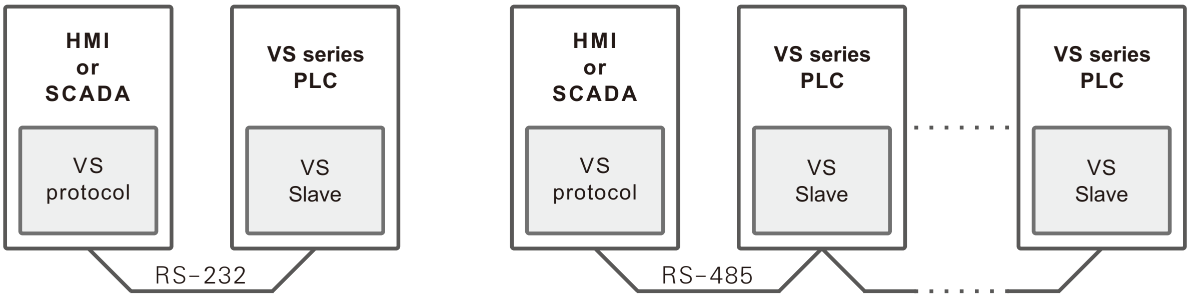
Item	Model Name	Main Specification
Comm. Expansion Card	VS-485-EC	RS-485 Comm. Expansion Card: One non-isolated RS-485 port with TX / RX indicators; dist. 50m Max.
	VS-485A-EC	RS-485 Comm. Expansion Card: One isolated RS-485 port with TX / RX indicators; dist. 1000m Max.
	VS-D485-EC	RS-485 Comm. Expansion Card: Dual non-isolated RS-485 ports with TX / RX indicators; dist. 50m Max.
	VS-D485A-EC	RS-485 Comm. Expansion Card: Dual isolated RS-485 ports with TX / RX indicators; dist. 1000m Max.
	VS-D232-EC	RS-232C Comm. Expansion Card: Dual non-isolated RS-232 ports with TX / RX indicators; dist. 15m Max.; wiring by the RX / TX / SG terminals
	VS-D52A-EC	RS-485 + RS-232C Comm. Expansion Card: One isolated RS-485 port (1000m) & one non-isolated RS-232C port (15m), both with TX / RX indicators and wiring by terminals
	VS-ENET-EC	Ethernet + RS-485 Communication Expansion Card: One Ethernet port (with additional non-isolated RS-485, dist. 50m) & one non-isolated RS-485 port (dist. 50m), both with TX / RX indicators



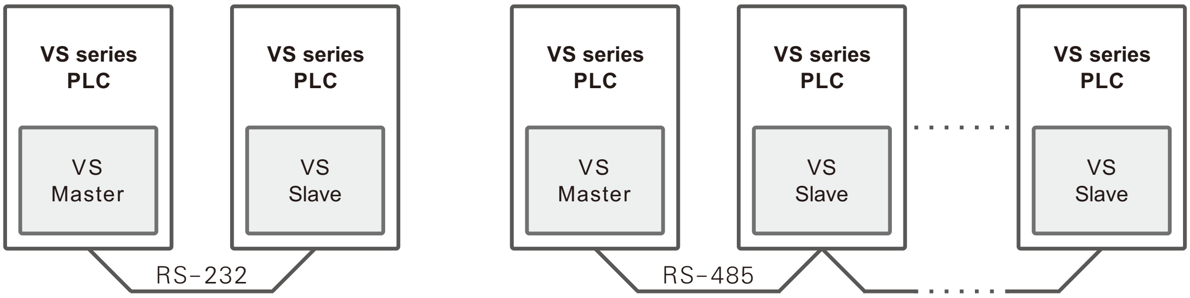
485
485A
D485
D485A
D52A
D232
VIGOR
RS-485
RS-232
CP2
CP3
CP4
CP5
D+ D− SG TR TR
CH1
CH2
DC IN 24V 0V
RX TX SG
CP1 Terminal
RS-485
Programming Port
(Mini USB)
VS3-32MR
RUN
STOP
MC
INPUT X
OUTPUT Y
DC24V INPUT
EC1
EC2
EC3
S/S X0 X1 X2 X3 X4 X5 X6 X10 X11 X12 X13 X14 X15 X16 X17
C0 Y0 Y1 Y2 Y3 C1 Y4 Y5 Y6 Y7 C2 Y10 Y11 Y12 Y13 C3 Y14 Y15 Y16 Y17
PWR RUN ERR

The communication application types supported by the VS series PLC are briefly introduced below:

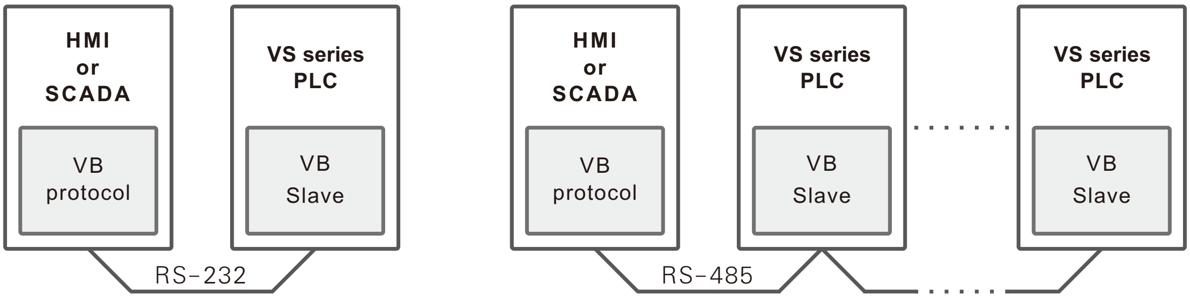
- The VS Computer Link Slave (hereby referred to as the VS Slave)
  When the VS PLC’s communication port has been set to the application type of the “VS Computer Link Slave”, the human-machine interface (HMI) or supervisory control and data acquisition system (SCADA) can access the VS PLC’s data through the “VS Computer Link protocol” (hereby referred to as the VS Protocol).



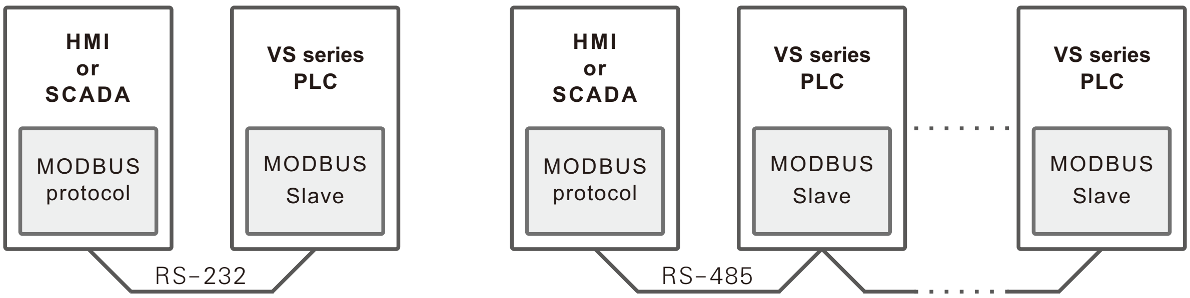
- The VS Computer Link Master (hereby referred to as the VS Master)
  When the VS PLC’s communication port has been set to the application type of the “VS Computer Link Master”, it cooperates with the LINK instruction and a LINK communication table to operate communication procedures. This Master links and communicates with Slaves through the “VS Computer Link protocol” (hereby referred to as the VS Protocol).



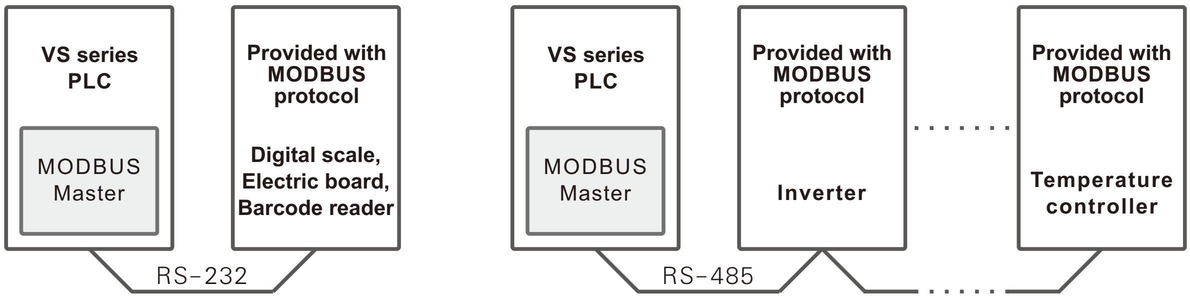
- The VB Computer Link Slave (hereby referred to as the VB Slave)
  When the VS PLC’s communication port has been set to the application type of the “VB Computer Link Slave”, the human-machine interface (HMI) or supervisory control and data acquisition system (SCADA) can access the VS PLC’s data through the “VB Computer Link protocol” (hereby referred to as the VB protocol).



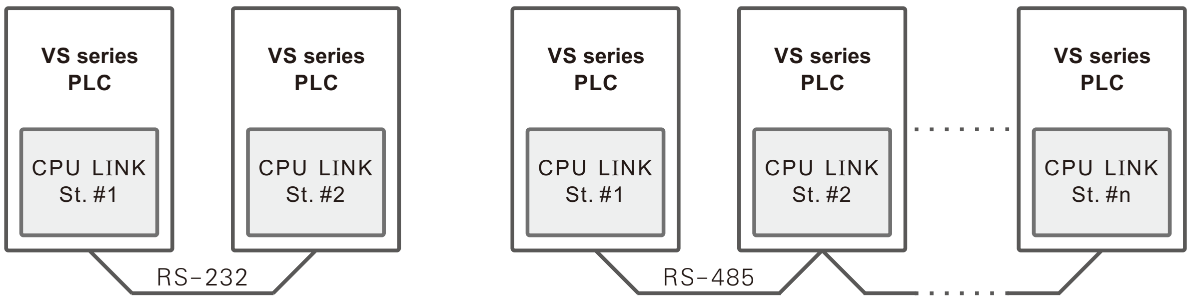
- The MODBUS Slave
  When the VS PLC’s communication port has been set to the application type of the “MODBUS Slave”, the human-machine interface (HMI) or supervisory control and data acquisition system (SCADA) can access the VS PLC’s data through the “MODBUS protocol”.



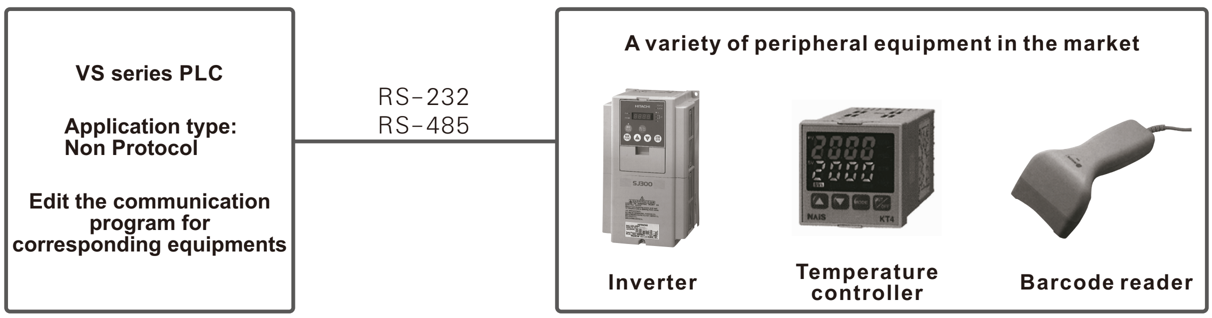
- The MODBUS Master
  When the VS PLC’s communication port has been set to the application type of the “MODBUS Master”, it cooperates with the MBUS instruction and a MODBUS communication table to operate communication procedures. This Master links and communicates with MODBUS Slaves (such as inverters, temperature controllers, power meters, etc.) through the “MODBUS protocol”.



- The CPU Link
  The VS series PLC uses this application to share instant data between VS PLCs to serve the purpose of distributed control.
  When the communication ports of VS PLCs have been set the application for to share instant data between PLCs, select one of the PLC to cooperate with the CPUL instruction and CPUL communication table, through its particular protocol to share instant data.



- The Non Protocol communication
  When the VS PLC’s communication port has been set to the application type of the “Non Protocol”, the PLC does not execute any specified communication protocol. All communication procedures are operated by the PLC's user program. Then the RS instruction is used to receive and send out data in order to complete the communication operation. This application type is often applied to communicate between PLC and peripheral, such as temperature controllers, inverters and barcode readers.

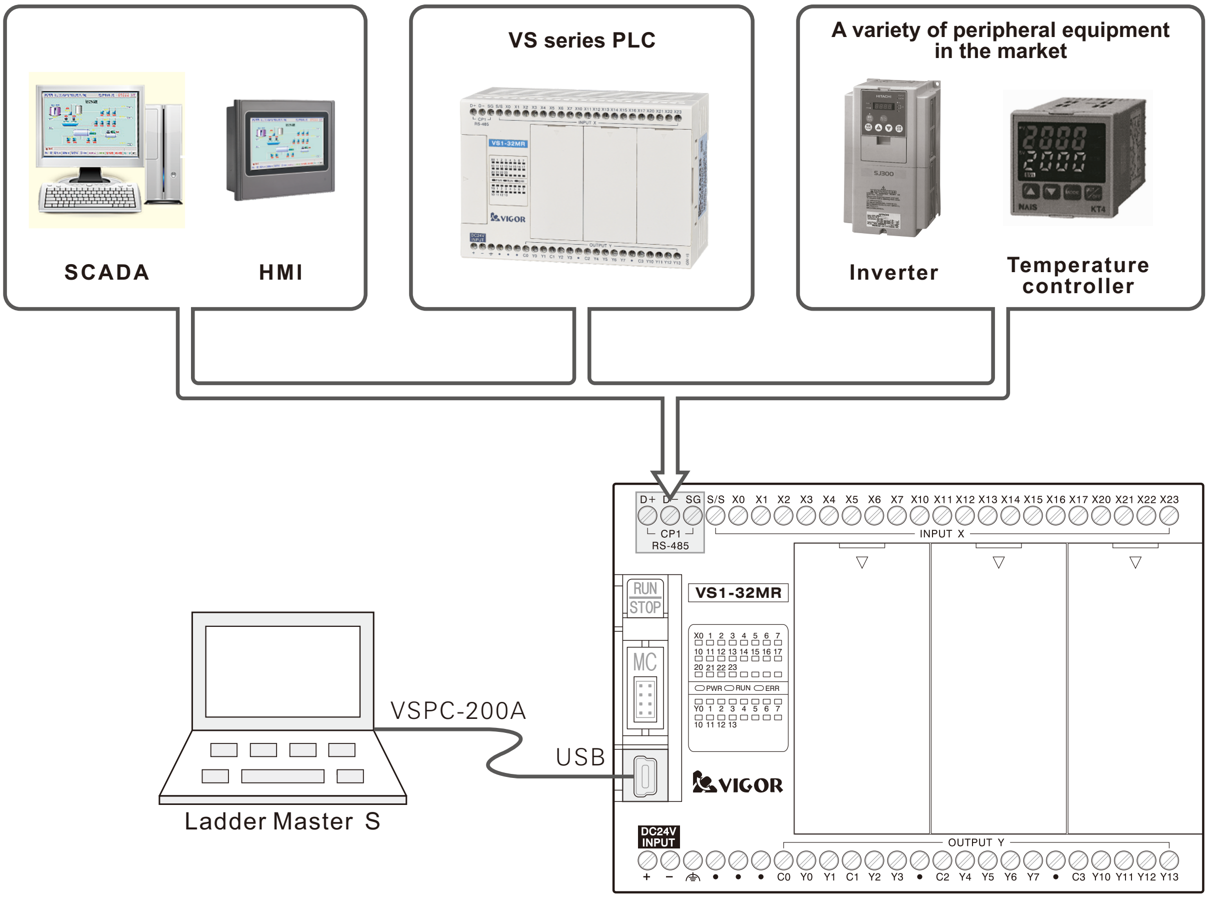


7-2-1 Main Unit Built-in Communication Port

Every VS PLC Main Unit is equipped with a built-in mini USB port, which is the programming interface with 12 Mbps transmission rate and uses the “VS Computer Link Protocol”. User can follow the protocol to make a communications procedure for to access the data in the VS PLC.

VIGOR provides the VSPC-200A noise suppression USB cable that is designed for industrial environment use, could reduce the USB communication failed. Therefore, use a computer to do the programming can acquire stable and rapid communication quality.

Furthermore, every VS PLC Main Unit is equipped with a built-in CP1 communication port, which is a non-isolated RS-485 interface. Through the installed project (which includes the port’s parameter setting and appropriate program), this communication port can choose one application from a variety of communication functions.



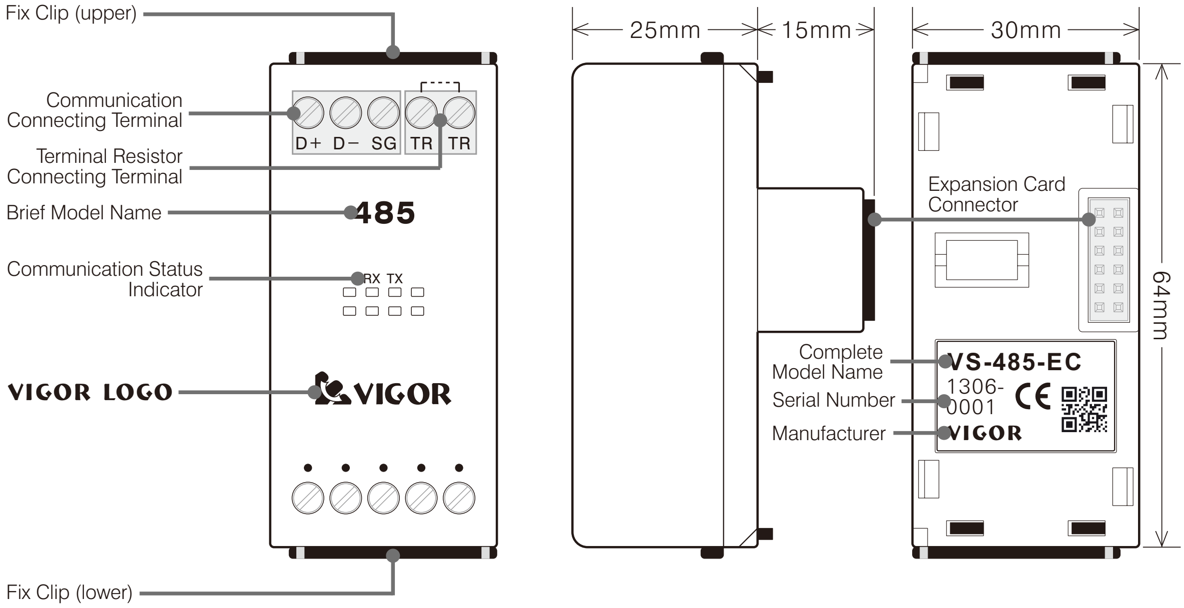
- The specification of CP1

Item	Specification
Communication Interface	RS−485
Isolation Method	No isolation
Max. Communication Distance	50 M
Communication Method	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)

7-2-2 VS-485-EC Communication Expansion Card

The VS-485-EC Communication Expansion Card offers one set of non-isolated RS-485 port. By way of the parameter setting and well planned program at the VS series PLC Main Unit, the port can perform one of various communication functions.

- Product Exterior



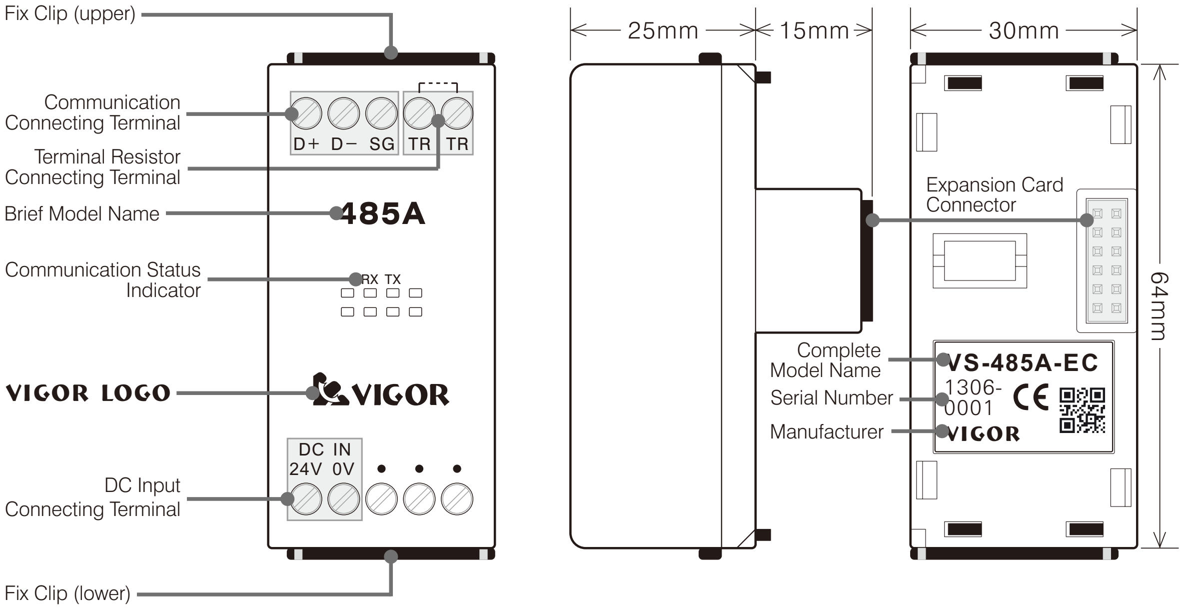
- Product Specification

Item	Specification
Communication Interface	RS–485
Isolation Method	No isolation
Communication Indicator	TX (transmitting) and RX (receiving)
Max. Communication Distance	50 M
Communication Method	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)
Power Consumption	DC5V 50mA (from PLC Main Unit)
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block
Terminal Resistor	120Ω, enabled when two TR terminals are short-connected
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)

7-2-3 VS-485A-EC Communication Expansion Card

The VS-485A-EC Communication Expansion Card offers one set of isolated RS-485 port. By way of the parameter setting and well planned program at the VS series PLC Main Unit, the port can perform one of various communication functions.

- Product Exterior



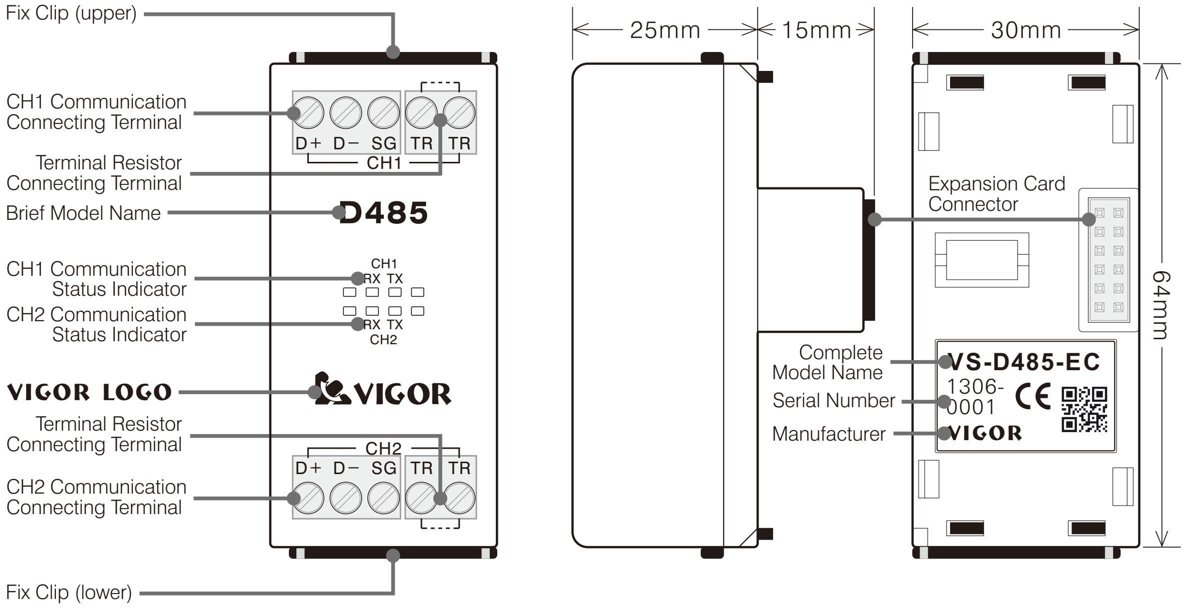
- Product Specification

Item	Specification
Communication Interface	RS-485
Isolation Method	PLC internal circuit and external driving circuit are isolated by a magnetic coupler
Communication Indicator	TX (transmitting) and RX (receiving)
Max. Communication Distance	1000 M
Communication Method	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)
Power Consumption	DC24V 25mA from the DC input terminal
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block
Terminal Resistor	120Ω, enabled when two TR terminals are short-connected
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)

7-2-4 VS-D485-EC Communication Expansion Card

The VS-D485-EC Communication Expansion Card offers two sets of non-isolated RS-485 port. By way of the parameter setting and well planned program at the VS series PLC Main Unit, each port can separately perform one of various communication functions.

- Product Exterior



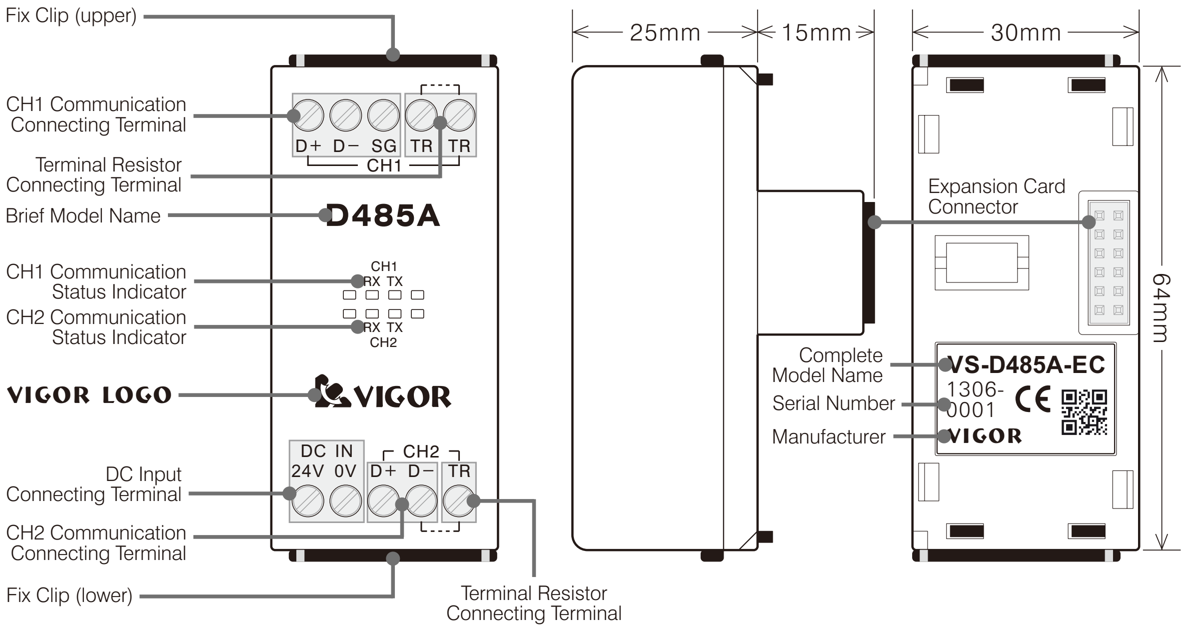
- Product Specification

Item	Specification	
	CH1	CH2
Communication Interface	RS−485	RS−485
Isolation Method	No isolation	No isolation
Communication Indicator	TX (transmitting) and RX (receiving)	TX (transmitting) and RX (receiving)
Max. Communication Distance	50 M	50 M
Communication Method	Half-duplex	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)	
Power Consumption	DC5V 100mA (from PLC Main Unit)	
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block	Fixed 5mm Screw-Clamp Terminal Block
Terminal Resistor	120Ω, enabled when two TR terminals are short-connected	
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)	

7-2-5 VS-D485A-EC Communication Expansion Card

The VS-D485A-EC Communication Expansion Card offers two sets of isolated RS-485 port. By way of the parameter setting and well planned program at the VS series PLC Main Unit, each port can separately perform one of various communication functions.

- Product Exterior



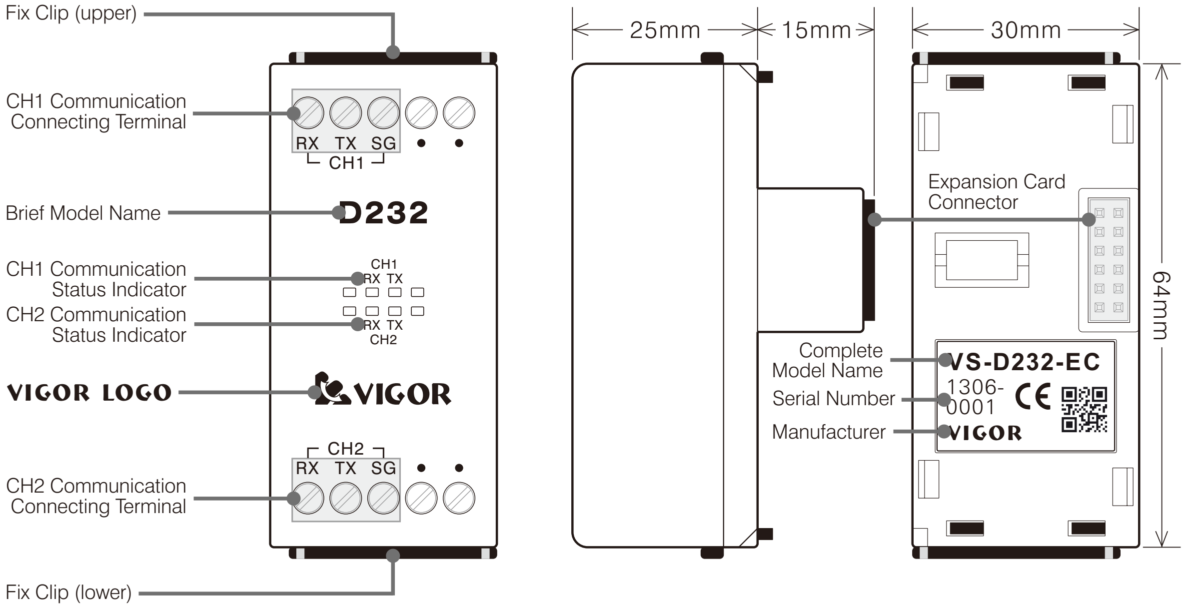
- Product Specification

Item	Specification	
	CH1	CH2
Communication Interface	RS−485	RS−485
Isolation Method	PLC internal circuit and external driving circuit are isolated by a magnetic coupler; no isolation between two channels	
Communication Indicator	TX (transmitting) and RX (receiving)	TX (transmitting) and RX (receiving)
Max. Communication Distance	1000 M	1000 M
Communication Method	Half-duplex	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)	
Power Consumption	DC 24 V 50m A from the DC input terminal	
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block	Fixed 5mm Screw-Clamp Terminal Block
Terminal Resistor	120Ω, enabled when two TR terminals are short-connected	120Ω, enabled when the D- and TR terminals are short-connected
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)	

7-2-6 VS-D232-EC Communication Expansion Card

The VS-D232-EC Communication Expansion Card offers two sets of non-isolated RS-232C port. By way of the parameter setting and well planned program at the VS series PLC Main Unit, each port can separately perform one of various communication functions.

- Product Exterior



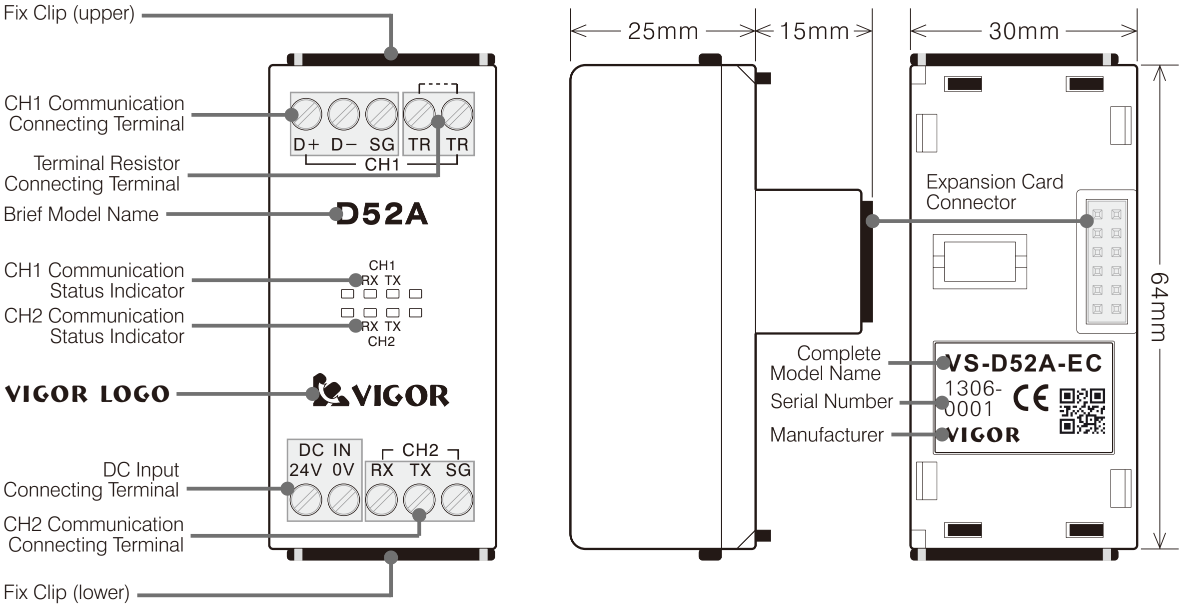
- Product Specification

Item	Specification	
	CH1	CH2
Communication Interface	RS−232C	RS−232C
Isolation Method	No isolation	No isolation
Communication Indicator	TX (transmitting) and RX (receiving)	TX (transmitting) and RX (receiving)
Max. Communication	15 M	15 M
Communication Method	Half-duplex	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)	
Power Consumption	DC5V 25mA (from PLC Main Unit)	
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block	Fixed 5mm Screw-Clamp Terminal Block
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)	

7-2-7 VS-D52A-EC Communication Expansion Card

The VS-D52A-EC Communication Expansion Card offers one isolated RS-485 port and one non-isolated RS-232C port. By way of the parameter setting and well planned program at the VS series PLC Main Unit, each port can separately perform one of various communication functions.

- Product Exterior



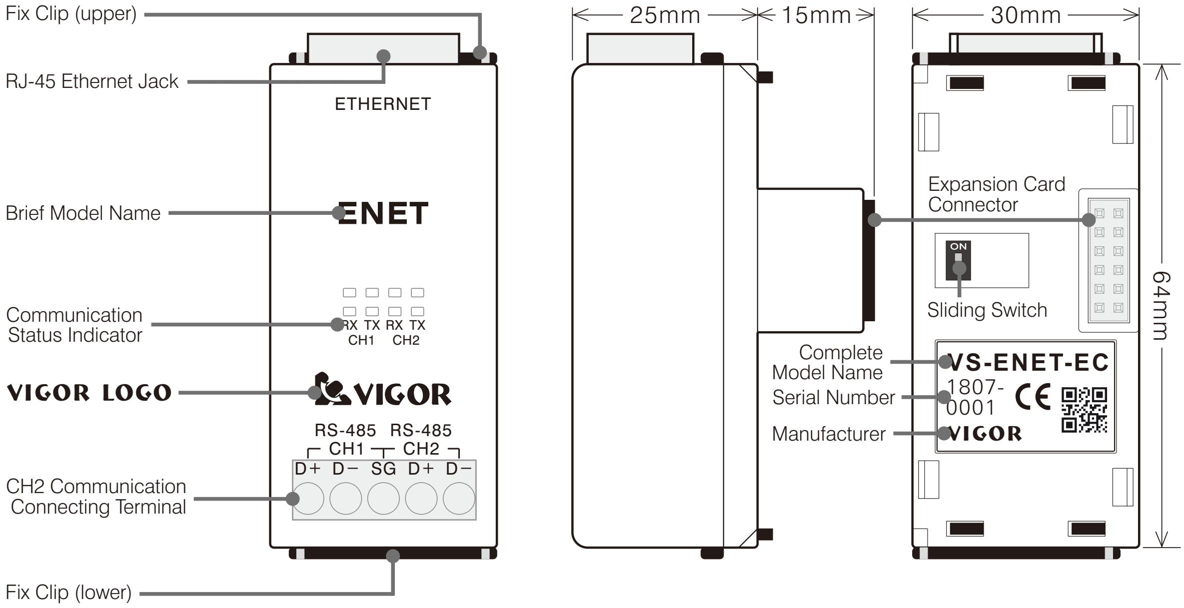
- Product Specification

Item	Specification	
	CH1	CH2
Communication Interface	RS-485	RS-232C
Isolation Method	PLC internal circuit and external driving circuit are isolated by a magnetic coupler	No isolation
Communication Indicator	TX (transmitting) and RX (receiving)	TX (transmitting) and RX (receiving)
Max. Communication Distance	1000 M	15 M
Communication Method	Half-duplex	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)	
Power Consumption	DC24V 50mA (from the DC input terminal) + DC5V 25mA (from PLC Main Unit)	
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block	Fixed 5mm Screw-Clamp Terminal Block
Terminal Resistor	120Ω, enabled when two TR terminals are short-connected	—
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)	

7-2-8 VS-ENET-EC Communication Expansion Card

The VS-ENET-EC Communication Expansion Card is a dual port card. It offers one Ethernet port (with an additional non-isolated RS-485 interface) and one non-isolated RS-485 port.
By way of the parameter setting and well planned program at the VS series PLC Main Unit, each port can separately perform one of various communication functions.

● Product Exterior



● Product Specification

Item	Specification about the Ethernet
Physical Transport Layer	10BASE-T / 100BASE-TX
Transport Protocol	TCP (Client / Server); UDP (Client / Server)
Application Protocol	Transparent or conversion MODBUS TCP to MODBUS RTU
Connected Serial Port	CH1
IP Address Allocation	Automatic by DHCP server or Static (manual) allocation
Unique Identifier	The media access control address (MAC address) is used
Comm. Parameter Modify	Use the specific configuration software
Status Indicator	LINK (green, ON when physical link is established); Data (yellow, blink when data is transferring)
Interface	Ethernet RJ-45 jack

Item	Specification about Serial Communication Ports	
	CH1*	CH2
Communication Interface	RS−485	RS−485
Isolation Method	No isolation	No isolation
Communication Indicator	TX (transmitting) and RX (receiving)	TX (transmitting) and RX (receiving)
Max. Communication	50 M	50 M
Communication Method	Half-duplex	Half-duplex
Baud Rate	By the setting of installed project (115,200 bps. Max.)	
Power Consumption	Ethernet DC5V 100mA + RS-485 ports DC5V 100mA (total is 200mA from PLC Main Unit)	
Wiring Method	Fixed 5mm Screw-Clamp Terminal Block	Fixed 5mm Screw-Clamp Terminal Block
Parameter Configuration	By the installed project (via the "COM Port setting" page in the programming software)	

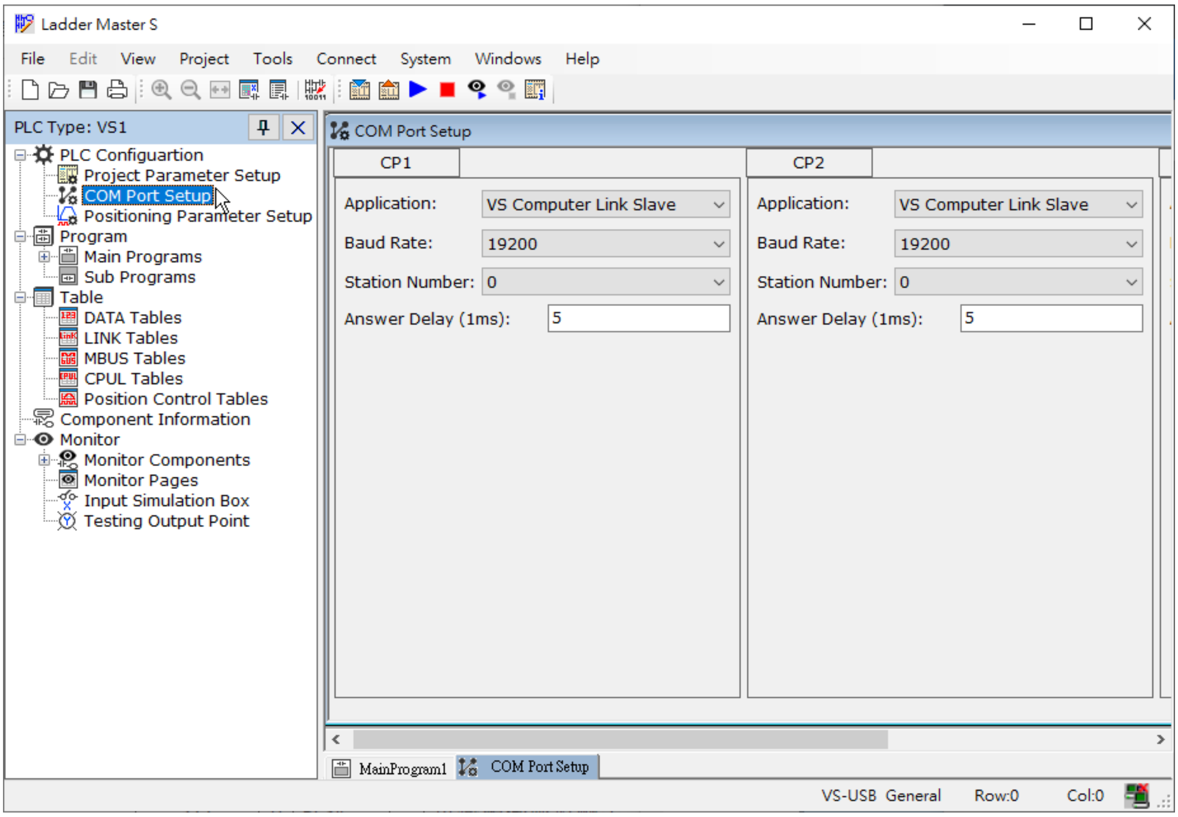
* For the CH1 of the VS-ENET-EC, there is a sliding switch under the card. If the communication Master is the CP2/CP4 at the PLC, please turn the switch ON. If the Master's request is from the Ethernet, turn the switch OFF.

7-3 Communication Types

The VS series PLCs have a complete selection of communication functions. They can be connected with a maximum of five communication ports CP1～CP5. Furthermore, each one is a multi-functional communication port and can be applied separately to a variety of communication application types. Each communication operating type will be introduced below.

7-3-1 Selections of Communication Types

Since the CP1～CP5 are multi-functional communication ports, the operation type of each port must be set precisely before use.
The operation types of communication ports are specified from the Ladder Master S programming software. The setting procedure is shown below:



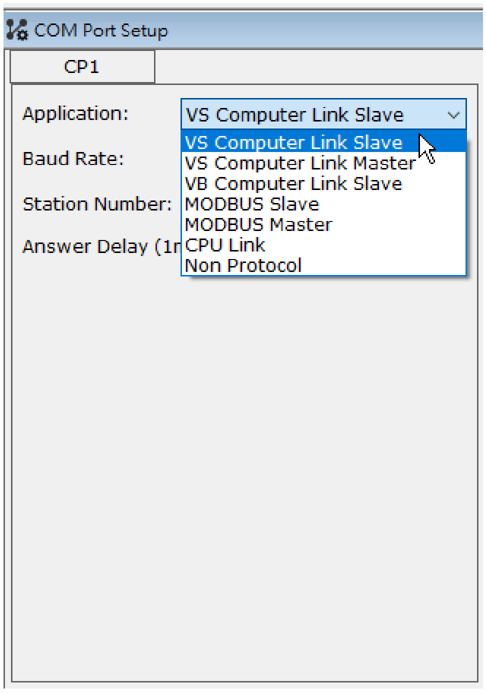
Select "COM Port Setup" from the Navigation Pane of the Project

A sheet as shown in the right diagram will appear, select an application type according to the request

There are corresponding items for each application type, which will guide user to set relevant parameters

After the settings of all communication ports are completed, click the ☒ on the upper corner at the setup window to finish and close it.

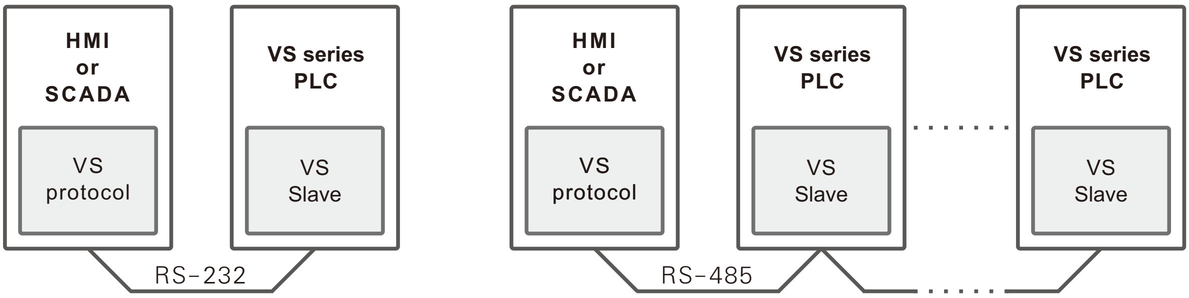
After the setup is completed, should run the "Write Project to PLC" function. Then, the parametes take effect on the PLC.



7-3-2 VS Computer Link Slave

When the VS PLC's communication port has been set to the application type of the "VS Computer Link Slave", the human-machine interface (HMI) or supervisory control and data acquisition system (SCADA) can access the VS PLC's data through the "VS Compurter Link Protocol" (hereby referred to as VS Protocol).

A computer or HMI through the "VS Protocol" could communicate with the VS PLCs. The HMI or SCADA supplier base on the "VS Protocol" to compile its driving programs, by way of that communicates with VS PLC's to construct a monitoring area network.

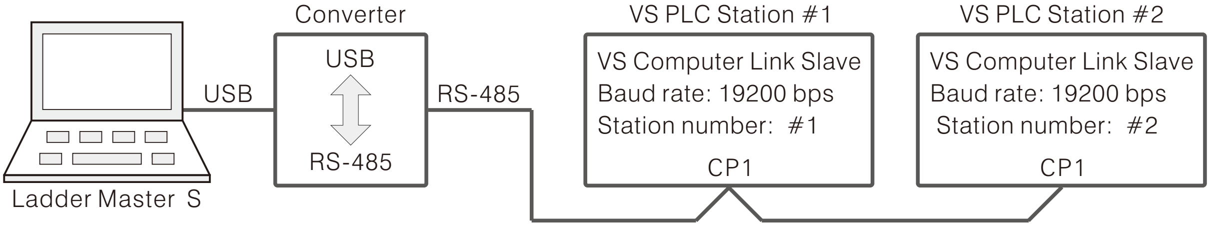


Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	VS Compurter Link Protocol (hereby referred to as VS Protocol)	
Commection Method	Half-duplex	
Communication Paramete	Bits-per-character: 8 bits; parity check: None; stop bit: 1 bit	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
No. of Slave Stations	1 Slave station only	Max. 255 stations (Must insert the RS-485 repeater(s) if the RS-485 wiring with more than 32 devices.)
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	CP1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the X, Y, M, S, T, C, D, R are included	

- The VS PLC will respond to the communication requirement by any device (such as a computer, HMI,...) if that follows the particular "VS Compurter Link Protocol" to give correct command. For the detail of the VS Protocol, please refer to the section "7-4 VS Series PLC Communication Protocol" for details.
- Usually, HMI or SCADA supplier base on the "VS Compurter Link Protocol" to compile driving program. Therefore, the HMI or SCADA user just needs to select the correct communication type when planning the system, so that can communicate with VS PLCs to construct a monitoring area network.
- Some software or HMI may not be equipped with the driving program of the VS series PLC. In this case, the "MODBUD Slave" application type can be used to connect them. Please refer to the section "7-3-5 MODBUD Slave" for details.

- Application Example

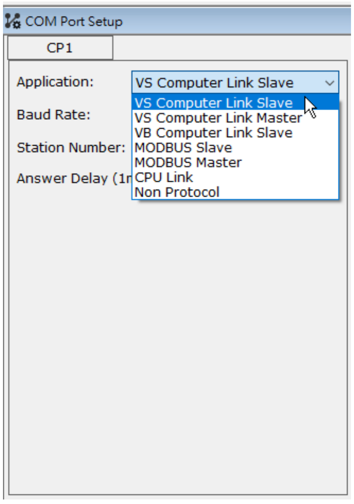
In this example, a computer uses its USB port to connect with an USB to RS-485 converter to transfer the communication signal to the RS-485 interface. Then, the RS-485 is connected to two VS series PLC's CP1 those station numbers are assigned to #1 and #2. After that, executes the Ladder Master S programming software on the computer to connect with the station #1 and station #2 PLCs for to read the project, control STOP/RUN and monitor those PLCs.



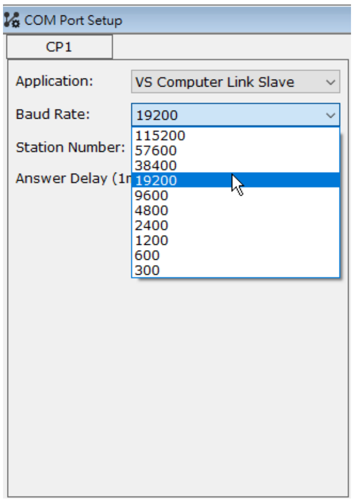
Please follow the procedures below to operate the test:

① Edit the project of PLC Station #1

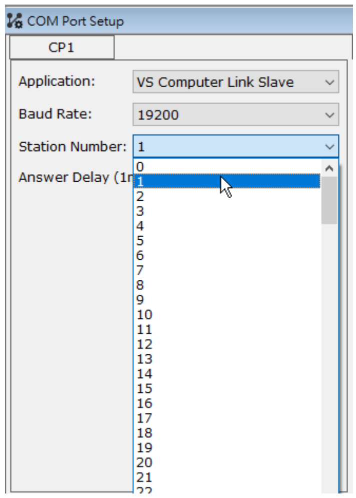
Use the Ladder Master S to set the CP1's parameters of PLC Station #1 and compile relevant program. Then, connect to the USB programming port of Station #1 and write the project into the PLC Station #1.



Set the application type:
VS Computer Link Slave



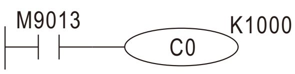
Set the baud rate:
19200 bps



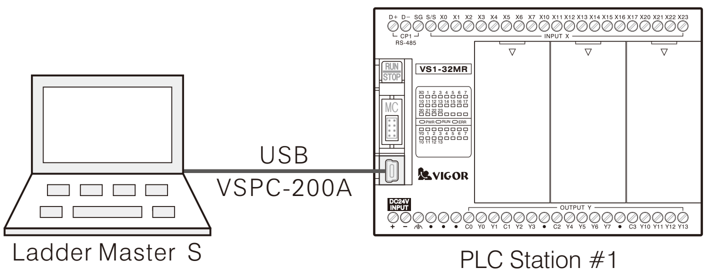
Set the station number:
#1

※ Usually the default delay response time is not necessary to change, except that:
When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

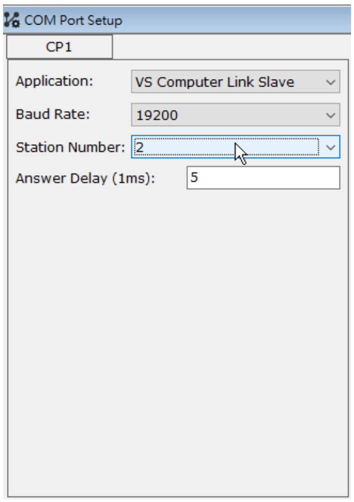
The program writes to the PLC Station #1.



By the USB programming port, the project is written into the PLC Station #1

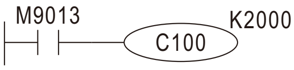


② Edit the project of PLC Station #2
Use the Ladder Master S to set the CP1's parameters of PLC Station #2 and compile relevant program. Then, connect to the USB programming port of Station #2 and write the project into the PLC Station #2.

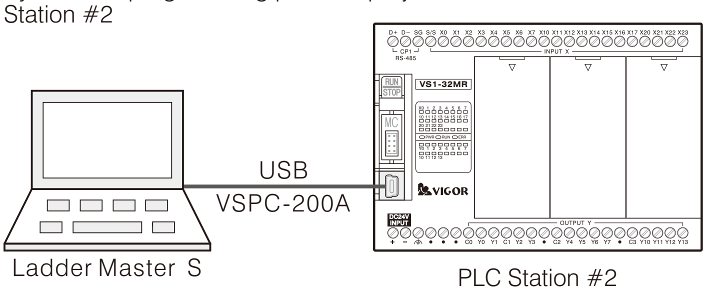


Set the station number:
#2

The program writes to the PLC Station #2.



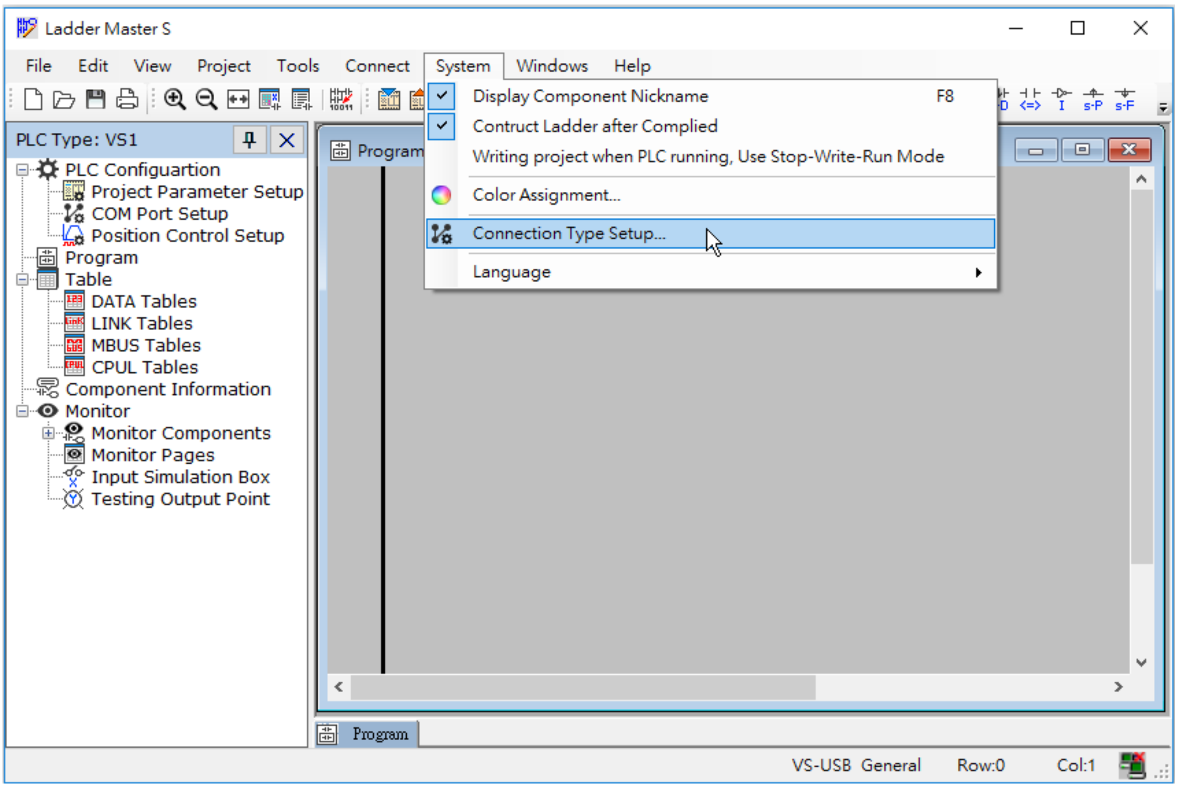
By the USB programming port, the project is written into the PLC Station #2



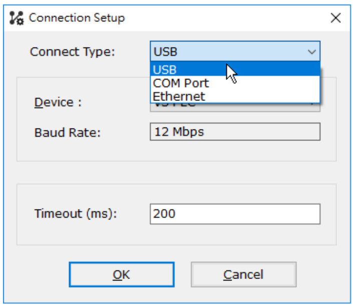
③ After complete to edit and load the projects to those two PLCs sequentially, do the wiring jobs between the computer, RS-485 converter and two PLCs' CP1.

④ Usually the application of the SCADA software is to monitor equipments through the communication network. In this example, the Ladder Master S replaces the SCADA software to execute the monitoring work. However, the Ladder Master S can only be connected to a single PLC at a time. Therefore, it is necessary to link and test with different PLC stations one by one.
In the real applications, when a communication system is finished the wiring but which cannot function normally, may use the Ladder Master S to test each station one by one to ensure the correctness of communication network.

To set the connection method of the Ladder Master S:

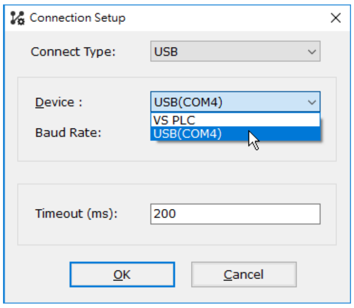


To set the connection method



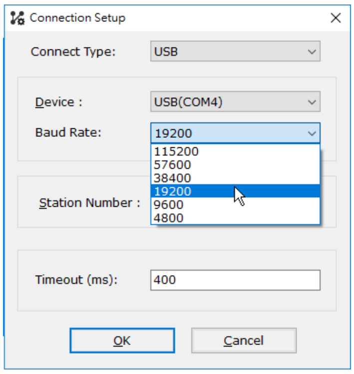
This example is using the USB to RS-485 converter, thus the “USB” is selected. For using the computer’s own communication port, select the “COM Port”.

To set the connection device



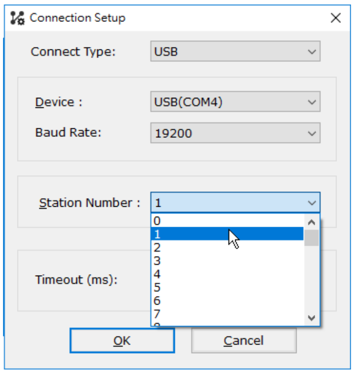
In this example, the given port is COM4 after the USB to RS-485 converter is applied.

To set the baud rate



Set the baud rate as same as all PLCs’ in the network circuit, that is 19200 bps in this example.

⑤ To test the connected PLC Station #1.



Set the connecting station number to the #1.

To read the PLC’s project → On the screen, it shows the program of the PLC Station #1 as written in the procedure ①.

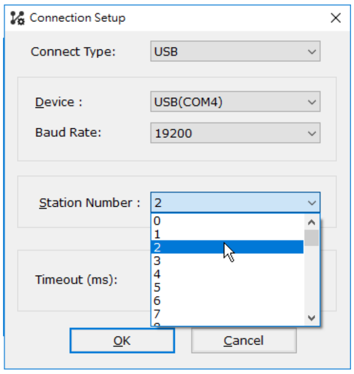
↓

To monitor the PLC → The ladder diagram on the screen by the monitor mode to display status.

↓

To let the PLC RUN → The Present Value of C0 in the program increases with a frequency of 1 per second.

⑥ To test the connected PLC Station #2.



Set the connecting station number to the #2.

To read the PLC’s project → On the screen, it shows the program of the PLC Station #2 as written in the procedure ②.

↓

To monitor the PLC → The ladder diagram on the screen by the monitor mode to display status.

↓

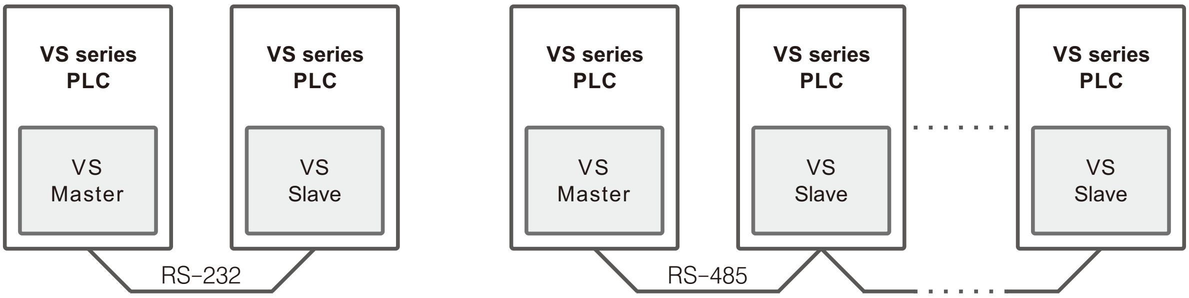
To let the PLC RUN → The Present Value of C100 in the program increases with a frequency of 1 per second.

7-3-3 VS Computer Link Master

When the VS PLC's communication port has been set to the application type of the "VS Computer Link Master", it should cooperate with the LINK instruction and a LINK communication table to operate communication procedures. This Master PLC links and communicates with Slaves through the "VS Computer Link protocol" (hereby referred to as the VS Protocol).

It is available to connect with several VS PLCs by the RS-485 interface.
Select one of these linked PLCs and set its application type as the "VS Computer Link Master" (this application type). This PLC is referred to as the Master station and the other PLCs are the Slaves. Therefore, all the Slave PLCs should set the application type as the "VS Computer Link Slave". The only one Master station of the connected PLCs should cooperate with the LINK instruction (FNC 89) and a LINK communication table to designate its data transferring between the Slave and Master.
Through this, the Master can sequential read and store data for all the Slave PLC stations on the net, therefore reaches the purpose of data transmission.

This application type is suitable for sharing a large amount of data between PLCs.



Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	VS Compurter Link Protocol (hereby referred to as VS Protocol)	
Commection Method	Half-duplex	
Communication Paramete	Bits-per-character: 8 bits; parity check: None; stop bit: 1 bit	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
No. of Slave Stations	1 Slave station only	Max. 256 stations (Must insert the RS-485 repeater(s) if the RS-485 wiring with more than 32 devices.)
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	CP1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the X, Y, M, S, T, C, D, R are included	

FNC 89	LINK S1 S2 n	Easy Link Communication Instruction	1	2	M	3
			○	○	○	○

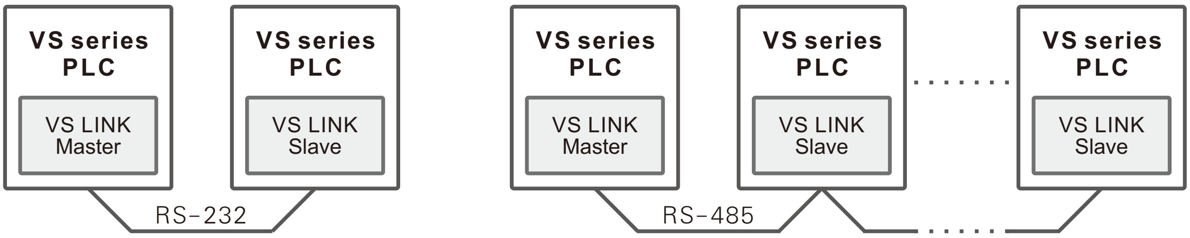
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
n																○		
• For the VS1, VS2 or VSM series, n = 1~3; for the VS3 series, n = 1~5																		

S1 : the communication table that describes the data receiving and transmitting

S2 : the working area for the instruction that occupies 4 registers

n : to assign the communication port, 1~5 = CP1~CP5

- The VS series PLC uses this instruction to transmit or get the designated data via its Communication Port CP1~CP5 with other VS PLCs.
- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the "Application type:" become the "VS Computer Link Master". Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the "Project" -- "COM Port Setting".
- This instruction could be used to transfer for various devices that is including all the Y, M, S, T, C, D and R (except for special M and special D).
- As shown in the figure below, select one of these linked PLCs as the Master station and the rest as the Slave stations. First, use the Ladder Master S to orderly open every Slave PLC's project and set its "Application type:" as the "VS Computer Link Slave" then to set an unique station No. (ranging from 0~254). After that, connect to the programming port of every PLC to load the project to each Slave PLC.
  Next, write this instruction in the Master PLC's program and set its "Application type:" as the "VS Computer Link Master", then edit the LINK communication table to designate data transferring actions. After that, to load this project to the Master PLC. Following these procedures, the purpose of data delivery between the PLCs can be reached.



- When X20 = "ON", the LINK instruction starts execution. According to the contents described in the "CP1_DATA_TRANS" communication table, it proceeds data write or read actions to the appointed Slave PLC stations, while the D100~D103 will be occupied to store the state of instruction execution.
- When the contents of the communication table specified by the S1 are execution completed from beginning to end, the M9104 will be "ON" for a scan time then the procedure will start over again from the first item set of the table.
- When the X20 turns from "ON" to "OFF", this instruction stops and the data sharing immediately stop but data which has transferred previously will still remain.
- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
■M9104	CP1 LINK / MBUS instruction on execution table complete once flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
■M9114	CP2 LINK / MBUS instruction on execution table complete once flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
■M9124	CP3 LINK / MBUS instruction on execution table complete once flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
■M9134	CP4 LINK / MBUS instruction on execution table complete once flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.
■M9144	CP5 LINK / MBUS instruction on execution table complete once flag.

• The communication table is assigned by the (S1) of the instruction.

An example of the “LINK” provided by programming tool the Ladder Master S is shown below.

Item No.	Command	Device at Master	Direction	Slave St. No.	Device at Slave	Length	Word / Bit	Disable Contact
1	Read	D0	<--	1	D0	10	—	M1
2	Write	D10	-->	1	D10	5	—	M1
3	Read	D20	<--	2	D0	10	—	M2
4	Write	D30	-->	2	D10	5	—	—
5	Write	D40	-->	255	D40	20	—	M0

The first item set in the table means that the Master reads the data D0~D9 from the Slave #1 and stores them at the D0~D9 of the Master.

The second item set in the table means that the Master writes its data at D10~D14 to the D10~D14 of the Slave #1.

The third item set in the table means that the Master reads the data D0~D9 from the Slave #2 and stores them at the D20~D29 of the Master.

The fourth item set in the table means that the Master writes its data at D30~D34 to the D10~D14 of the Slave #2.

The fifth item set in the table means that the Master writes its data at D40~D59 to all the D40~D59 of the Slaves. To set the Slave station #255 means that the Master writes the data to all of the Slaves. Note that Read Command cannot be used in this application.

The last column of the table is for the Disable Contact. If that specified contact is “ON”, the communication item set will be ignored. For example, if M1 = “ON”, the first and second item sets in the table will skip. This is the new function of the VS series, which can help designers effectively manage the operation of the communication table. A communication item set does not need to specify a Disabled Contact (such as the fourth item set), so that the command of the set does not have the disable control function.

• The working area of the executive instruction is starting from the (S2) (using D100~D103 as the example).

(S2)		Description
D100	Lower 8 bits	The record of the Slave's station number when the first communication error is occurred
	Upper 8 bits	Instruction working status 0: Normal data transmitting / receiving 4: The error is caused by the assigned device is inappropriate 7: The error is caused by the communication command is incorrect A: The communication setting is normal but no response from the Slave station (Time-out occurs) B: Abnormal communication
D101 ~ D103	The working area is required for the system when this instruction is performed	

The LINK instruction will reset the D100 to be 0 before the first item set at the table is executed.

If any communication error occurs during the execution of the instruction, a code will be recorded in D100. Only when the content value of D100 is 0, the recording action can be executed. Therefor, when there are possibilities of several errors, users can use the program to move out the content value of D100 then reset it to 0. This way allows the D100 to record the next error.

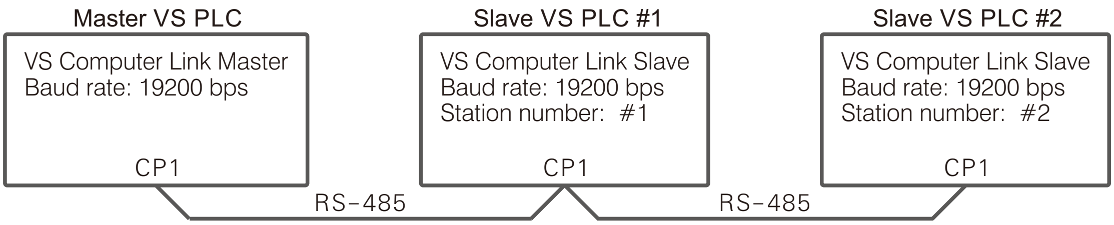
• To edit a communication table
Use the Ladder Master S to set up a LINK communication table and through its interactive window can set up and edit a communication table easily.
In the structure of VS series PLC, the communication tables are a part of the project. When the programmer to copy or access the project, those tables will be duplicated automatically with the program.

● Application Example

In this example, those CP1s of three VS PLCs are connected by the RS-485 interface and execute the VS Computer Link.

In the following figure, let the first one from the left be the Master station of VS Computer Link to execute the "VS Computer Link Master" application type and set the baud rate to be 19200 bps. Add the LINK instruction in the program and then edit the LINK communication table, thereby to process data transmitting from or to the VS Computer Link Slaves.

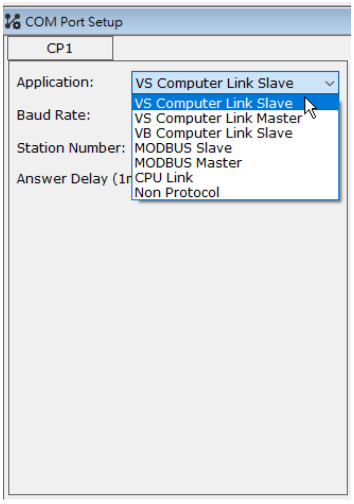
Another two are the VS Computer Link Slave PLCs, those are executing the "VS Computer Link Slave" application type. The baud rate of those connected ports should be the same as the port of the Master PLC. Then, individually set the station numbers of Slaves become the #1 and #2. Afterward, write the relevant control programs into the PLCs.



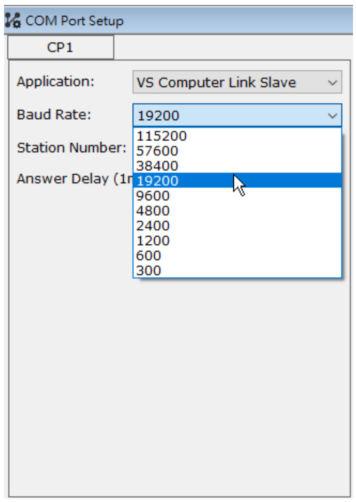
Please follow the procedures below to operate the test:

① Edit the project of Slave PLC #1

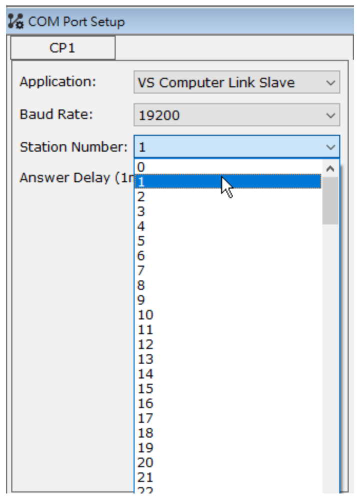
Use the Ladder Master S to set the CP1's parameters of Slave PLC #1 and compile relevant program. Then, connect to the USB programming port of Slave #1 and write the project into the PLC.



Set the application type:
VS Computer Link Slave



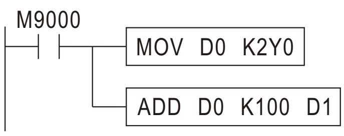
Set the baud rate:
19200 bps



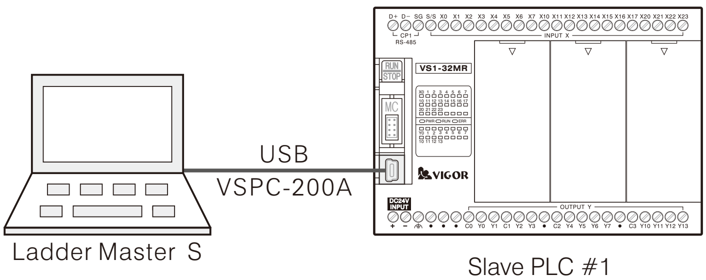
Set the station number:
#1

※ Usually the default delay response time is not necessary to change, except that:
When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

The program writes to the Slave PLC #1.

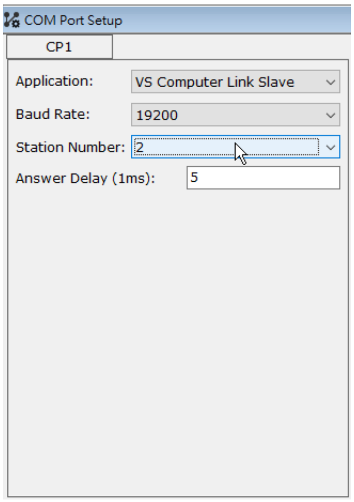


By the USB programming port, the project is written into the Slave PLC #1



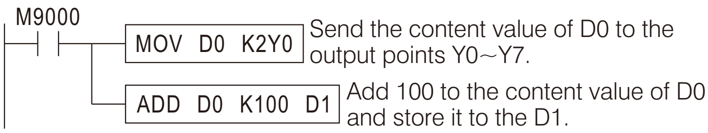
② Edit the project of Slave PLC #2

Use the Ladder Master S to set the CP1's parameters of Slave PLC #2 and compile relevant program. Then, connect to the USB programming port of Slave #2 and write the project into the PLC.

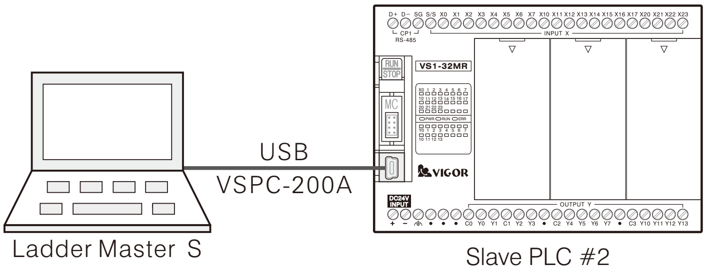


Set the station number:
#2

The program writes to the Slave PLC #2



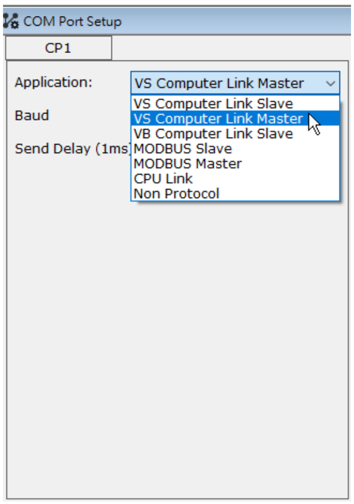
By the USB programming port, the project is written into the Slave PLC #2



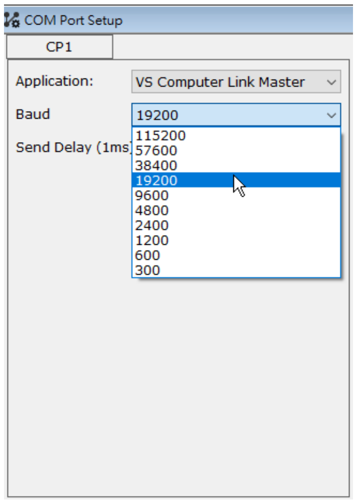
Slave PLC #2

③ Edit the project of the Master PLC

Use the Ladder Master S to set the CP1's parameters of Master PLC. Compile the LINK communication table, the LINK instruction and relevant program. Then, connect to the USB programming port of Master and write the project into the PLC.



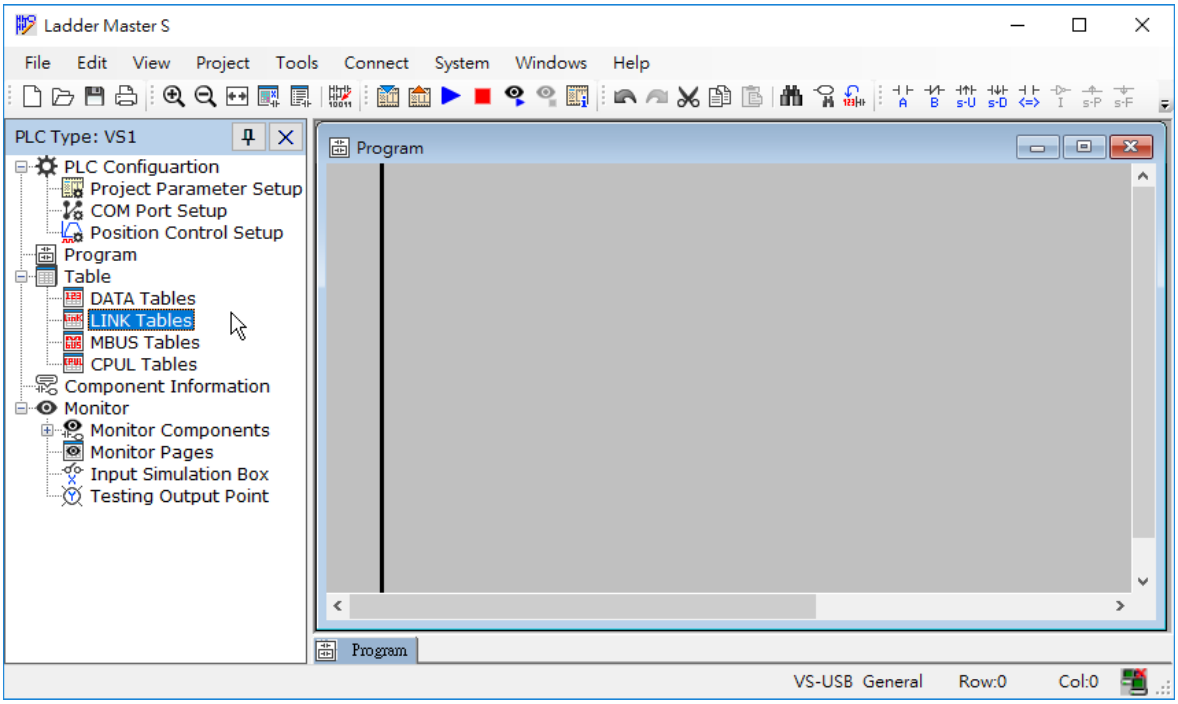
Set the application type:
VS Computer Link Master



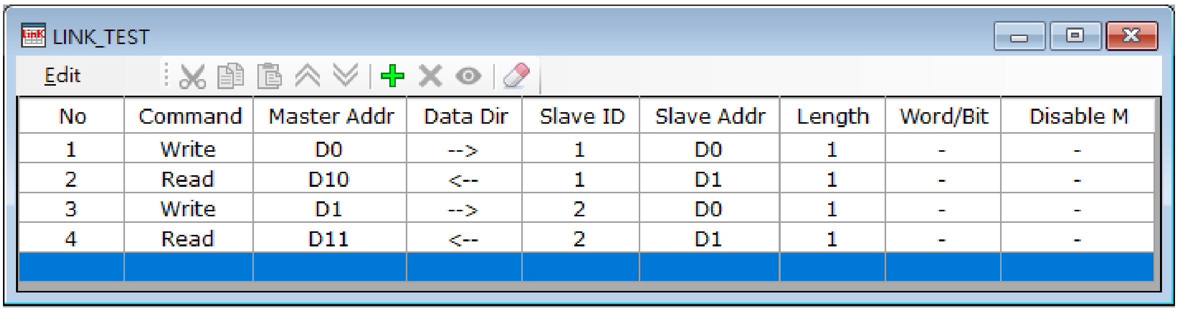
Set the baud rate:
19200 bps

※ Usually the default delay response time is not necessary to change, except that:
When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

Open a new LINK communication table at the Master PLC.



Name this LINK communication table as “LINK_TEST” first, and then compile its contents.

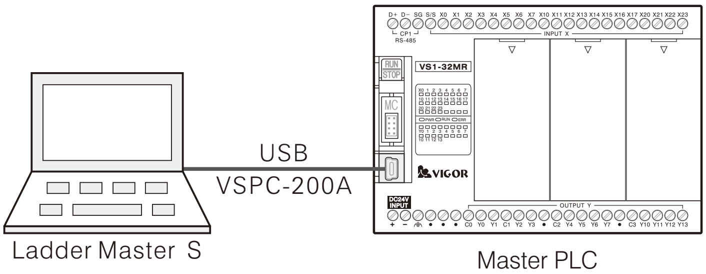


No	Command	Master Addr	Data Dir	Slave ID	Slave Addr	Length	Word/Bit	Disable M
1	Write	D0	-->	1	D0	1	-	-
2	Read	D10	<--	1	D1	1	-	-
3	Write	D1	-->	2	D0	1	-	-
4	Read	D11	<--	2	D1	1	-	-

Edit the program for the Master PLC.

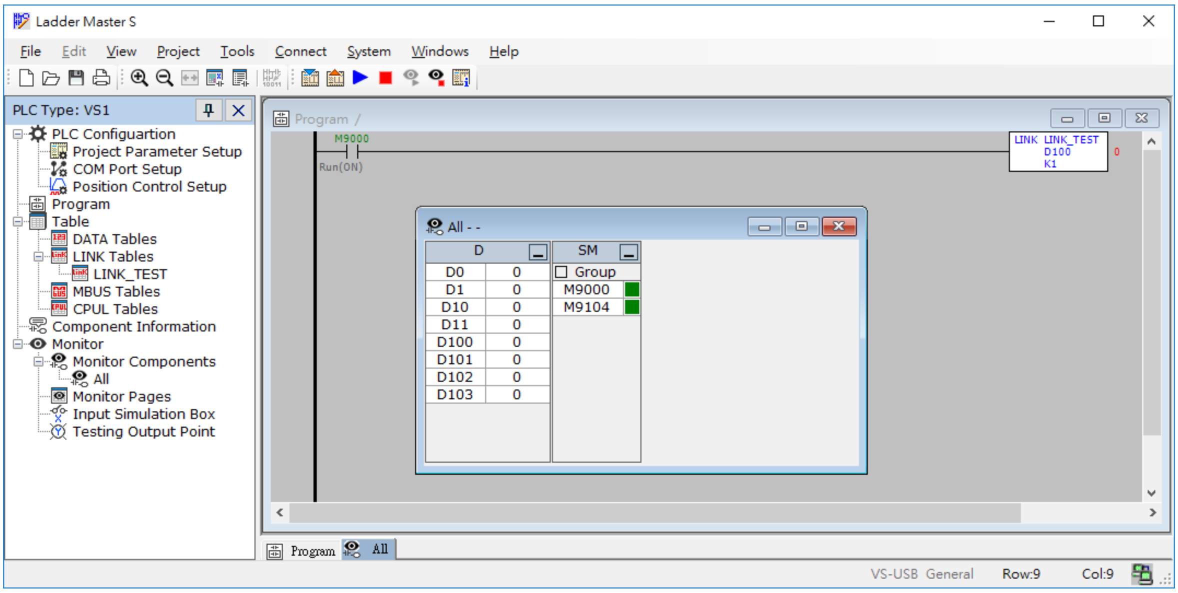
The LINK instruction bases on the contents of the “LINK_TEST” table to operate communication.

By the USB programming port, the project is written into the Master PLC.



④ After sequentially edit and load the projects to those PLCs, do the wiring jobs between three PLCs. Then, connect the computer to the USB programming port of the Master PLC, to test and monitor the process through the Ladder Master S.

⑤ Read the project from the Master PLC then execute the monitor mode, at the screen to add and monitor the components D0, D1, D10 and D11.



When the example is performing, the data at every PLC will follow its own program and the "LINK_TEST" communication table to execute the operation below.

1. The content value of Master's D0 is written into the D0 at the Slave PLC #1 via communication (the first command in the communication table). At the Slave #1, the content value of D0 is sent to its output points Y0~Y7.
   Therefore, when the value of Master's D0 is 1, the Y0 at the Slave #1 turns "ON"; when the value of Master's D0 is 5, the Y0 and Y2 at the Slave #1 turn "ON", and so forth.

   At the PLC Slave #1, its content value of D0 will be added 100 up and stored to the D1.

   Afterward, the PLC Master reads the value of D1 at the Slave #1 via the communication and stores the value to the Master's D10 (the second command in the communication table).
   Thus, to observe the value of D10 at the Master that is always equal to the content value of D0 added up to 100 if the communication is successful.

2. The content value of Master's D1 is written into the D0 at the Slave PLC #2 via communication (the third command in the communication table). At the Slave #2, the content value of D0 is sent to its output points Y0~Y7. Therefore, when the value of Master's D1 is 1, the Y0 at the Slave #2 turns "ON"; when the value of Master's D1 is 7, the Y0, Y1 and Y2 at the Slave #1 turn "ON", and so forth.

   At the PLC Slave #2, its content value of D0 will be added 100 up and stored to the D1.

   Afterward, the PLC Master reads the value of D1 at the Slave #1 via the communication and stores the value to the Master's D11 (the fourth command in the communication table).
   Thus, to observe the value of D11 at the Master that is always equal to the content value of D1 added up to 100 if the communication is successful.

By the VS Computer Link communication, the LINK instruction and its table at the Master PLC that produces the following results:

1.The content value of the Master's D0 is output to the points Y0~Y7 of the Slave #1.

2.The content value of the Master's D10 equals to the sum of the content value of its D0 and 100.

3.The content value of the Master's D1 is output to the points Y0~Y7 of the Slave #2.

4.The content value of the Master's D11 equals to the sum of the content value of its D1 and 100.

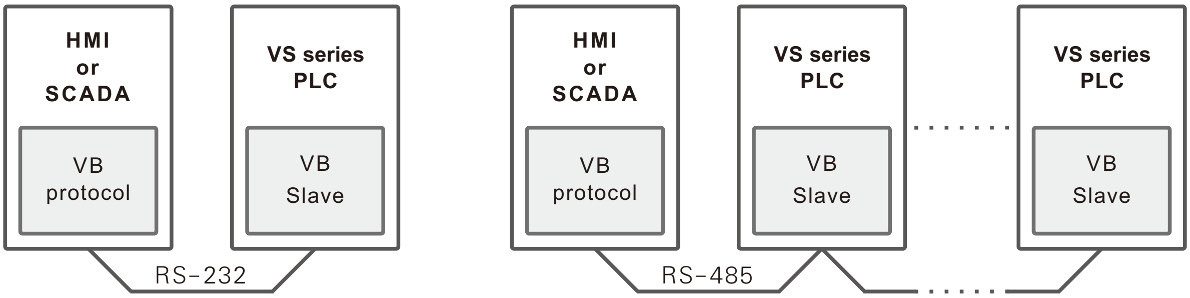
7-3-4 VB Computer Link Slave

When the VS PLC's communication port has been set to the application type of the "VS Computer Link Slave", the human-machine interface (HMI) or supervisory control and data acquisition system (SCADA) can access the VS PLC's data through the "M, VB and VH series Communication Protocol" (hereby referred to as VB protocol).

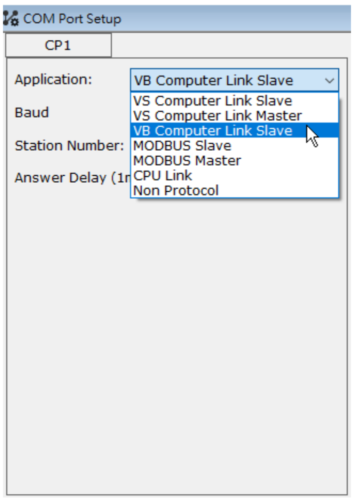
This application type is mainly applied to the following situations:

1. When the old versions of HMI or SCADA can only support the "M, VB and VH Protocol" but not support the "VS Computer Link Protocol".
2. When the existed "VB Protocol" network is added with a new VS series PLC, the added VS PLC must work under this application type.

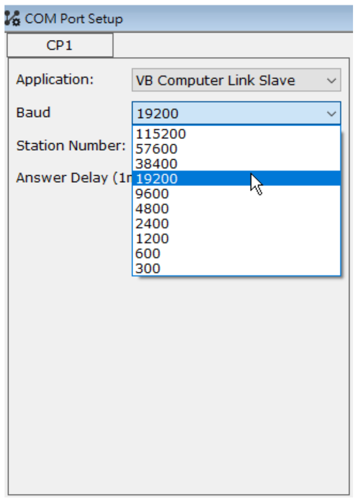
When the VS PLC works under this application type, the components can be accessed are limited to the previously VB series PLC can support. (The components which the VS PLC can be used are more then the VB series have.)



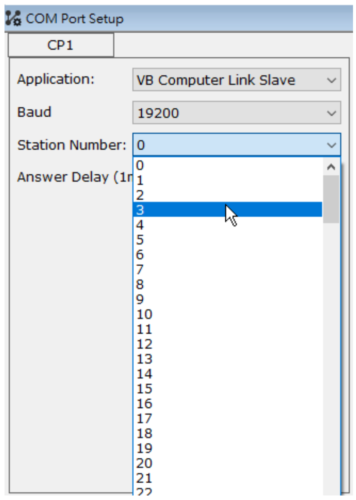
Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	M, VB and VH series Communication Protocol (hereby referred to as VB protocol)	
Commection Method	Half-duplex	
Communication Paramete	Bits-per-character: 7 bits; parity check: Even; stop bit: 1 bit	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
No. of Slave Stations	1 Slave station only	Max. 256 stations (Must insert the RS-485 repeater(s) if the RS-485 wiring with more than 32 devices.)
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	CP1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the X, Y, M, S, T, C and D are included	



Set the application type:
VB Computer Link Slave



Set the baud rate

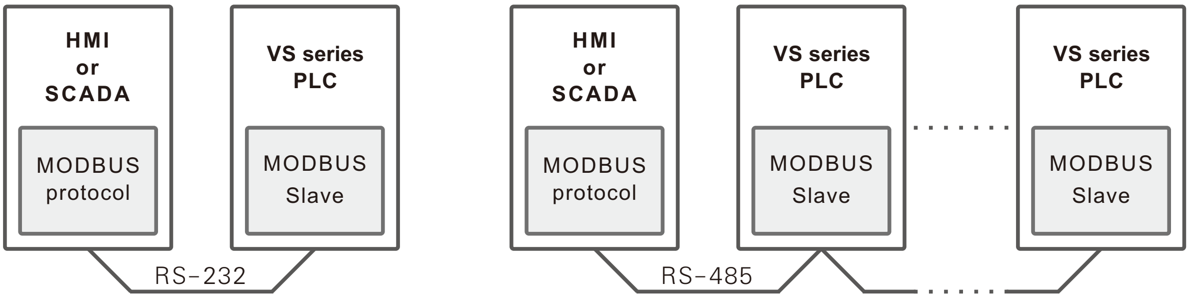


Set the station number

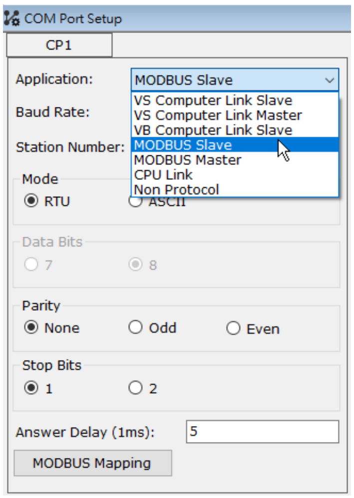
※ Usually the default delay response time is not necessary to change, except that:
When the suspect of bad communication is caused by the problem of signal collision in the circuit, try to extend the delay response time.

7-3-5 MODBUS Slave

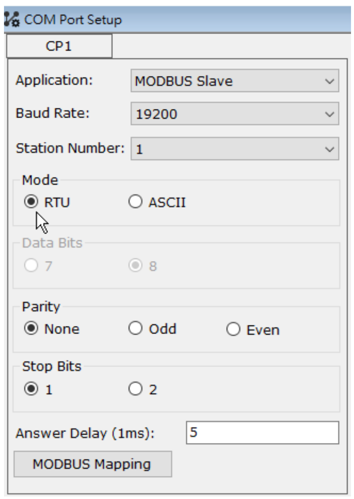
MODBUS is a very popular communication protocol on the market. Various common industrial products such as the human-machine interface (HMI) or the supervisory control and data acquisition system (SCADA) all support the “MODBUS protocol”. Nonetheless, if the selected HMI or SCADA does not support the “VS Protocol”, use the “MODBUS protocol” to connect it with the VS series PLC’s.



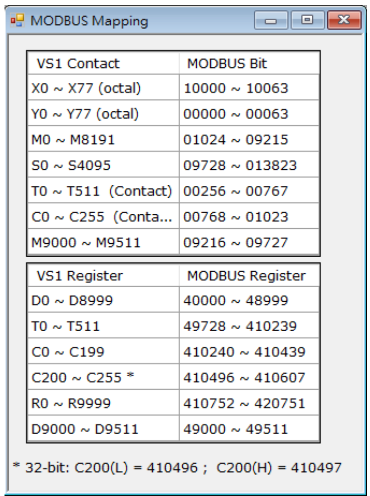
Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	MODBUS Protocol	
Commection Method	Half-duplex	
Communication Paramete	Format: ASCII or RTU; bits-per-character: 7 bits / 8 bits; parity check: None / Even / Odd; stop bit: 1 bit / 2 bits	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
No. of Slave Stations	1 Slave station only	Max. 247 stations (Must insert the RS-485 repeater(s) if the RS-485 wiring with more than 32 devices.)
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	CP1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the X, Y, M, S, T, C, D, R are included	



Set the application type:
MODBUS Slave



Set the corresponding parameters in the sheet



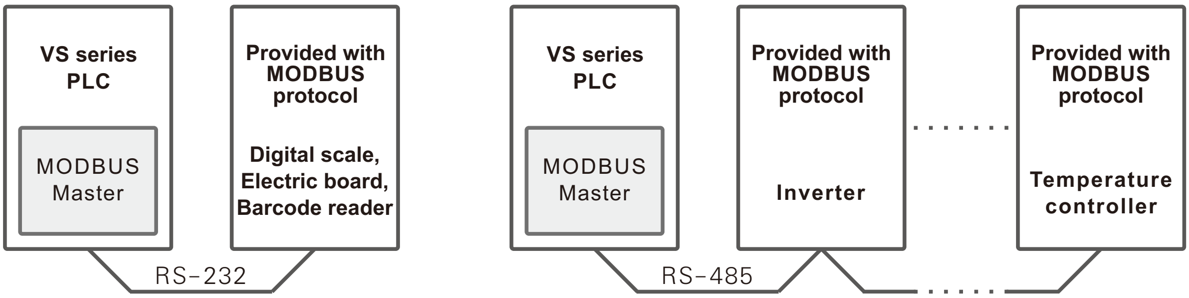
The MODBUS Mapping can be expanded to get the VS PLC’s components

※ Usually the default delay response time is not necessary to change, except that:
When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

7-3-6 MODBUS Master

Many automation devices and equipment (e.g. inverters, temperature controllers) support the "MODBUS protocol". Hence, the VS PLC provides the MBUS instruction and MBUS communication table, that cooperatively produces the MODBUS commands to access devices which possess the MODBUS communication function, to serve the purpose of exchanging data between the PLC and these devices.

When the VS PLC's communication port has been set to the application type of the "MODBUS Master", it cooperates with the MBUS instruction and a MODBUS communication table to operate communication procedures. This Master links and communicates with MODBUS Slaves (such as inverters, temperature controllers, power meters, etc.) through the "MODBUS protocol".



Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	MODBUS Protocol	
Commection Method	Half-duplex	
Communication Paramete	Format: ASCII or RTU; bits-per-character: 7 bits / 8 bits; parity check: None / Even / Odd; stop bit: 1 bit / 2 bits	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
No. of Slave Stations	1 Slave station only	Max. 247 stations (Must insert the RS-485 repeater(s) if the RS-485 wiring with more than 32 devices.)
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	CP1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the X, Y, M, S, T, C, D, R are included	

FNC 149	MBUS (S1) (S2) (n)	MODBUS Communication Instruction	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
n																○		
• S2 occupies 4 components • For the VS1, VS2 or VSM series, n = 1~3; for the VS3 series, n = 1~5																		

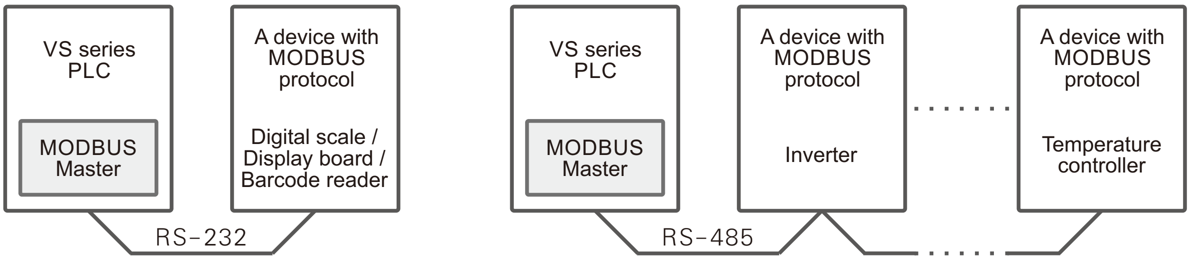
```
                     (S1)                (S2)  (n)
 X20
─┤ ├────[MBUS  ACCESS_MBS_EQUIP  D100  K1]
```

S1 : the communication table that describes the data receiving and transmitting

S2 : the working area for the instruction that occupies 4 registers

n : to assign the communication port, 1~5 = CP1~CP5

- The VS series PLC uses this instruction to transmit or get the designated data via its Communication Port CP1~CP5 with other peripherals which use the MODBUS ASCII or RTU protocol.
- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the "Application type:" become the "MODBUS Master". Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the "Project" -- "COM Port Setting".
- As shown in the figure below, connect the PLC with other device(s) by the RS-232 or RS-485 wiring. Furthermore, to set a unique station number (1~247) for every device and make the other parameters identical. Besides, use the Ladder Master S to open the project of the VS series PLC and write this MBUS instruction in the Master PLC's program, then set the port's "Application type:" as the "MODBUS Master" also to set the communication parameters in that window. After that, edit the MODBUS communication table to designate data transferring actions. Moreover, to load this project to the Master PLC. Following these procedures, the purpose of data delivery between the PLC and peripherals can be reached.



- When X20 = "ON", the MBUS instruction starts execution. According to the contents described in the "ACCESS_MBS_EQUIP" communication table, it proceeds data write or read actions to the appointed peripherals, while the D100~D103 will be occupied to store the state of instruction execution.
- When the contents of the communication table specified by the (S1) are execution completed from beginning to end, the M9104 will be "ON" for a Scan Time then the procedure will start over again from the first item set of the table.
- When the X20 turns from "ON" to "OFF", this instruction stops and the data sharing immediately stop but data which has transferred previously will still remain.
- The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
■M9104	CP1 LINK / MBUS instruction on execution table complete once flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
■M9114	CP2 LINK / MBUS instruction on execution table complete once flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
■M9124	CP3 LINK / MBUS instruction on execution table complete once flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
■M9134	CP4 LINK / MBUS instruction on execution table complete once flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.
■M9144	CP5 LINK / MBUS instruction on execution table complete once flag.

• The communication table is assigned by the S1 of the instruction.

An example of the “MBUS” provided by programming tool the Ladder Master S is shown below.

Item No.	Command	Device at Master	Direction	Slave St. No.	Device at Slave	Length	Word/Bit	Disable Contact
1	H03 Read (4x)	D0	<--	1	1000	10	—	M1
2	H10 Write (4x)	D10	-->	1	1100	5	—	M1
3	H03 Read (4x)	D20	<--	2	1000	10	—	M2
4	H10 Write (4x)	D30	-->	2	1100	5	—	—
5	H10 Write (4x)	D40	-->	0	100	20	—	M0

The first item set in the table means that the Master reads the data 41000~41009 from the Slave #1 and stores them at the D0~D9 of the Master.

The second item set in the table means that the Master writes its data at D10~D14 to the 41100~41104 of the Slave #1.

The third item set in the table means that the Master reads the data 41000~41009 from the Slave #2 and stores them at the D20~D29 of the Master.

The fourth item set in the table means that the Master writes its data at D30~D34 to the 41100~41104 of the Slave #2.

The fifth item set in the table means that the Master writes its data at D40~D59 to the 40100~40119 of all the Slaves. To set the Slave station #0 means that the Master writes the data to all of the Slaves. Note that Read Command cannot be used in this application.

The last column of the table is for the Disable Contact. If that specified contact is “ON”, the communication item set will be ignored. For example, if M1 = “ON”, the first and second item sets in the table will skip. This is the new function of the VS series, which can help designers effectively manage the operation of the communication table. A communication item set does not need to specify a Disabled Contact (such as the fourth item set), so that the command of the set does not have the disable control function.

• The working area of the executive instruction is starting from the S2 (using D100~D103 as the example).

S2	**Description**	
D100	Lower 8 bits	The record of the Slave's station number when the first communication error is occurred
	Upper 8 bits	Instruction working status 0: Normal data transmitting / receiving 2: The length of the received data is incorrect 4: The error is caused by the assigned device is inappropriate 7: The error is caused by the communication command is incorrect A: The communication setting is normal but no response from the Slave station (Time-out occurs) B: Abnormal communication
D101 ~ D103	The working area is required for the system when this instruction is performed	

The MBUS instruction will reset the D100 to be “0” before the first item set at the table is executed.

If any communication error occurs during the execution of the instruction, a code will be recorded in D100. Only when the content value of D100 is “0”, the recording action can be executed. Therefor, when there are possibilities of several errors, users can use the program to move out the content value of D100 then reset it to “0”. This way allows the D100 to record the next error.

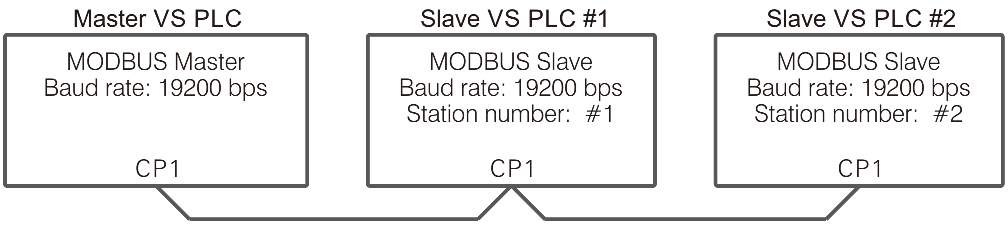
• To edit a communication table

Use the Ladder Master S to set up a MBUS communication table and through its interactive window can set up and edit a communication table easily.

In the structure of VS series PLC, the communication tables are a part of the project. When the programmer to copy or access the project, those tables will be duplicated automatically with the program.

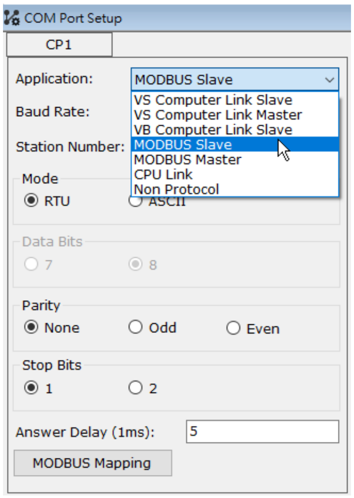
- Application Example

In this example, those CP1s of three VS PLCs are connected by the RS-485 interface and execute the MODBUS communication.
In the following figure, let the first one from the left be the Master station of MODBUS communication to execute the "MODBUS Master" application type and set the baud rate to be 19200 bps. Add the MBUS instruction in the program and then edit the MBUS communication table, thereby to process data transmitting from or to the MODBUS Slaves.
Another two are the MODBUS Slave PLCs, executing the "MODBUS Slave" application type. The baud rate of those connected ports should be the same as the port of the Master PLC. Then, individually set the station numbers of Slaves become the #1 and #2. Afterward, write the relevant control programs into the PLCs.

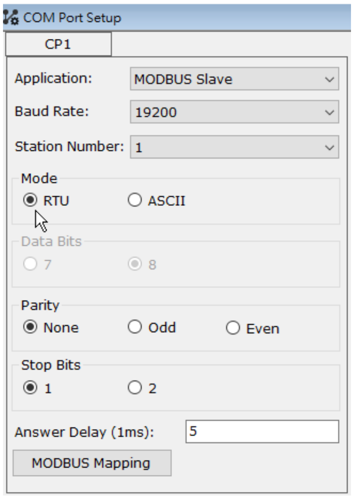


Please follow the procedures below to operate the test:

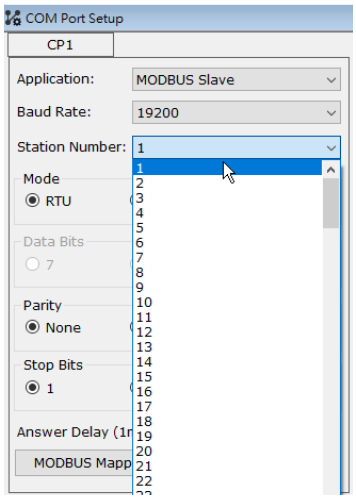
① Edit the project of Slave PLC #1
Use the Ladder Master S to set the CP1's parameters of Slave PLC #1 and compile relevant program. Then, connect to the USB programming port of Slave #1 and write the project into the PLC.



Set the application type: MODBUS Slave



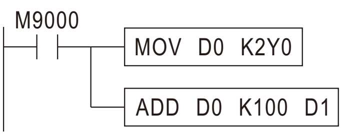
Set the baud rate and relevant parameters



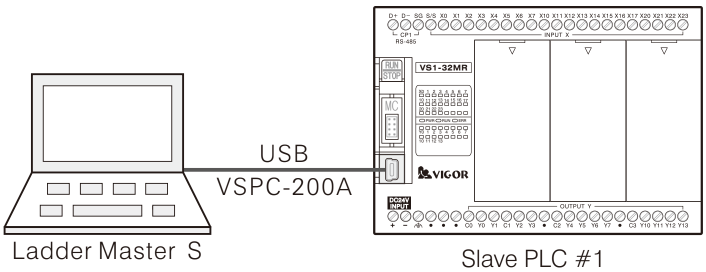
Set the station number: #1

※ Usually the default delay response time is not necessary to change, except that:
When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

The program writes to the Slave PLC #1.

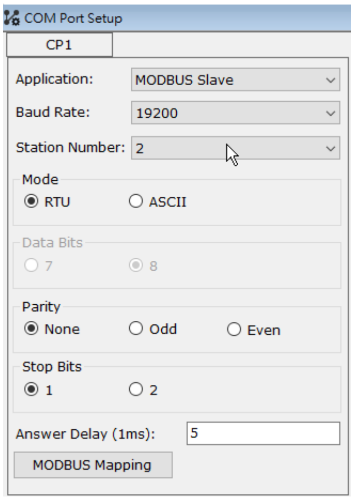


By the USB programming port, the project is written into the Slave PLC #1



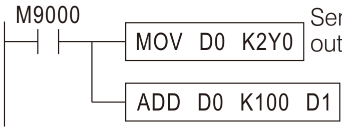
② Edit the project of Slave PLC #2

Use the Ladder Master S to set the CP1's parameters of Slave PLC #2 and compile relevant program. Then, connect to the USB programming port of Slave #2 and write the project into the PLC.



Set the station number: #2

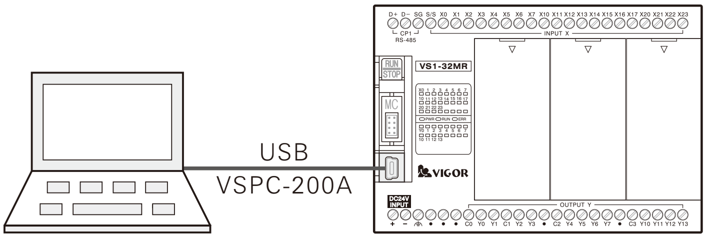
The program writes to the Slave PLC #2



Send the content value of D0 to the output points Y0~Y7.

Add 100 to the content value of D0 and store it to the D1.

By the USB programming port, the project is written into the Slave PLC #2

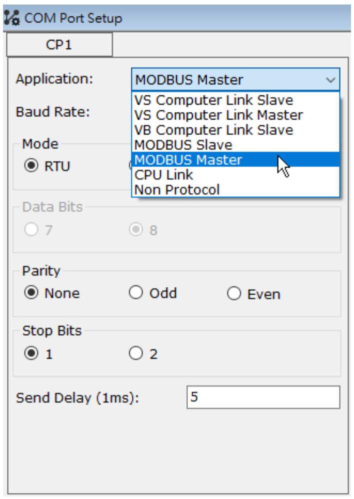


Ladder Master S

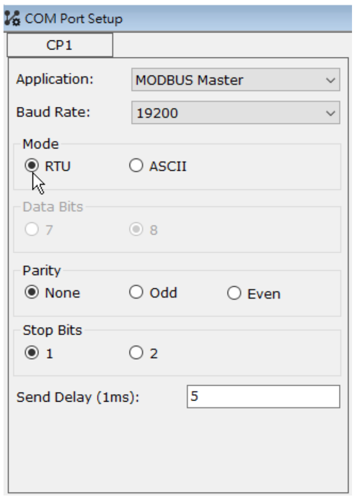
Slave PLC #2

③ Edit the project of the Master PLC

Use the Ladder Master S to set the CP1's parameters of Master PLC. Compile the MBUS communication table, the MBUS instruction and relevant program. Then, connect to the USB programming port of Master and write the project into the PLC.



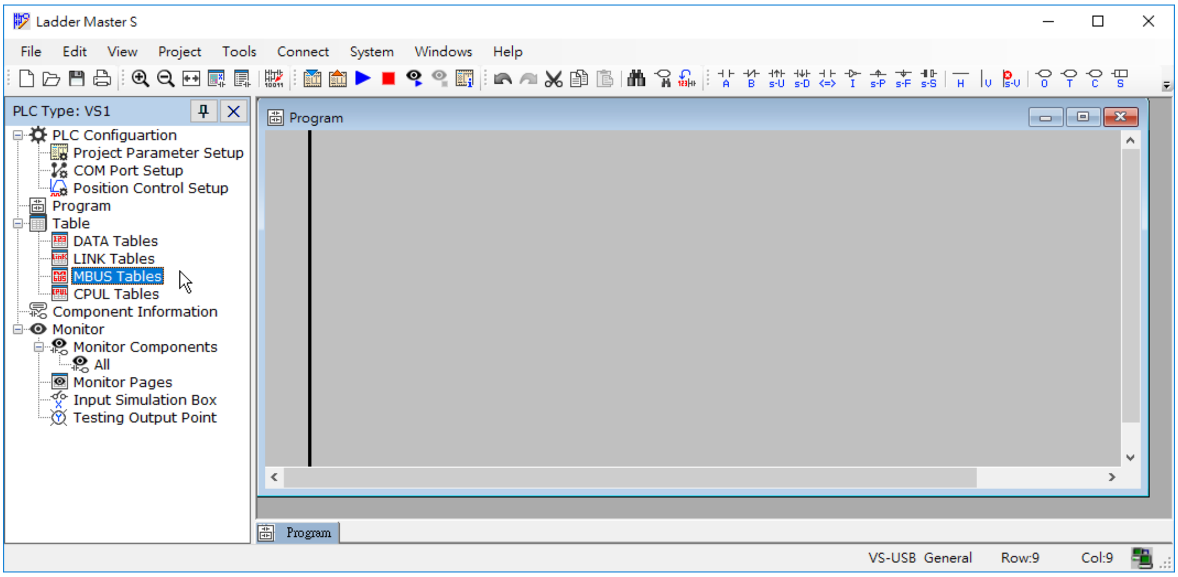
Set the application type: MODBUS Master



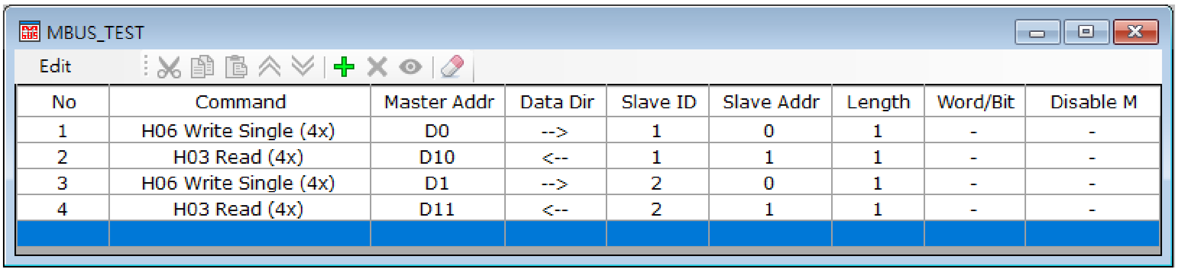
Set the baud rate and relevant parameters

※ Usually the default delay response time is not necessary to change, except that: When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

Open a new MBUS communication table at the Master PLC



Name this MBUS communication table as “MBUS_TEST” first, and then compile its contents.



MBUS_TEST

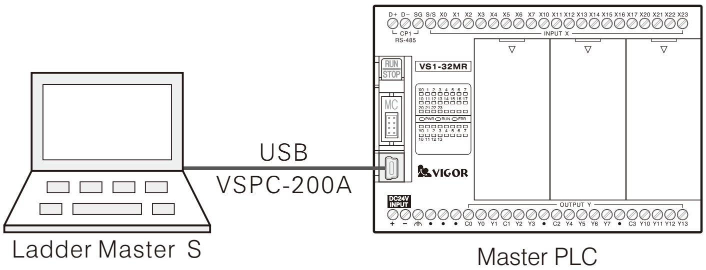
Edit

No	Command	Master Addr	Data Dir	Slave ID	Slave Addr	Length	Word/Bit	Disable M
1	H06 Write Single (4x)	D0	-->	1	0	1	-	-
2	H03 Read (4x)	D10	<--	1	1	1	-	-
3	H06 Write Single (4x)	D1	-->	2	0	1	-	-
4	H03 Read (4x)	D11	<--	2	1	1	-	-

Edit the program for the Master PLC.

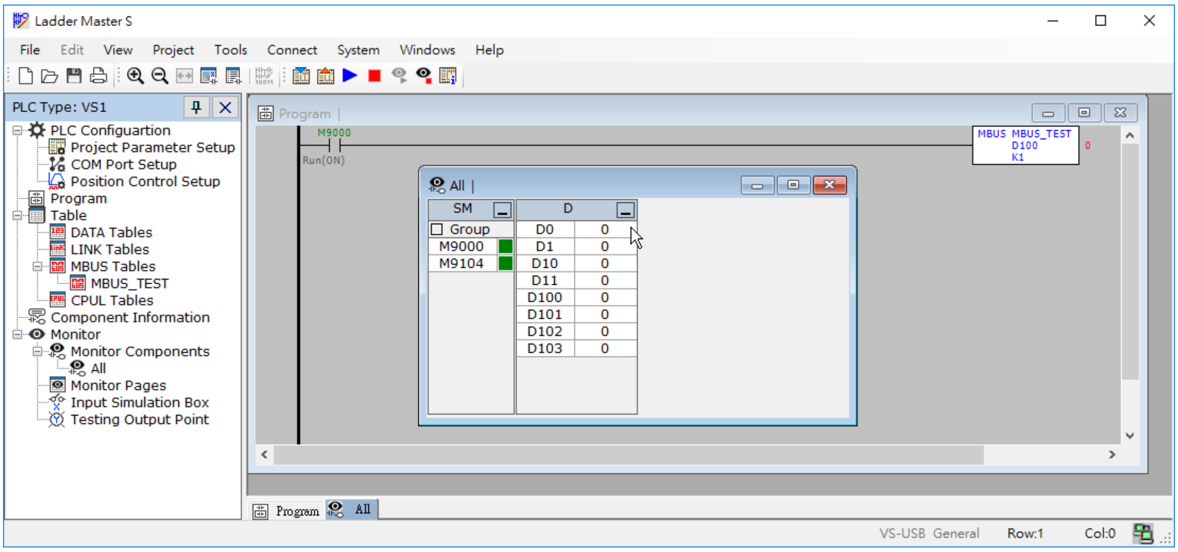
The MBUS instruction bases on the contents of the “MBUS_TEST” table to operate communication.

By the USB programming port, the project is written into the Master PLC.



④ After sequentially edit and load the projects to those PLCs, do the wiring jobs between three PLCs. Then, connect the computer to the USB programming port of the Master PLC, to test and monitor the process through the Ladder Master S.

⑤ Read the project from the Master PLC then execute the monitor mode, at the screen to add and monitor the components D0, D1, D10 and D11.



When the example is performing, the data at every PLC will follow its own program and the communication table to execute the operation below.

1. The content value of Master's D0 is written into the 40000 at the Slave PLC #1 via communication (the first command in the communication table that 40000 is corresponding to the D0).

   At the Slave #1, the content value of D0 is sent to its output points Y0~Y7. Therefore, when the value of Master’s D0 is 1, the Y0 at the Slave #1 turns “ON”; when the value of Master's D0 is 5, the Y0 and Y2 at the Slave #1 turn “ON”, and so forth.

   At the PLC Slave #1, its content value of D0 will be added 100 up and stored to the D1.

   Afterward, the PLC Master reads the value of 40001 at the Slave #1 via the communication and stores the value to the Master's D10 (the second command in the communication table that 40001 is corresponding to the D1). Thus, to observe the value of D10 at the Master that is always equal to the content value of D0 added up to 100 if the communication is successful.

2. The content value of Master’s D1 is written into the 40000 at the Slave PLC #2 via communication (the third command in the communication table that 40000 is corresponding to the D0). At the Slave #2, the content value of D0 is sent to its output points Y0~Y7. Therefore, when the value of Master's D1 is 1, the Y0 at the Slave #2 turns “ON”; when the value of Master's D1 is 7, the Y0, Y1 and Y2 at the Slave #1 turn “ON”, and so forth.

   At the PLC Slave #2, its content value of D0 will be added 100 up and stored to the D1.

   Afterward, the PLC Master reads the value of D1 at the Slave #1 via the communication and stores the value to the Master's D11 (the fourth command in the communication table that 40001 is corresponding to the D1). Thus, to observe the value of D11 at the Master that is always equal to the content value of D1 added up to 100 if the communication is successful.

By the MODBUS communication, the MBUS instruction and its table at the Master PLC that produces the following results:

1.The content value of the Master's D0 is output to the points Y0~Y7 of the Slave #1.

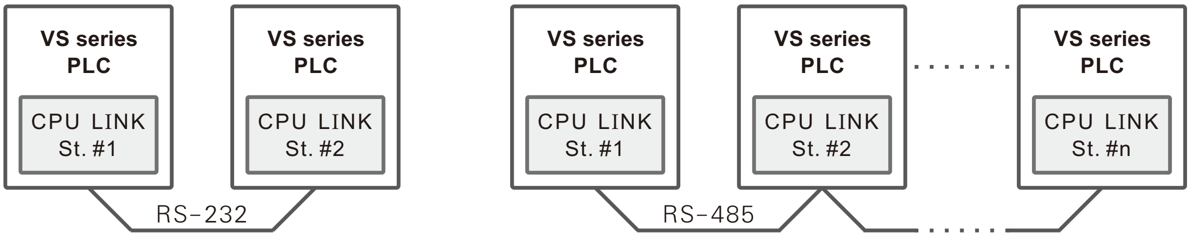
2.The content value of the Master's D10 equals to the sum of the content value of its D0 and 100.

3.The content value of the Master's D1 is output to the points Y0~Y7 of the Slave #2.

4.The content value of the Master's D11 equals to the sum of the content value of its D1 and 100.

7-3-7 CPU Link

The VS series PLC uses this application to share instant data between VS PLCs to serve the purpose of distributed control.
When the communication ports of VS PLCs have been set the application for to share instant data between PLC's, select one of the PLC to cooperate with the CPUL instruction and CPUL communication table, through its particular protocol to share instant data.



Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	CPU Link particular protocol	
Commection Method	Half-duplex	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
No. of Slave Stations	1 Slave station only	Max. 256 stations (Must insert the RS-485 repeater(s) if the RS-485 wiring with more than 32 devices.)
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	Cp1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the Y, M, S, T, C, D, R are included	

- The most of the VS series PLC communication application types are replying at the particular time that is when the PLC is finished its user program cycle. Thus, the spending time of entire communication does not only depend on the baud rate, but also affected by the PLC's Scan Time. Therefore, it is not easy to calculate the communication turnaround time.
- The CPU Link proceeds the communication by the way of immediate interrupt and thus its communication response is the fastest. Also, the communication turnaround time (TAT) can be calculated. Therefore, it is suitable for a distributed control system which needs to be quickly responded.
- The calculation of the communication turnaround time (TAT) is introduced below:
  The communication turnaround time is the total time to complete all the item sets in the communication table (excluding disabled item set).
  The calculation of the time for each communication item set is introduced below:

$$\frac{(11+\text{The number of data bytes to be transmitted})\times 11}{\text{Baud rate (bps)}}+20\mu s$$

Example: The communication baud rate is 38,400 bps. As the communication table at the right shows, there are three item sets. Then the communication turnaround time (TAT) is calculated as follows:

Item No.	Station No.	Device Range	Word/Bit	Disable Contact
1	0	D0 – D9	—	M1
2	1	M0 – M7	—	M1
3	2	D10	—	M2

The spending time of the 1st item set is: $\frac{(11+(10\times 2))\times 11}{38400}+20\mu s = 8.88ms+20\mu s = 8.9ms$

The spending time of the 2nd item set is: $\frac{(11+(1))\times 11}{38400}+20\mu s = 3.44ms+20\mu s = 3.46ms$

The spending time of the 3rd item set is: $\frac{(11+(1\times 2))\times 11}{38400}+20\mu s = 3.72ms+20\mu s = 3.74ms$

The communication turnaround time (TAT) is the total time taken when the 1st, 2nd and 3rd item sets are completed.

$$TAT = 8.9ms+3.46ms+3.74ms = 16.1ms$$

FNC 87	CPUL (S1) (S2) (n)	CPU Link Communication Instruction	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
n																○		
• S2 occupies 4 components • For the VS1, VS2 or VSM series, n = 1~3; for the VS3 series, n = 1~5																		

```
                      (S1)              (S2)  (n)
  X20
|-| |----[CPUL  DISTRIBUTED_CTRL  D100  K1]
```

S1 : the communication table that describes the data receiving and transmitting

S2 : the working area for the instruction that occupies 4 registers

n : to assign the communication port, 1~5 = CP1~CP5

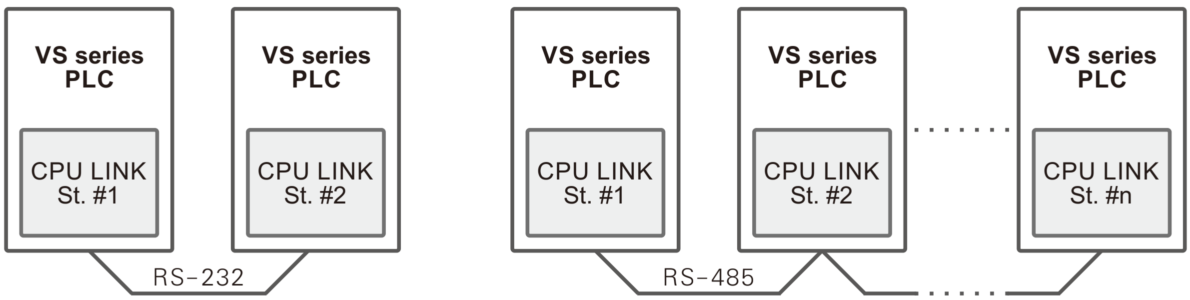
- The VS series PLC uses this instruction to share the particular data via its Communication Port CP1~CP5 with other VS PLCs, that achieved the purpose of distributed control.

- The main character of the CPU LINK is to share instant data.
  What we call "Instant" means all the communication work is dealt with instantly without being affected by the PLC scan time. Other communication method, such as Easy LINK, MODBUS communication and Non Protocol communication, will follow the PLC's executive cycle and thus is affected by the PLC scan time.
  What we call "share" means the way of the CPU LINK communication work is that each station in the communication network automatically transfers a part of data orderly to other stations in the network. This way the operation of the whole communication network can be understood through each station in the communication network.
  Through the function of CPU LINK that instantly sharing data with other PLCs, a distributed control system can be established.

- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the "Application type:" become the "CPU Link". Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the "Project" -- "COM Port Setting".

- This instruction could be used to transfer for various devices that is including all the Y, M, S, T, C, D and R (except for special M and special D).

- As shown in the figure below, first use the Ladder Master S to orderly open every PLC's project and set its "Application type:"as the "CPU Link" then to set an unique station No. (ranging from 0~254) hereafter connect to the programming port of every PLC to load the project to each PLC.
  Afterward, write this instruction in one of the PLC's program then edit the CPUL communication table to designate data sharing actions, hereafter to load this project to the specific PLC. Following these procedures, the purpose of sharing instant data between the PLCs is reached.



- When X20 = "ON", the CPUL instruction starts execution. According to the contents described in the "DISTRIBUTED_CTRL" communication table, it will arrange sharing particular data between stations, while the D100~D103 will be occupied to store the state of instruction execution.

- When the contents of the communication table specified by the (S1) are execution completed from beginning to end, it will start over again from the first item set of the table.

- When the X20 turns from "ON" to "OFF", this instruction stops and the data sharing immediately stop but data which has transferred previously will still remain.

- The communication table is assigned by the S1 of the instruction.
  An example of the “CPU Link” provided by programming tool the Ladder Master S is shown below.

Item No.	Station No.	Device Range	Word/Bit	Disable Contact
1	0	D0 – D9	—	M1
2	1	D10 – D19	—	M1
3	2	D20 – D29	—	M2
4	3	D30 – D39	—	—

※ When the device is selected to be the bit device (Y, M, S or the contact of T, C), the initially ID number at the device range must be a multiplier of 8, such as Y0, Y10, Y20, M0, M24 or T8. Also, the length must be a multiplier of 8 as well, such as M0~M23 (24 devices), T8~T15 (8 devices), Y10~Y27 (16 devices), M24~M63 (40 devices).

The first item set in the table means that Station #0 transmits its D0~D9 to the D0~D9 of other stations in the communication network.

The second item set in the table means that Station #1 transmits its D10~D19 to the D10~D19 of other stations in the communication network.

The third item set in the table means that Station #2 transmits its D20~D29 to the D20~D29 of other stations in the communication network.

The fourth item set in the table means that Station #3 transmits its D30~D39 to the D30~D39 of other stations in the communication network.

The last column of the table is for the Disable Contact. If that specified contact is “ON”, the communication item set will be ignored. For example, if M1 = “ON”, the first and second item sets in the table will skip. This is the new function of the VS series, which can help designers effectively manage the operation of the communication table. A communication item set does not need to specify a Disabled Contact (such as the fourth item set), so that the command of the set does not have the disable control function.

- The working area of the executive instruction is starting from the S2 (using D100~D103 as the example).

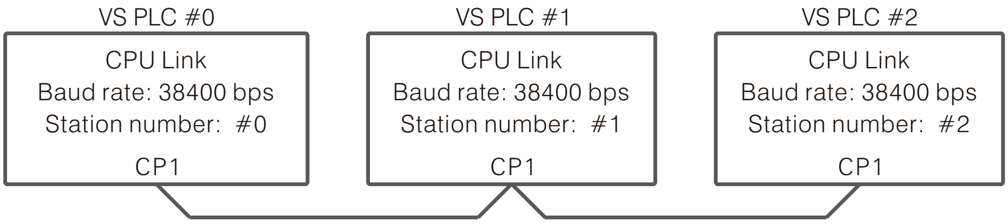
<table>
<tr><th>S2</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">D100</td><td>Lower 8 bits</td><td>The record of the station number when the first communication error is occurred</td></tr>
<tr><td>Upper 8 bits</td><td>Instruction working status
0: Normal data transmitting / receiving
A: The communication setting is normal but no response from the Slave station (Time-out occurs)
B: Abnormal communication</td></tr>
<tr><td>D101
∫
D103</td><td colspan="2">The working area is required for the system when this instruction is performed</td></tr>
</table>

If any communication error occurs during the execution of the instruction, a code will be recorded in D100. Only when the content value of D100 is 0, the recording action can be executed.
Therefor, when there are possibilities of several errors, users can use the program to move out the content value of D100 then reset it to 0. This way allows the D100 to record the next error.

- To edit a communication table
  Use the Ladder Master S to set up a CPUL communication table and through its interactive window can set up and edit a communication table easily.
  In the structure of VS series PLC, the communication tables are a part of the project. When the programmer to copy or access the project, those tables will be duplicated automatically with the program.

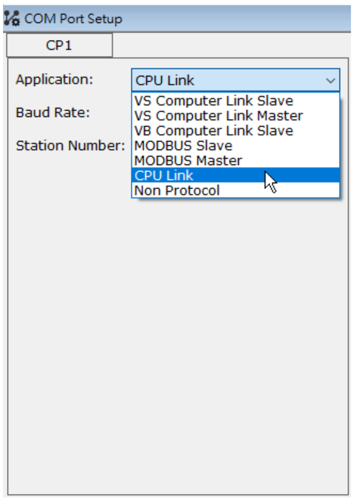
● Application Example

In this example, those CP1's of three VS PLC's are connected by the RS-485 interface and execute the CPU Link communication.
In the following figure, let the first one from the left be the station #0 of the CPU Link communication to execute the "CPU Link" application type and set the baud rate to be 38400 bps. Add the CPUL instruction in the program and then edit the CPUL communication table, thereby to process data sharing between the all CPU Link PLC's.
Another two are also the CPU Link PLC's, executing the "CPU Link" application type. The baud rate of those connected ports should be the same as the port of the Station #0. Then, individually set the station numbers of those become the #1 and #2. Afterward, write the relevant control programs into the PLC's.

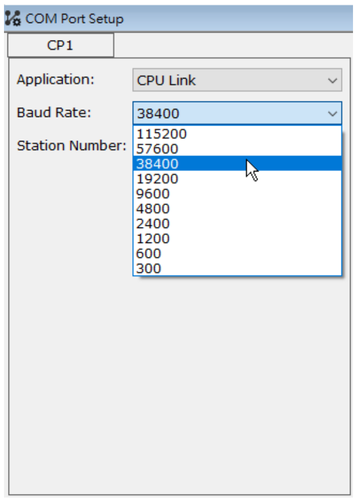


Please follow the procedures below to operate the test:

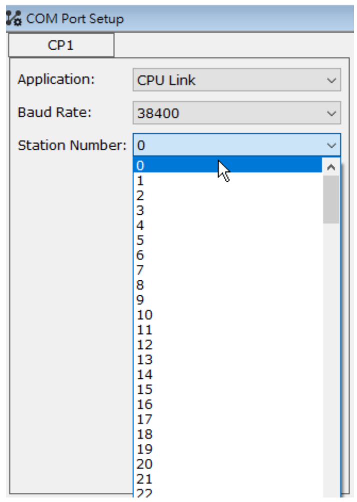
① Edit the project of VS PLC #0
Use the Ladder Master S to set the CP1's parameters of VS PLC #0, edit the communication table and compile relevant program. Then, connect to the USB programming port of VS PLC #0 and write the project into the PLC.



Set the application type:
CPU Link

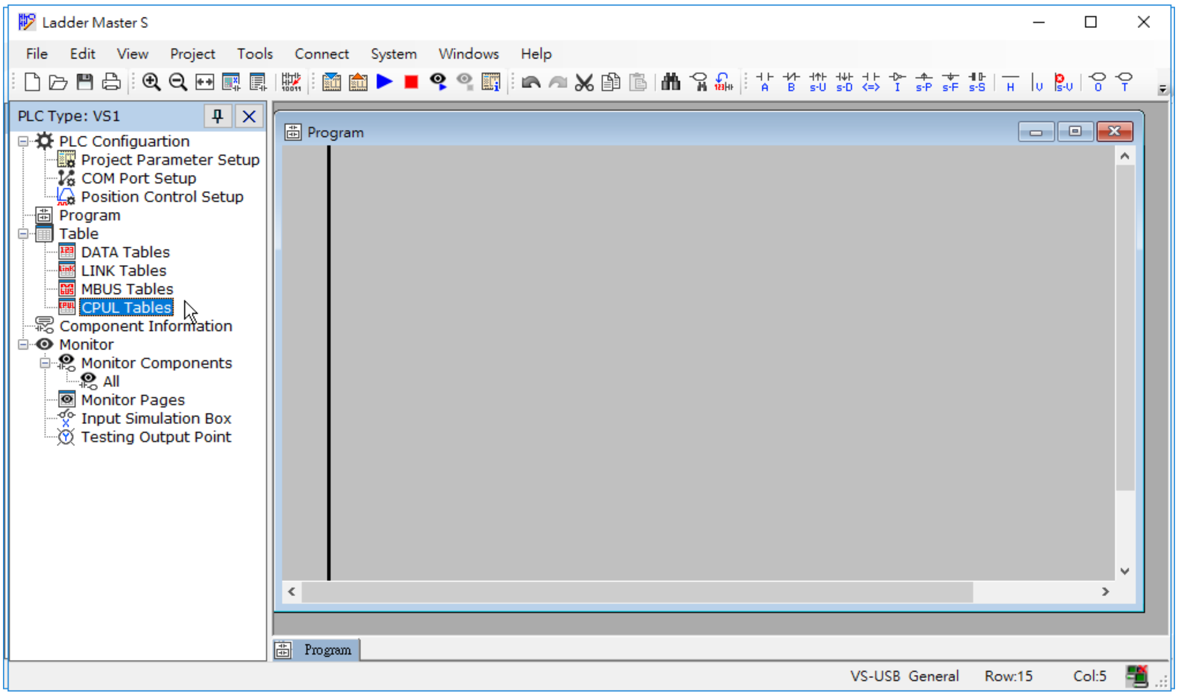


Set the baud rate:
38400 bps.

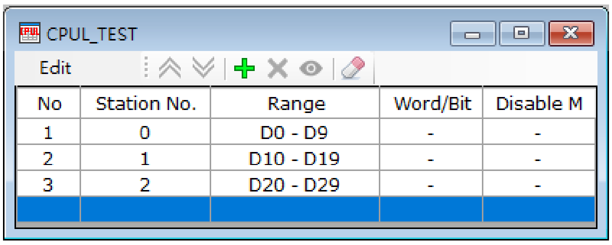


Set the station number:
#0

Open a new CPUL communication table at the VS PLC #0



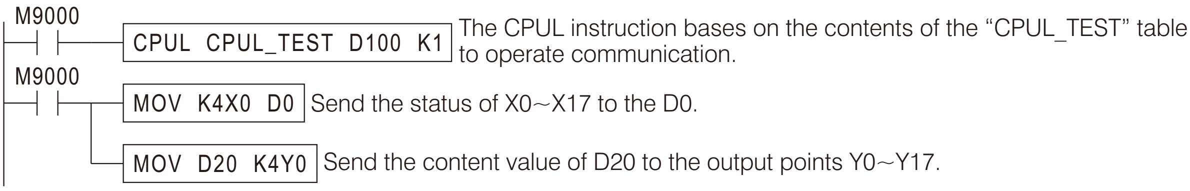
Name this CPUL communication table as the "CPUL_TEST" first, and then compile its contents.



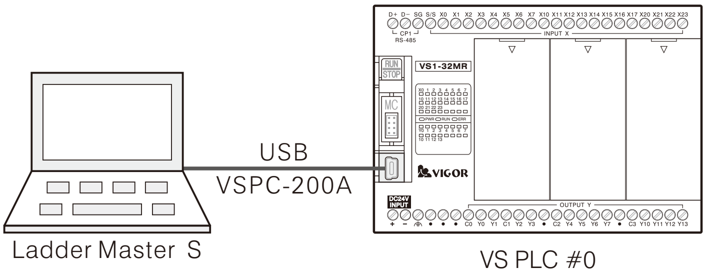
CPUL_TEST

No	Station No.	Range	Word/Bit	Disable M
1	0	D0 - D9	-	-
2	1	D10 - D19	-	-
3	2	D20 - D29	-	-

Edit the program for the VS PLC #0.

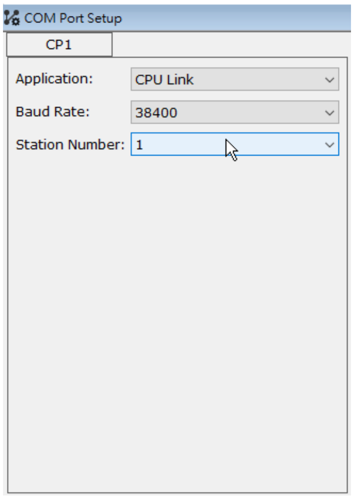


By the USB programming port, the project is written into the VS PLC #0.



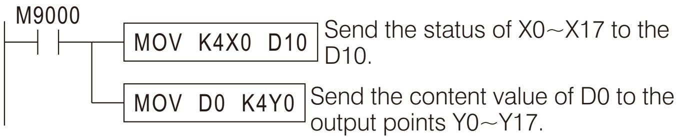
② Edit the project of VS PLC #1

Use the Ladder Master S to set the CP1's parameters of VS PLC #1 and compile relevant program. Then, connect to the USB programming port of VS PLC #1 and write the project into the PLC.

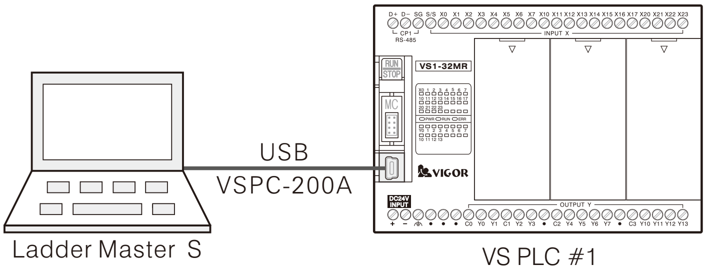


Set the station number:
#1

The program writes to the VS PLC #1.

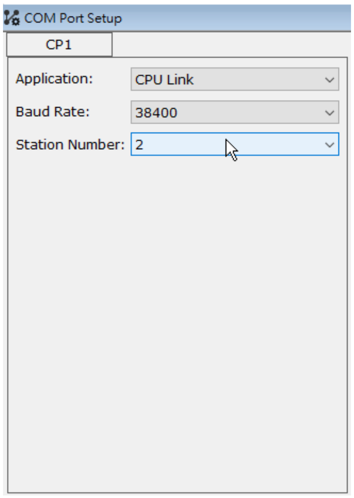


By the USB programming port, the project is written into the VS PLC #1



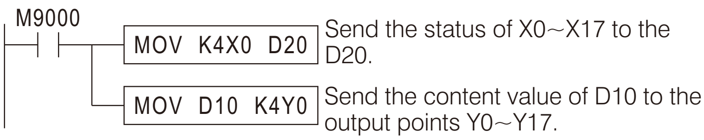
③ Edit the project of VS PLC #2

Use the Ladder Master S to set the CP1's parameters of VS PLC #2 and compile relevant program. Then, connect to the USB programming port of VS PLC #2 and write the project into the PLC.

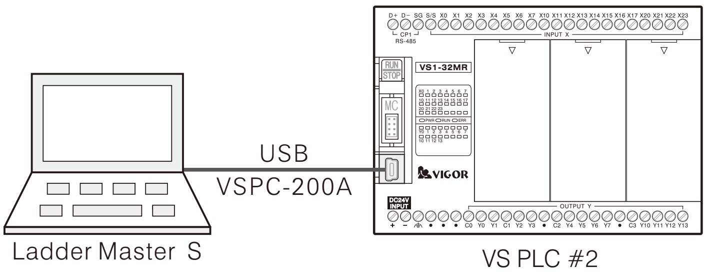


Set the station number:
#2

The program writes to the VS PLC #2.



By the USB programming port, the project is written into the VS PLC #2



④ After sequentially edit and load the projects to those PLC's, do the wiring jobs between three PLC's. Then, trigger the inputs and monitor the response at the other PLC's.

⑤ When the example is performing, the data at every PLC will follow its own program and the communication table to execute the operation below.

VS PLC #0: its statuses X0~X17 send to the D0, and the content value of D20 sends to the output points Y0~Y17.

VS PLC #1: its statuses X0~X17 send to the D10, and the content value of D0 sends to the output points Y0~Y17.

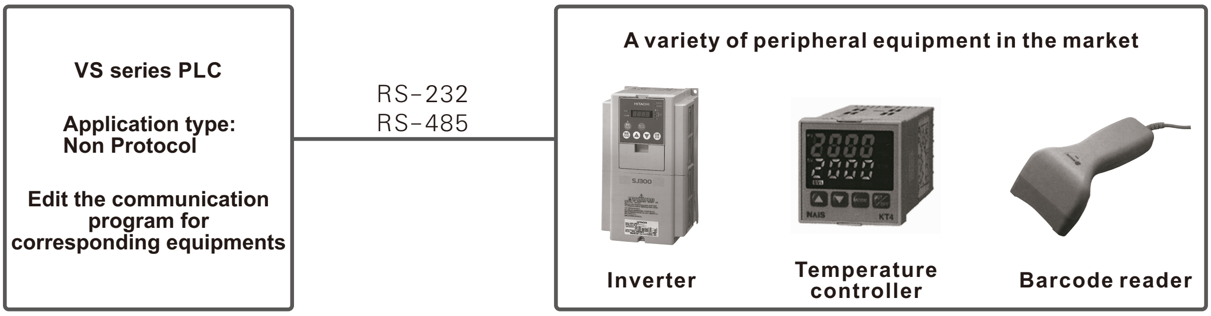
VS PLC #2: its statuses X0~X17 send to the D20, and the content value of D10 sends to the output points Y0~Y17.

By the CPU Link communication, the CPUL instruction and its table at the VS PLC #0 that produces the following results:

The statuses X0~X17 of the VS PLC #0 is reflected to the points Y0~Y17 of the VS PLC #1. The statuses X0~X17 of the VS PLC #1 is reflected to the points Y0~Y17 of the VS PLC #2. The statuses X0~X17 of the VS PLC #2 is reflected to the points Y0~Y17 of the VS PLC #0.

7-3-8 Non Protocol

When the VS PLC's communication port has been set to the application type of the "Non Protocol", the PLC does not execute any specified communication protocol. All communication procedures are operated by the PLC's user program. Then the RS instruction is used to receive and send out data in order to complete the communication operation. This application type is often applied to communicate between PLC and peripheral, such as temperature controllers, inverters and barcode readers.

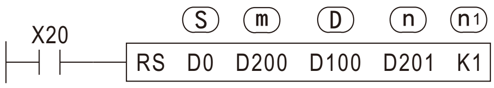


Item	Specification	
Communication Interface	RS-232	RS-485
Communication Protocol	VS Compurter Link Protocol (hereby referred to as VS Protocol)	
Commection Method	Half-duplex	
Communication Paramete	Bits-per-character: 8 bits; parity check: None; stop bit: 1 bit	
Baud Rate	Selectable: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps.	
Distance	Up to 15 meters	Non-isolated: 50 m.; Isolated: 1 km.
Linking Equipment	CP2~CP5: VS-D232-EC CP3, CP5: VS-D52A-EC(CH2)	CP1: Main Unit built-in CP2~CP5: VS-D485-EC, VS-D485A-EC CP2, CP4: VS-485-EC, VS-485A-EC, VS-D52A-EC(CH1)
Available Model	VS series PLC (including VS1, VS2, VSM and VS3)	
Transferable Device	All the X, Y, M, S, T, C, D, R are included	

FNC 80	RS (S) (m) (D) (n) (n1)	Receive/Send Communication Instruction	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●					
m													○			○		
D													●					
n													○			○		
n1																○		

• m, n = 0~4096 • For the VS1, VS2 or VSM series, n_1 = 1~3; for the VS3 series, n_1 = 1~5



S : the head register ID number for the data to be sent

m : the length of the data to be sent

D : the head register ID number for the receiving data

n : the length of the data want to receive

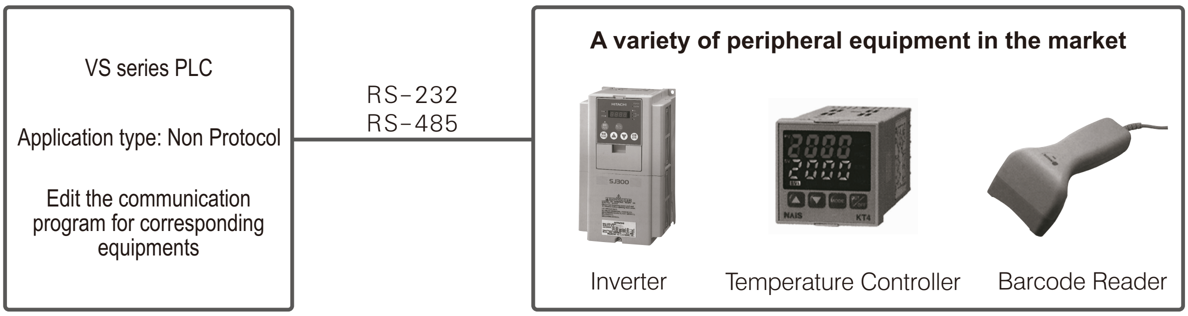
n_1 : to appoint the communication port, 1~5 = CP1~CP5

- The VS series PLC uses this instruction to send out or receive the data via its Communication Port CP1~CP5, and that is by the serial communication interface to communicate with external peripheral equipment.

- As many peripheral equipments in the market (e.g. inverters, barcode readers, card readers, electronic displays, etc.) are equipped with serial communications interfaces and have their own protocols. In order to transfer data between the PLC and those peripherals, a PLC user need to write the program with communicable function therefor this RS instruction is to adapt with these communication protocols.

- The CP1~CP5 are multi-functional communication ports. Each port can choose an appropriate communication type from its various functions. When this instruction is using a port, should choose the “Application type:” become the “Non Protocol”. Regarding the application type selection and related parameter setting, please specify it from the programming software Ladder Master S and at the “Project” -- “COM Port Setting”.

- As shown in the figure below, use the Ladder Master S to set the communication port's “Application type:” of the VS series PLC as the “Non Protocol” and at the same page to set other relevant parameters. Set the station No. and related parameters for each peripheral. Next, in the VS series PLC, according to the communication protocol of the peripheral equipments, compile the program with relevant communication.



- Designate (m) to be K0 when there is not need to send out data and designate (n) to be K0 when there is not need to receive data.

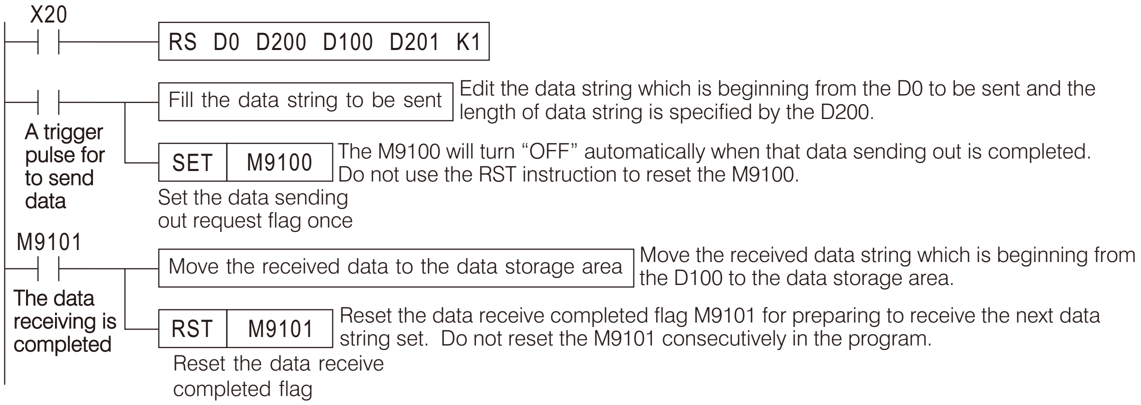
- The data transmissions can be divided into the 16-bit mode (M9161 = “OFF”) or 8-bit mode (M9161 = “ON”) when the RS instruction is performed. Also, the mode flag M9161 should be set before the RS instruction is started.

• The related special devices are summarized below: (■: Means read only.)

Relay ID No.	Description
M9100	CP1 RS instruction data sending out request flag.
M9101	CP1 RS instruction data receive completed flag.
M9102	CP1 RS instruction data receive time-out flag.
M9103	CP1 RS / LINK / MBUS instruction on communication abnormal flag.
M9110	CP2 RS instruction data sending out request flag.
M9111	CP2 RS instruction data receive completed flag.
M9112	CP2 RS instruction data receive time-out flag.
M9113	CP2 RS / LINK / MBUS instruction on communication abnormal flag.
M9120	CP3 RS instruction data sending out request flag.
M9121	CP3 RS instruction data receive completed flag.
M9122	CP3 RS instruction data receive time-out flag.
M9123	CP3 RS / LINK / MBUS instruction on communication abnormal flag.
M9130	CP4 RS instruction data sending out request flag.
M9131	CP4 RS instruction data receive completed flag.
M9132	CP4 RS instruction data receive time-out flag.
M9133	CP4 RS / LINK / MBUS instruction on communication abnormal flag.
M9140	CP5 RS instruction data sending out request flag.
M9141	CP5 RS instruction data receive completed flag.
M9142	CP5 RS instruction data receive time-out flag.
M9143	CP5 RS / LINK / MBUS instruction on communication abnormal flag.

Register ID No.	Description
■D9101	The CP1's amount of residual data to be sent out by the instruction RS.
■D9102	The CP1's amount of the data already received by the instruction RS.
■D9111	The CP2's amount of residual data to be sent out by the instruction RS.
■D9112	The CP2's amount of the data already received by the instruction RS.
■D9121	The CP3's amount of residual data to be sent out by the instruction RS.
■D9122	The CP3's amount of the data already received by the instruction RS.
■D9131	The CP4's amount of residual data to be sent out by the instruction RS.
■D9132	The CP4's amount of the data already received by the instruction RS.
■D9141	The CP5's amount of residual data to be sent out by the instruction RS.
■D9142	The CP5's amount of the data already received by the instruction RS.

Sequence of Data Sending and Receiving (Using the CP1 as an example)



The Related Flags and Data Registers

① The Data Sending Out Flag M9100:

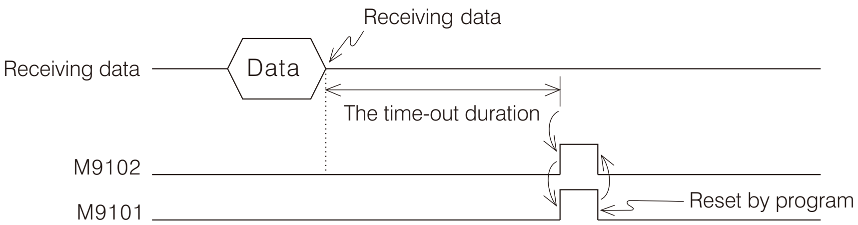
- When the conditional contact X20 = “ON”, the RS instruction is performed. At this time, if a pulse signal forces the status of M9100 to be “ON”, the content values of the registers which are beginning from D0 will be sent out via the serial communication port CP1. After the data sending is completed, the M9100 will be reset to “OFF” automatically.

② The Receive Completed Flag M9101:

- When the conditional contact X20 = “ON”, the RS instruction is performed. The CP1 of PLC is ready to receive communication data.
- When the data receiving is completed, the M9101 = “ON”. At this moment, the received data in the buffer should be moved to the data storage area, and then M9101 could be reset to “OFF”. Afterwards, PLC will be ready for the status of receiving immediately.

③ The Receive Time-out Flag M9102:

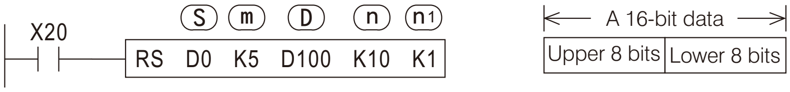
- When the data of receiving is paused and the waiting time exceeds the Time-out duration (designated by the “COM Port Setting”), the M9102 will turn “ON” to represent the happening of Time-out also the Receive Completed flag M9101 will be forced “ON” to terminate the data receiving action.
  The M9102 will not reset automatically, must use an instruction in the program to reset the status of M9101 then the M9102 will be reset too.
- By using the function of the Time-out flag, the PLC could receive the data of transferring from peripheral equipments which without the particular length or “End of Text”.
- The Time-out duration is specified by using the programming software Ladder Master S – “Project” – “COM Port Setting”.



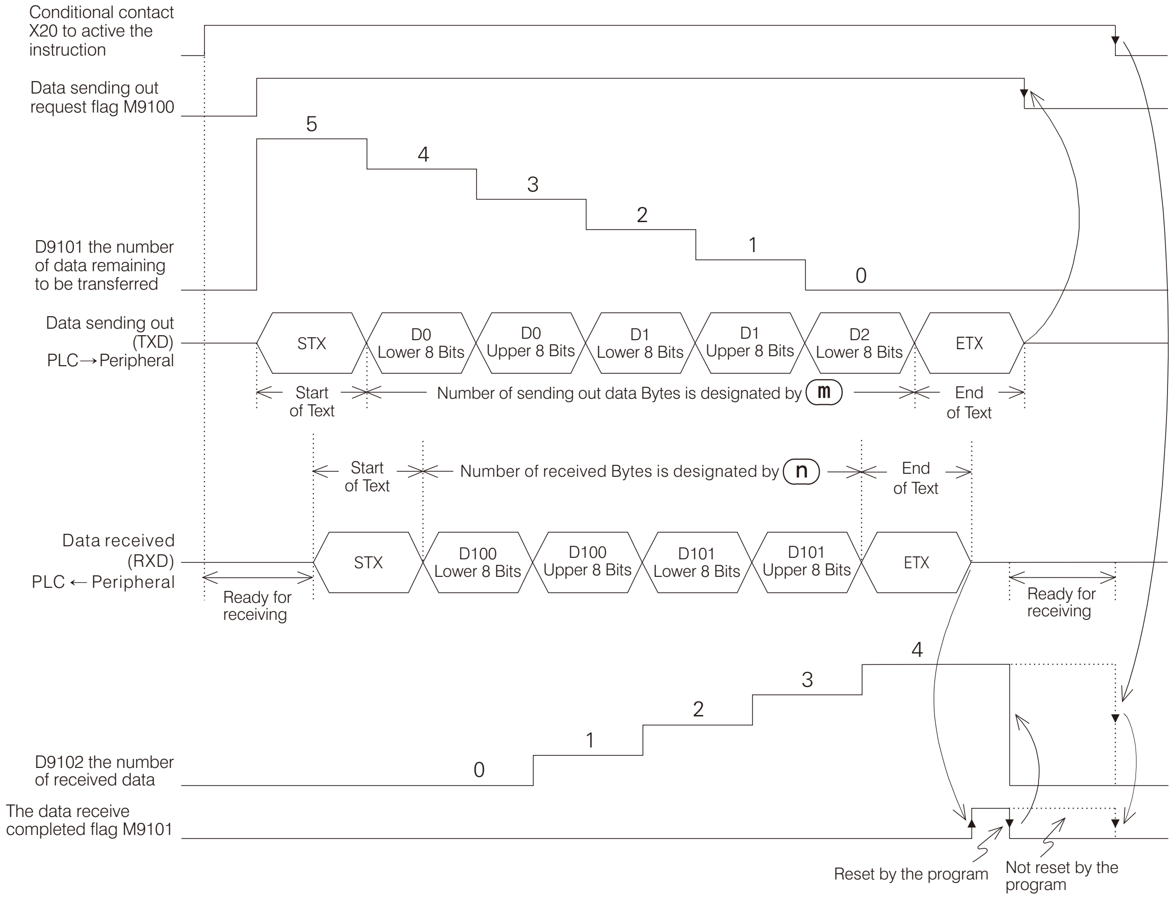
④ The Communication Abnormal Flag M9103:

- When the RS instruction occurs Parity error or Framing error, the Communication Abnormal Flag M9103 will turn ON. The M9103 must use an instruction in the program to reset the status of M9103.

Description of Data Sending / Receiving (by the CP1): 16-bit Mode (M9161 = "OFF")



Divide the 16-bit data into two Bytes of data: Upper 8 bits data and Lower 8 bits data.



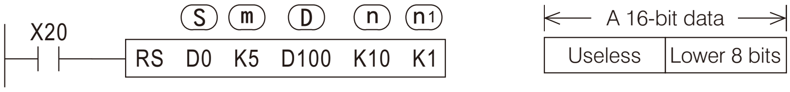
Either one of the condition below is completed will cause the receive completed flag M9101 "ON" and stop to receive.

① The number of assigned n bytes is satisfied.

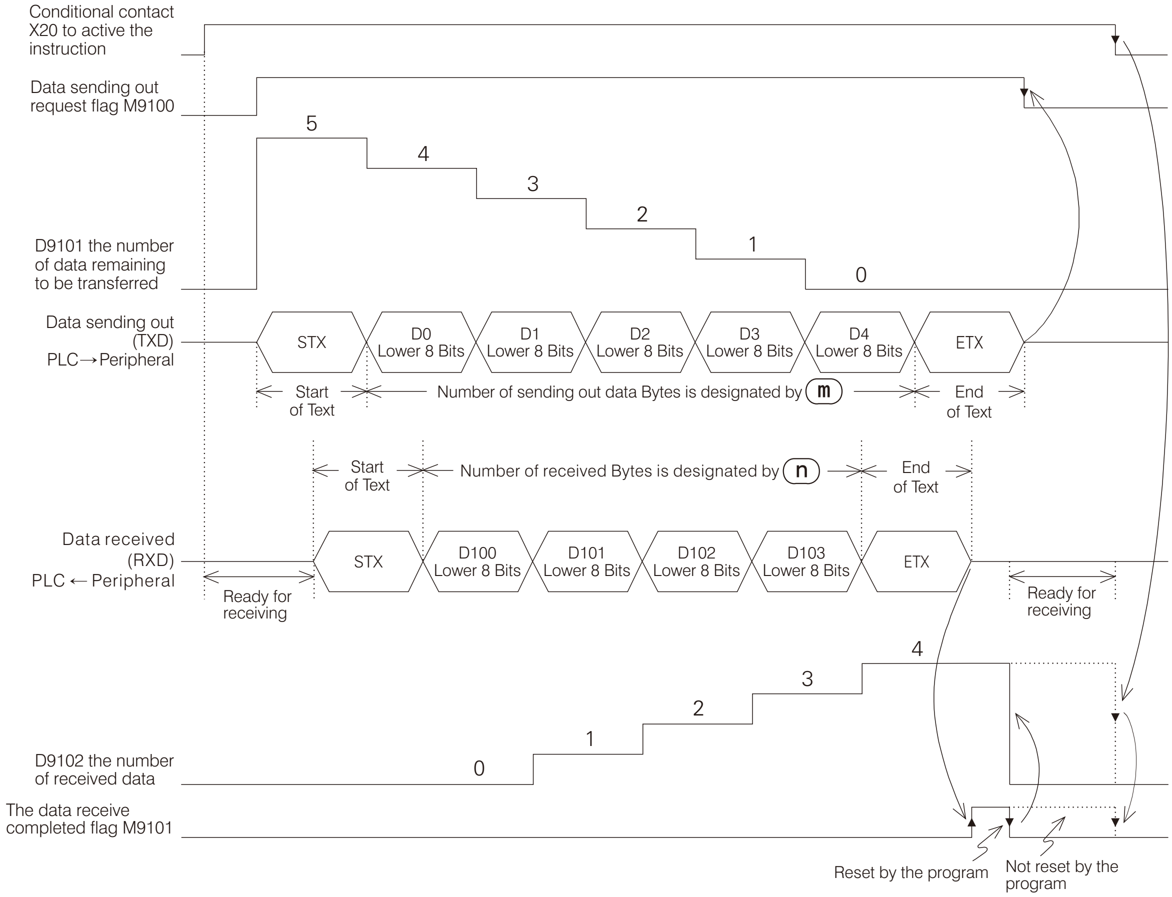
② The End of Text is received.

③ The occurrence of Time-out.

Description of Data Sending / Receiving (by the CP1): 8-bit Mode (M9161 = "ON")



The Lower 8 bits of the 16-bit data is treated as one Byte data. (the Upper 8 bits data are useless)



Either one of the condition below is completed will cause the receive completed flag M9101 “ON” and stop to receive.

① The number of assigned n bytes is satisfied.

② The End of Text is received.

③ The occurrence of Time-out.

- Application Example

In this example, two VS PLC's CP1 are connected by the RS-485 interface and execute the Non Protocol communication.
In the following figure, let the left one executes the "Non Protocol" application type and set the baud rate to be 19200 bps.
Let the right one executes the "VS Computer Link Slave" application type. The baud rate should be the same as the port of the left PLC also set the station number become the Slave #1.
Afterward, edit a relevant communication program and write that into the left PLC. The program uses the RS instruction and follows the "VS Compurter Link Protocol" (hereby referred to as VS Protocol) format to access the right side PLC's data.
Actually, there are simpler ways to exchange data between VS series PLC's in the real applications. This main purpose of this example is to demonstrate the usage of the "Non Protocol" and the RS instruction.

- The application example below briefly explains the relevant descriptions to the "VS Protocol". For the details about the protocol, please refer to the section "7-4 VS Series PLC Communication Protocol".

The parameters of the VS Protocol are --- bits-per-character: 8 bits ; parity check: NONE ; stop bit: 1bit.

The rule of the SUM Check code: Sum up the content values of data from the beginning of the station No. to the end of the data block. Convert the last two digits of the total HEX value into two ASCII codes to be the SUM Check code. Both the data sending and receiving devices should operate the same code-checking procedures that can ensure the data transmission to be correct.

DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
			L	H			L	~	H	L	H			H	L
10H	02H	00H	07H	00H	20H	A0H	34H	12H	00H	05H	00H	10H	03H	31H	32H

Sum up those HEX values and store at the last 2 digits by the ASCII codes
00H+07H+00H+20H+A0H+34H+12H+00H+05H+00H = 112H

The communication command is to read the content value of the D1 from the Slave Station #1. (Assume the content value of the D1 is 0000H)

The command data string is sent to the Slave Station #1

DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
			L	H			L	~	H	L	H			H	L
10H	02H	01H	07H	00H	20H	A0H	01H	00H	00H	01H	00H	10H	03H	43H	41H

The feedback data string from the Slave Station #1

DLE	ACK	Station No.	Number of data Bytes		Error code	Content value of D1		DLE	ETX	SUM Check	
			L	H		L	H			H	L
10H	06H	01H	03H	00H	00H	00H	00H	10H	03H	30H	34H

The communication command to write a value into the D0 in the Slave Station #1. (Assume the data to be written is 0000H)

The command data string is sent to the Slave Station #1

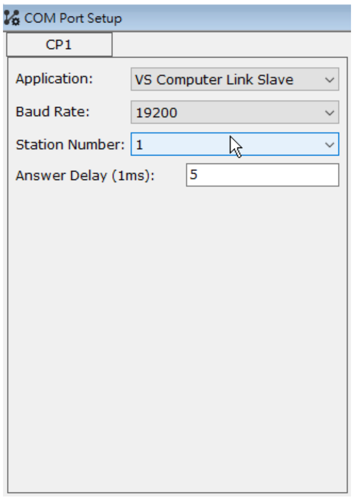
DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		Data write to D0		DLE	ETX	SUM Check	
			L	H			L	~	H	L	H	L	H			H	L
10H	02H	01H	09H	00H	28H	A0H	00H	00H	00H	01H	00H	00H	00H	10H	03H	44H	33H

The feedback data string from the Slave Station #1

DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
			L	H				H	L
10H	06H	01H	01H	00H	00H	10H	03H	30H	32H

• Please follow the procedures below to operate the test:

① Edit the project of Slave PLC #1
Use the Ladder Master S to set the CP1's parameters of Slave PLC #1 and compile relevant program. Then, connect to the USB programming port of Slave #1 and write the project into the PLC.



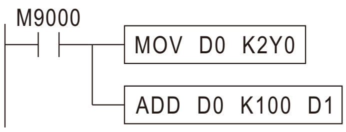
Set the application type: VS Computer Link Slave

Set the baud rate: 19200 bps

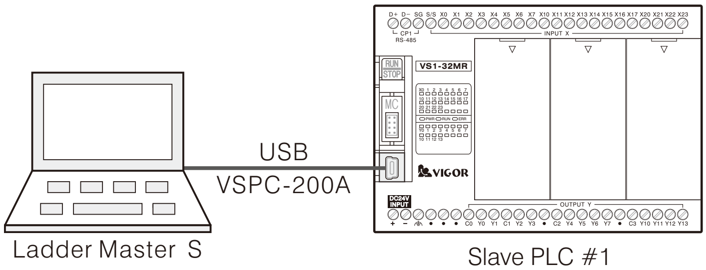
Set the station number: #1

※ Usually the default delay response time is not necessary to change, except that:
When the communication mistake is caused by the signal collision in the circuit, try to extend the delay response time.

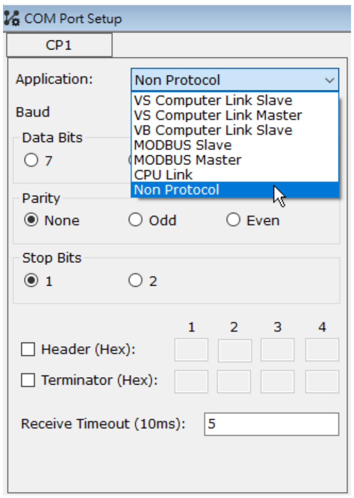
The program writes to the Slave PLC #1



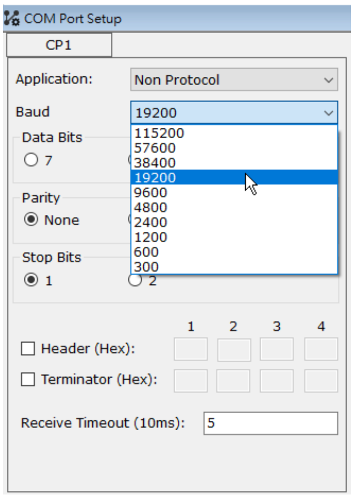
By the USB programming port, the project is written into the Slave PLC #1



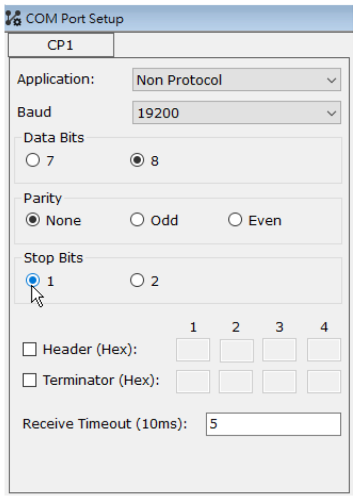
② Edit the project of the Master PLC (by the Non Protocol and the RS instruction)
Use the Ladder Master S to set the CP1's parameters of Master PLC. Compile the RS instruction and relevant program. Then, connect to the USB programming port of Master and write the project into the PLC.



Set the application type:
Non Protocol

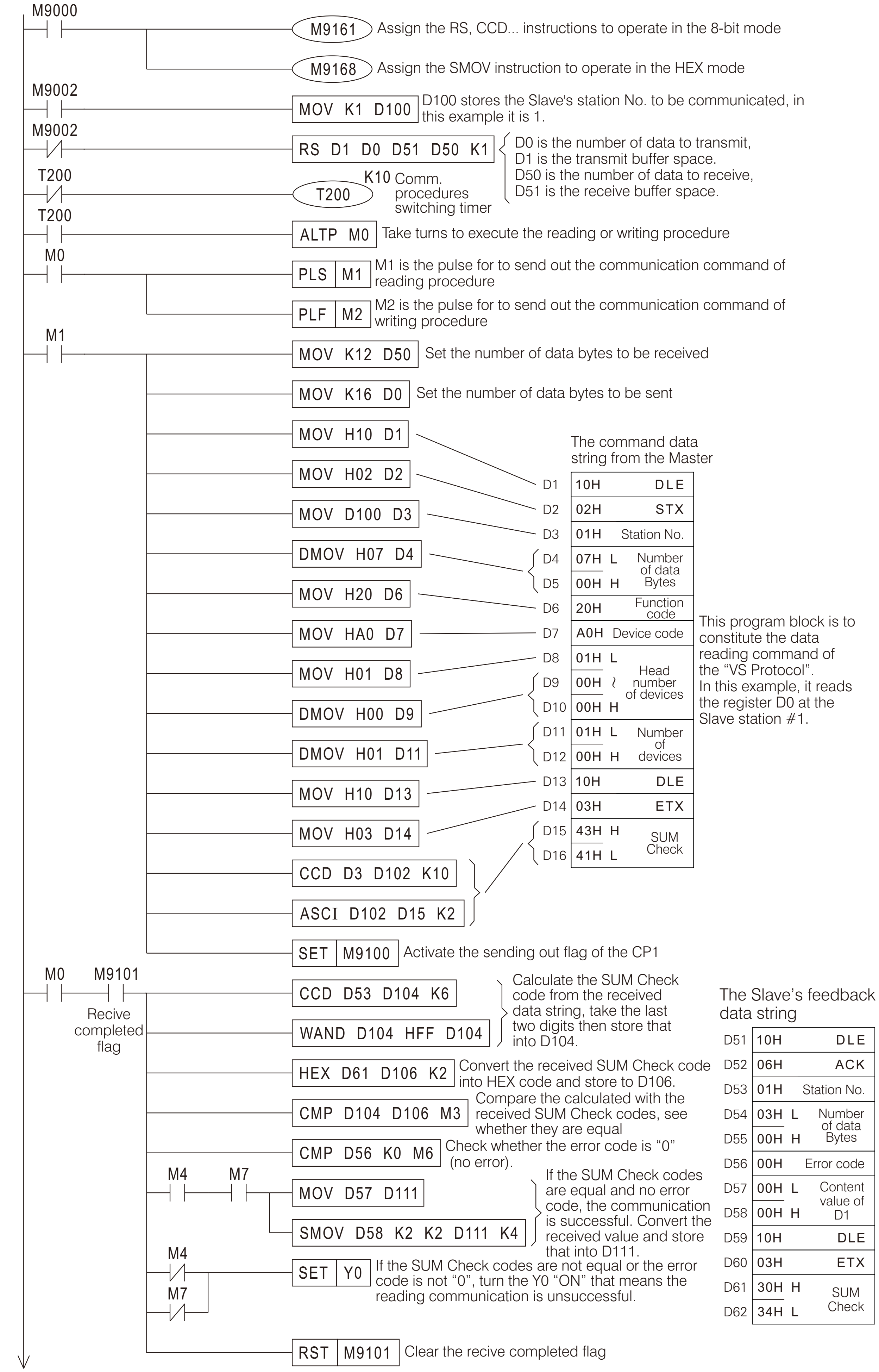


Set the baud rate to be
19200 bps.

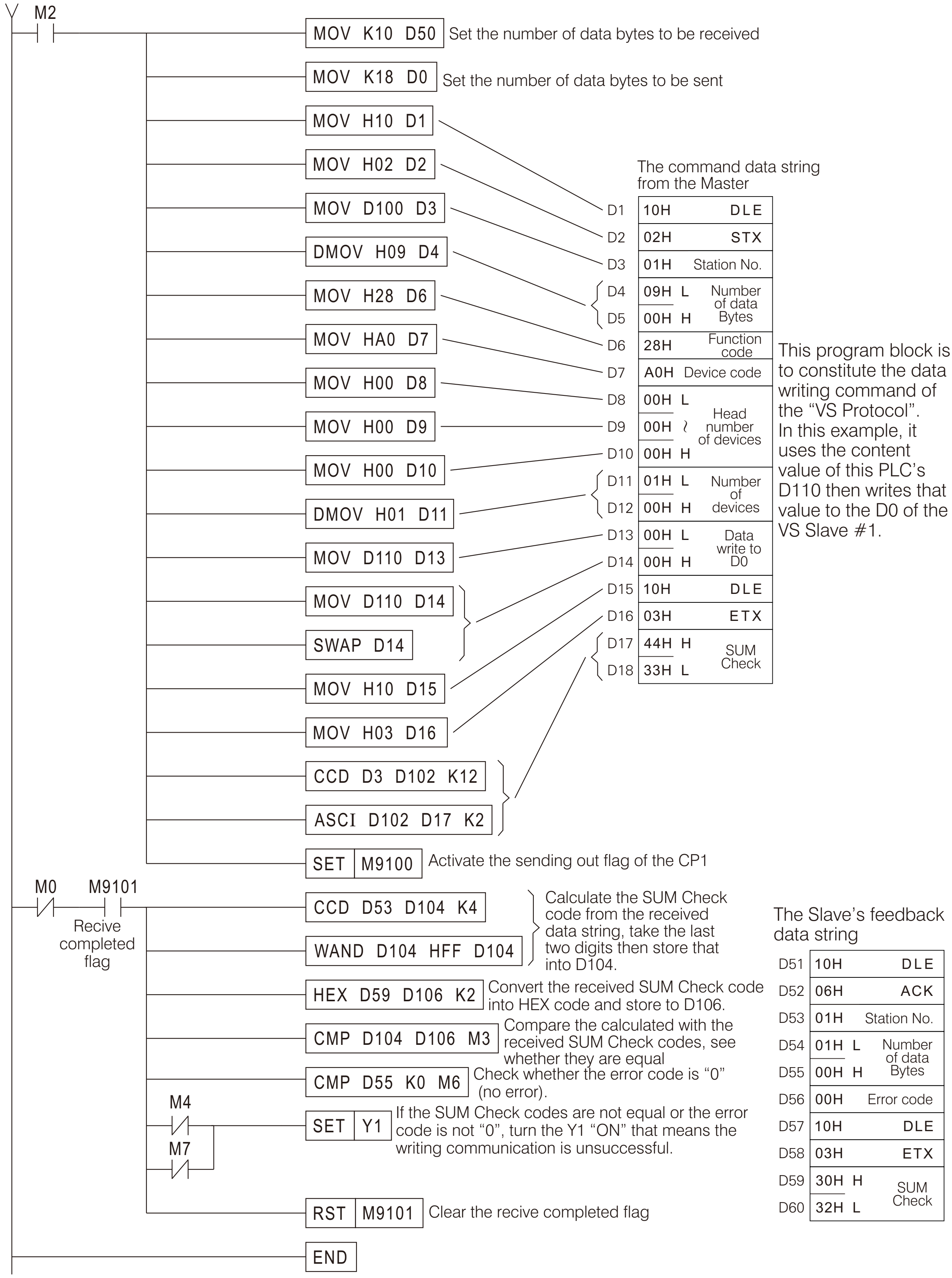


According to the VS Protocol to set the parameters at the sheet of Non Protocol (8, None, 1).

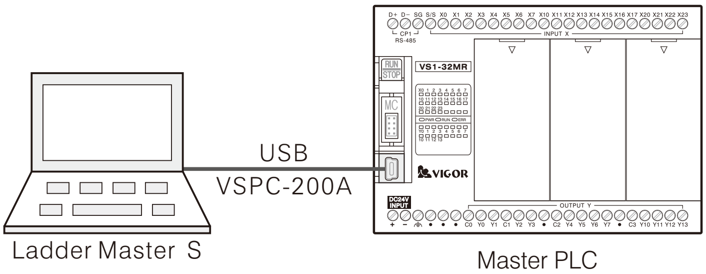
Edit the program for the left PLC (Master).



Continue on next page


M2
MOV K10 D50
Set the number of data bytes to be received
MOV K18 D0
Set the number of data bytes to be sent
MOV H10 D1
MOV H02 D2
MOV D100 D3
DMOV H09 D4
MOV H28 D6
MOV HA0 D7
MOV H00 D8
MOV H00 D9
MOV H00 D10
DMOV H01 D11
MOV D110 D13
MOV D110 D14
SWAP D14
MOV H10 D15
MOV H03 D16
CCD D3 D102 K12
ASCI D102 D17 K2
SET M9100
Activate the sending out flag of the CP1
The command data string from the Master
D1 10H DLE
D2 02H STX
D3 01H Station No.
D4 09H L
D5 00H H
Number of data Bytes
D6 28H Function code
D7 A0H Device code
D8 00H L
D9 00H
D10 00H H
Head number of devices
D11 01H L
D12 00H H
Number of devices
D13 00H L
D14 00H H
Data write to D0
D15 10H DLE
D16 03H ETX
D17 44H H
D18 33H L
SUM Check
This program block is to constitute the data writing command of the “VS Protocol”. In this example, it uses the content value of this PLC’s D110 then writes that value to the D0 of the VS Slave #1.
M0
M9101
Recive completed flag
CCD D53 D104 K4
WAND D104 HFF D104
Calculate the SUM Check code from the received data string, take the last two digits then store that into D104.
HEX D59 D106 K2
Convert the received SUM Check code into HEX code and store to D106.
CMP D104 D106 M3
Compare the calculated with the received SUM Check codes, see whether they are equal
CMP D55 K0 M6
Check whether the error code is “0” (no error).
M4
M7
SET Y1
If the SUM Check codes are not equal or the error code is not “0”, turn the Y1 “ON” that means the writing communication is unsuccessful.
RST M9101
Clear the recive completed flag
END
The Slave’s feedback data string
D51 10H DLE
D52 06H ACK
D53 01H Station No.
D54 01H L
D55 00H H
Number of data Bytes
D56 00H Error code
D57 10H DLE
D58 03H ETX
D59 30H H
D60 32H L
SUM Check

By the USB programming port, the project is written into the Master PLC.



③ After sequentially edit and load the projects to those PLCs, do the wiring and testing jobs between two PLC's.

④ When the example is performing, the data at every PLC will follow its own program and the communication table to execute the operation below.
The left PLC (Master): According to the format of the "VS Protocol" to edit the program using the RS instruction.
The PLC Master reads the value of D1 at the right side Slave #1 via the communication and stores the value to the Master's D111.
Also, writes the content value of the Master's D110 to the D0 of Slave #1.
The right PLC (Slave #1): The content value of D0 is sent to its output points Y0~Y7.
Also, its content value of D0 will be added 100 up and stored to the D1.

Since the left PLC (Master) is based on the format of the "VS Protocol" to edit the program with the RS instruction, that can use the communication to access the data at the Slave #1. The operation produces the following results.

1. The content value of the Master's D110 is output to the points Y0~Y7 of the Slave #1.
2. The content value of the Master's D111 equals to the sum of the content value of its D110 and 100.

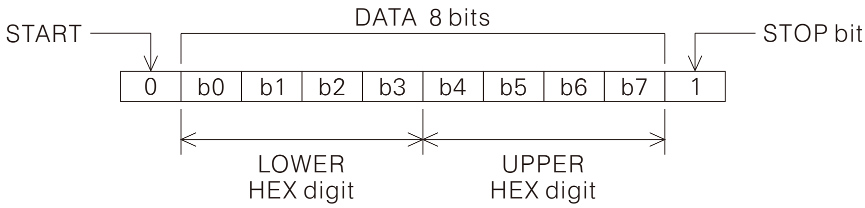
Try to change the content value of the left Master PLC's D110 and observe the changes of the content value of D111, also the output points Y0~Y7 at the right PLC Slave #1 are related to the value.

7-4 VS Series PLC Communication Protocol

A. The relevant communication parameters

- Bits-per-character: 8 bits
  Parity check: None
  Stop bit: 1 bit
  Baud rate: 300/600/1200/2400/4800/9600/19200/34800/57600/115200 bps. selectable (default: 19200 bps.)

- Syntax of a communication character



- This communication protocol adopted the mixed method of ASCII and the HEX code to transmit data. This protocol adopts a few ASCII codes, the conversion table below shows those characters and the corresponding ASCII codes.

Character	ASCII Code
STX	02H
ETX	03H
ACK	06H
DLE	10H

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:
  The available station number is between 0~254 (default: 0). If the communication command uses the station number 255, that is a broadcast command.

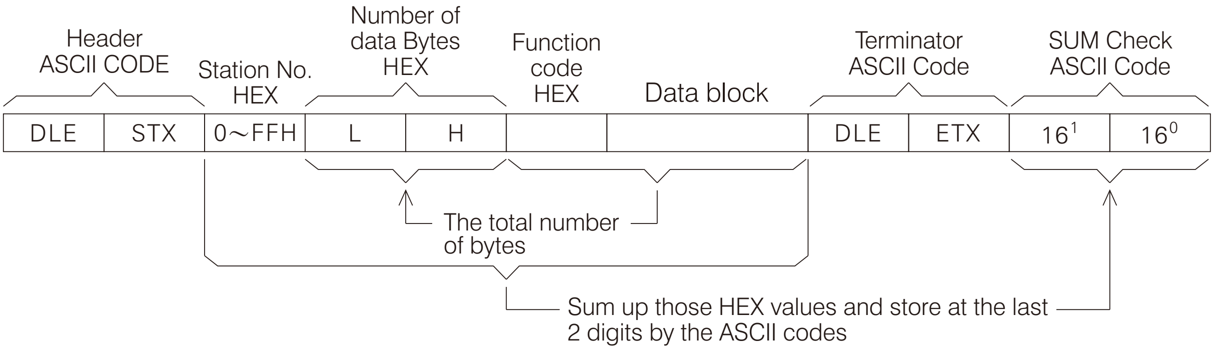
- Error code: When data string is feedback from a PLC, will include with an error code. The table lists the meaning of every error code.

Error Code	Description
00H	Communication is normal; no error
02H	Communication SUM Check Error
04H	The number of data bytes or the number of components is 0

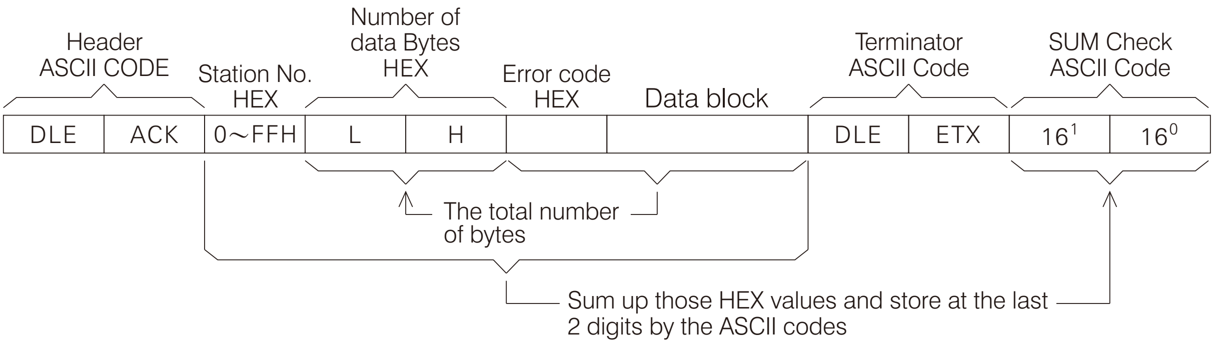
Error Code	Description
06H	The number of data bits exceeds the range
08H	Error ASCII conversion
31H	The command / function code is not existed

B. Communication protocol data format

- The communication command string is sent to the Slave PLC



- The communication feedback string from the Slave PLC



- Header: The starting characters of the data string to be transferred.
  The characters of the header for command string sent to the Slave is DLE(10H) + STX(02H) by the ASCII code; the header of feedback string from the Slave is DLE(10H) + ACK(06H) by the ASCII code.

- Station No.: The Slave's identification number.
  Every Slave PLC at the same communication circuit must have an unique station number, the Slave will respond to the command if its station number matches with the number in the command. Thus, the Master (computer) can use the station number to make a command for the particular Slave PLC.

- Number of data Bytes: Counting the number of bytes from the function or error code to the end of the data block.

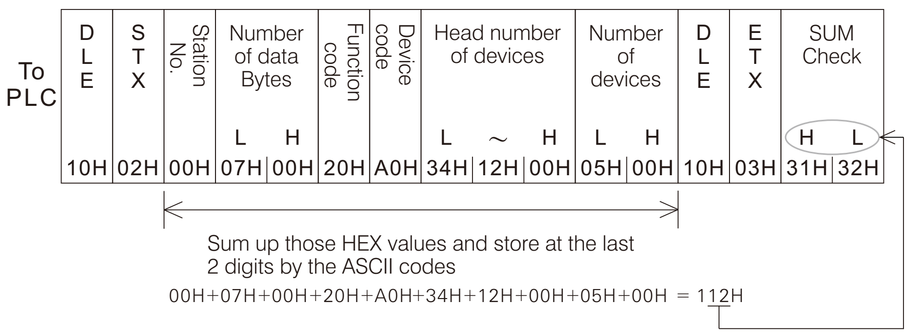
- Function code: The Master PLC or computer commands the Slave PLC to do the appointed task.

Command	Function Code	Object of Device
Word Device Read	20H	D,SD,R,T,C
Word Device Write	28H	D,SD,R,T,C
Bit Device Read	21H	the coil or contact of X, Y, M, SM, S, T and C
Bit Device Write	29H	the coil or contact of X, Y, M, SM, S, T and C

- Data block: The contents in the data block may include the head number of devices, the number of devices, the content values of sending data and so forth.

- Terminator: The last characters of the data string to be transferred.
  The characters of the terminator for command string is DLE(10H) + ETX(03H) by the ASCII code.

- SUM Check: Sum up the content values of HEX data from the beginning of station No. to the end of the data block. Use the accumulated last two digits (by the HEX format) and convert that into two ASCII codes to be the check code.
  To operate the same code-checking procedures at both data sending and receiving ends can ensure the data transmission is correct.

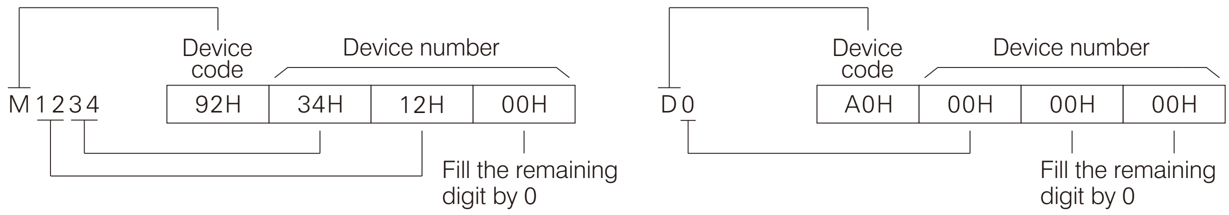


C. Statement of the Device code

Mapping the head device's "Device code" is often needed for the data block access, following describes the coding rules.
The Device code takes four Bytes. The first Byte represents the device type and the second to fourth Bytes represent the number of the device.

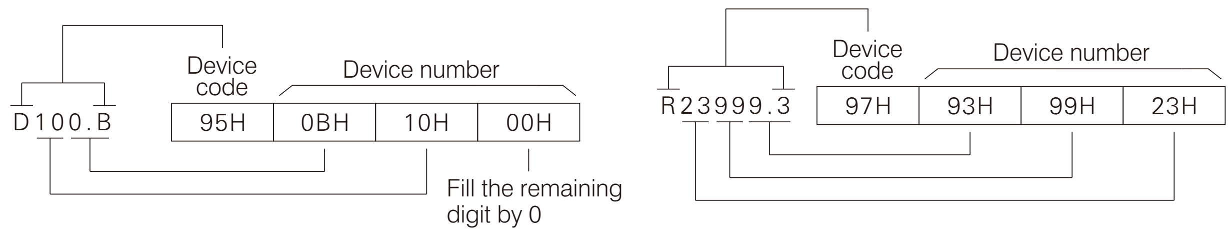
- General the bit devices and word devices follow the coding rule below:

The first Byte	The second Byte		The third Byte		The fourth Byte	
Device code	tens	ones	thousands	hundreds	hundred thousands	ten thousand



- The bit at a register (D.b or R.b) follows the coding rule below:

The first Byte	The second Byte		The third Byte		The fourth Byte	
Device code	ones	bit	hundreds	tens	ten thousand	thousands



Device	Device ID	Device Code	Range of Device No.	Examples of Device Code	
				Device ID	Code L→H
External Input X	X0~X377(Octal)	90H	0~377	X15	90H,15H,00H,00H
External Output Y	Y0~Y377(Octal)	91H	0~377	Y123	91H,23H,01H,00H
Auxiliary Relay M	M0~M8191	92H	0~8191	M1234	92H,34H,12H,00H
Step Relay S	S0~S4095	93H	0~4095	S100	93H,00H,01H,00H
Special Relay M	M9000~M9511	94H	0~511	M9012	94H,12H,00H,00H
Register D's Bit D.b	D0~D8999 at D.0~D.F	95H	0~8999 0~F	D123.F	95H,3FH,12H,00H
Register R's Bit R.b	R0~R23999 at R.0~R.F	97H	0~23999 0~F	R23999.3	97H,93H,99H,23H
Coil of a timer T	T0~T511	98H	0~511	T25	98H,25H,00H,00H
Contact of a timer T	T0~T511	99H	0~511	T123	99H,23H,01H,00H
Coil of a counter C	C0~C255	9CH	0~255	C0	9CH,00H,00H,00H
Contact of a counter C	C0~C255	9DH	0~255	C200	9DH,00H,02H,00H
Register D (content value)	D0~D8999	A0H	0~8999	D1000	A0H,00H,10H,00H
Special Register SD (content value)	D9000~D9511	A1H	0~511	D9001	A1H,01H,00H,00H
Register R (content value)	R0~R23999	A2H	0~23999	R12345	A2H,45H,23H,01H
Timer T (Present Value)	T0~T511	A8H	0~511	T255	A8H,55H,02H,00H
16-bit Counter C (Present valur)	C0~C199	ACH	0~199	C0	ACH,00H,00H,00H
32-bit Counter C (Present valur)	C200~C255	ADH	200~255	C235	ADH,35H,02H,00H

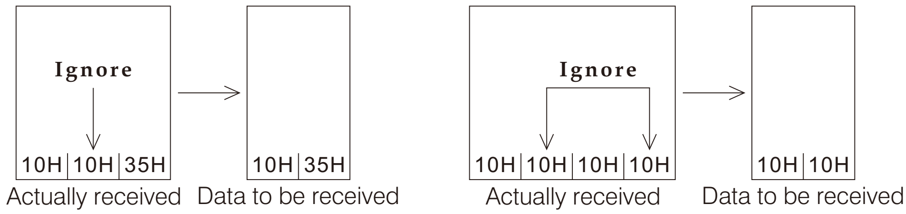
D. Statement of the communication command

- The sending equipment will bind the codes together that includes the header, terminator and check code in a communication command string. Except those three special purpose codes, if the content data include with the code 10H in the string, the code 10H should be repeated once.

  For example:

- When a receiving equipment gets two consecutive 10H codes, the second 10H code needs to be ignored.

  For example:



• Function code 20H: to read word devices (up to 64 words can be read at one command).

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
Command to PLC	10H	02H	00H ~ FFH	L	H	20H	XXH	L	~	H	L	H	10H	03H	H	L

	DLE	ACK	Station No.	Number of data Bytes		Error code	Data block			Data block		DLE	ETX	SUM Check	
Feedback from PLC (complete)	10H	06H	As above	L	H	00H	L	H	. . . . .	L	H	10H	03H	H	L

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
Feedback from PLC (error)	10H	06H	As above	01H	00H	XXH	10H	03H	H	L

Example 1: Read the content values of D1234~D1238 at the Salve #0. (five 16-bit word devices)
Suppose D1234 = 89ABH, D1235 = 1000H, D1236 = 2345H, D1237 = 0H and D1238 = 3FH at the Salve #0.

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
				L	H			L	~	H	L	H			H	L
Command to PLC	10H	02H	00H	07H	00H	20H	A0H	34H	12H	00H	05H	00H	10H	03H	31H	32H

	DLE	ACK	Station No.	Number of data Bytes		Error code	Value at D1234		Value at D1235		Value at D1236		Value at D1237		Value at D1238		DLE	ETX	SUM Check	
				L	H		L	H	L	H	L	H	L	H	L	H			H	L
Feedback from PLC (complete)	10H	06H	00H	0BH	00H	00H	ABH	89H	00H	10H	45H	23H	00H	00H	3FH	00H	10H	03H	46H	36H

Example 2: Read the Present Values of C235 and C236 at the Salve #0. (two 32-bit double word devices)
Suppose C235 = 236B9H and C236 = 11253648H at the Salve #0.
Since the C235 and C236 are 32-bit devices, each Present Value will be consisted of two words (Low word and High word), so the data for these C235 and C236 will need 4 words.

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
				L	H			L	~	H	L	H			H	L
Command to PLC	10H	02H	00H	07H	00H	20H	ADH	35H	02H	00H	02H	00H	10H	03H	30H	44H

	DLE	ACK	Station No.	Number of data Bytes		Error code	Present Value at C235				Present Value at C236				DLE	ETX	SUM Check	
				L	H		LL	LH	HL	HH	LL	LH	HL	HH			H	L
Feedback from PLC (complete)	10H	06H	00H	09H	00H	00H	B9H	36H	02H	00H	48H	36H	25H	11H	10H	03H	41H	45H

• Function code 21H: to read bit devices (up to 1024 bits can be read at one command).

Command to PLC

DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
10H	02H	00H ~ FFH	L	H	21H	XXH	L	~	H	L	H	10H	03H	H	L

Feedback from PLC (complete)

DLE	ACK	Station No.	Number of data Bytes		Error code	Data block (Transmit unit: Byte) ·····					DLE	ETX	SUM Check	
10H	06H	As above	L	H	00H	B0	B1			Bn	10H	03H	H	L

Feedback from PLC (error)

DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
10H	06H	As above	01H	00H	XXH	10H	03H	H	L

Example 1: Read the states of M10~M63 at the Salve #0 (total 54 bit devices)
Suppose the states of M10~M63 at the Salve #0 are as below: (composed to Byte values)

8 bits

55H	0	1	0	1	0	1	0	1	M10(B0)
AAH	1	0	1	0	1	0	1	0	M18(B1)
00H	0	0	0	0	0	0	0	0	M26(B2)
FFH	1	1	1	1	1	1	1	1	M34(B3)
ABH	1	0	1	0	1	0	1	1	M42(B4)
12H	0	0	0	1	0	0	1	0	M50(B5)
2EH	0	0	1	0	1	1	1	0	M58(B6)

M63

Fill the unused devices as "0"

※ 54 bits in total, 54 = 36H (the number of devices)

Command to PLC

DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		DLE	ETX	SUM Check	
			L	H			L	~	H	L	H			H	L
10H	02H	00H	07H	00H	21H	92H	10H	00H	00H	36H	00H	10H	03H	30H	30H

Feedback from PLC (complete)

DLE	ACK	Station No.	Number of data Bytes		Error code	Status (by Byte values) at M10~M63							DLE	ETX	SUM Check	
			L	H		B0	B1	B2	B3	B4	B5	B6			H	L
10H	06H	00H	08H	00H	00H	55H	AAH	00H	FFH	ABH	12H	2EH	10H	03H	46H	31H

• Function code 28H: to write word devices (up to 64 words can be write at one command).

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		Data to be written		.....	Data to be written		DLE	ETX	SUM Check	
Command to PLC	10H	02H	00H ~ FFH	L	H	28H	XXH	L	~	H	L	H	L	H		L	H	10H	03H	H	L

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
Feedback from PLC (complete)	10H	06H	As above	01H	00H	00H	10H	03H	H	L
Feedback from PLC (error)	10H	06H	As above	01H	00H	XXH	10H	03H	H	L

The error code in feedback string at a successful communication is 00H; but if the error code is not 00H, that means error happened (refer to the list at the beginning of this section).

Example 1: Write the values 2048H and 300H into the D7000 and D7001 at the Salve #0 (two 16-bit word devices)

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		Data write to D7000		Data write to D7001		DLE	ETX	SUM Check	
				L	H			L	~	H	L	H	L	H	L	H			H	L
Command to PLC	10H	02H	00H	0BH	00H	28H	A0H	00H	70H	00H	02H	00H	48H	20H	00H	03H	10H	03H	42H	30H

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
				L	H				H	L
Feedback from PLC (complete)	10H	06H	00H	01H	00H	00H	10H	03H	30H	31H

Example 2: Write the value 89AB1234H into the C210 at the Salve #0 (a 32-bit double word device)

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		Data write to C210				DLE	ETX	SUM Check	
				L	H			L	~	H	L	H	LL	LH	HL	HH			H	L
Command to PLC	10H	02H	00H	0BH	00H	28H	ADH	10H	02H	00H	01H	00H	34H	12H	ABH	89H	10H	03H	36H	44H

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
				L	H				H	L
Feedback from PLC (complete)	10H	06H	00H	01H	00H	00H	10H	03H	30H	31H

• Function code 29H: to write bit devices (up to 1024 bits can be write at one command).

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		Data block (Transmit unit: Byte)						DLE	ETX	SUM Check	
Command to PLC			00H ~												·····							
	10H	02H	FFH	L	H	29H	XXH	L	~	H	L	H	B0	B1				Bn	10H	03H	H	L

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
Feedback from PLC (complete)	10H	06H	As above	01H	00H	00H	10H	03H	H	L

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
Feedback from PLC (error)	10H	06H	As above	01H	00H	XXH	10H	03H	H	L

The error code in feedback string at a successful communication is 00H; but if the error code is not 00H, that means error happened (refer to the list at the beginning of this section).

Example 1: Write the status below into the M100~M127 at the Salve #0 (total 28 bit devices)

8 bits

EFH	1	1	1	0	1	1	1	1	M100(B0)
CDH	1	1	0	0	1	1	0	1	M108(B1)
35H	0	0	1	1	0	1	0	1	M116(B2)
00H	0	0	0	0	0	0	0	0	M124(B3)

Fill the unused devices as "0"

M127

※ 28 bits in total, 28 = 1CH (the number of devices)

	DLE	STX	Station No.	Number of data Bytes		Function code	Device code	Head number of devices			Number of devices		Data write to M100~M127				DLE	ETX	SUM Check	
Command to PLC				L	H			L	~	H	L	H	B0	B1	B2	B3			H	L
	10H	02H	00H	0BH	00H	29H	92H	00H	01H	00H	1CH	00H	EFH	CDH	35H	00H	10H	03H	44H	34H

	DLE	ACK	Station No.	Number of data Bytes		Error code	DLE	ETX	SUM Check	
Feedback from PLC (complete)				L	H				H	L
	10H	06H	00H	01H	00H	00H	10H	03H	30H	31H

MEMO

8. Statement of Positioning Control Functions

The VS series PLC Main Unit provided with four high-speed pulse output points, and thus can operate four axis controls. In the multi-axis positioning control applications, it is very suitable and effective.

The pulse output frequency of the VS1 and VS2 series transistor PLCs can reach up to 50 kHz, can be used to complete various of basic positioning control.

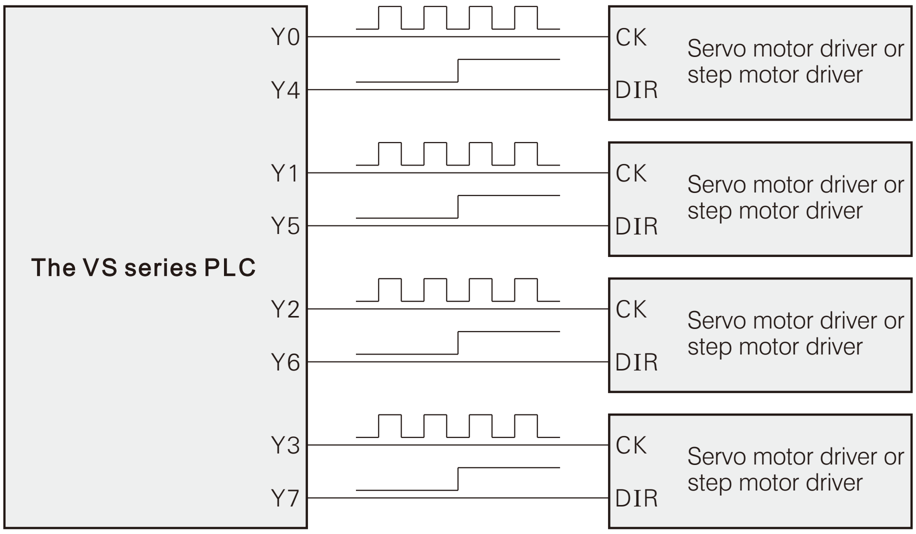
The VSM Motion Control Main Unit is equipped with quick reaction high-speed I/O points, because it is special design for positioning control applications with complete positioning functions.

The VS3 High Performance Main Unit can satisfy the most diverse and the highest level of applications, the transistor model also possesses complete positioning control functions.

The pulse output frequency of the VSM and VS3 series PLCs can reach to 200 kHz, furthermore the VSM-28L can even reach to 1 MHz. All the VS series PLC can support 17 positioning instructions of complete functions and thus can complete various types of exquisite and smooth positioning control.

When using the positioning control functions of the VS series PLC Main Units, some common precautions are noted below:

- The VS1 and VS2 series transistor models are provided with four 50 kHz high-speed pulse outputs, the VSM and VS3 transistor models are provided with four 200 kHz high-speed pulse outputs. Moreover, the VSM-28L provides four 1 MHz high-speed line-driver output circuits.
- There is no limitation on the number of times that positioning instructions are wrote at the user program. However, must regard to that do not use two or more instructions to drive the same output point at the same time.
- The load voltage of transistor output points is DC5V~24V, and for the general output uses the current capacity is 0~0.5 A; but when a point Y0~Y3 is assigned for the high-speed pulse function, its current is 0~100 mA.
- About the positioning control method at the VS series PLC, its controlling output form is “Pulse train + Direction signal”. The output points for the direction signals can use Y0~Y7 or internal auxiliary relays M. Nonetheless, those are better using Y0~Y7 to minimize output delay.



The VS series PLC's positioning control functions are achieved by the positioning instructions, parameters (by the Ladder Master S to configure) and relevant special devices, the following sections will introduce those respectively.

8-1 Positioning Parameter Setup

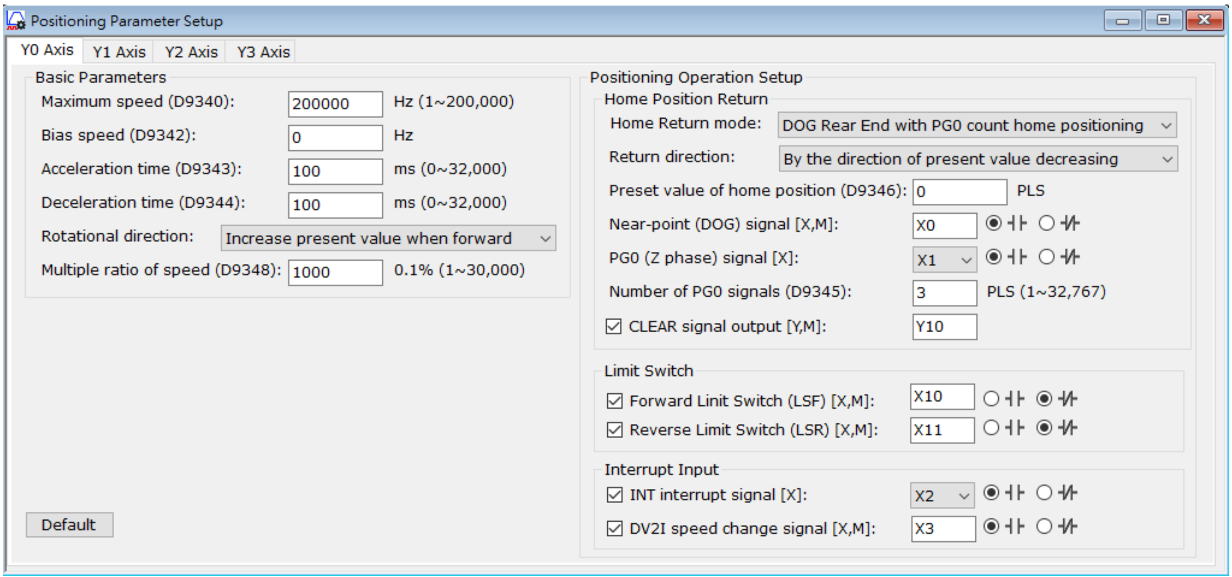
The Ladder Master S provides the Positioning Parameter Setup function to simplify the VS series PLC's process of setting relevant parameters. The contents of positioning parameters will be written onto the PLC with the project. After that, only the appropriate positioning control instructions are needed to complete the positioning control work easily.

In the most of cases, the Positioning Parameter Setup function can satisfy application requirements.

However, during the operation, some specific applications may need to change related parameters. Thus, a part of the positioning parameters are not only preset by the setting of the parameter setup page, but also can make changes at relevant special registers via the user program.

When PLC's power is turned from "OFF" to "ON" and the project had been written into the PLC, the corresponding parameters will be copied to relevant special registers. After that, if there is a demand to change positioning parameters during PLC is in operation, the user just needs to change the values in relevant special registers before to execute the positioning control instruction. Then, the positioning control instruction will base on the changed values in the special registers to execute the instruction. For example, to change the time of acceleration or deceleration.

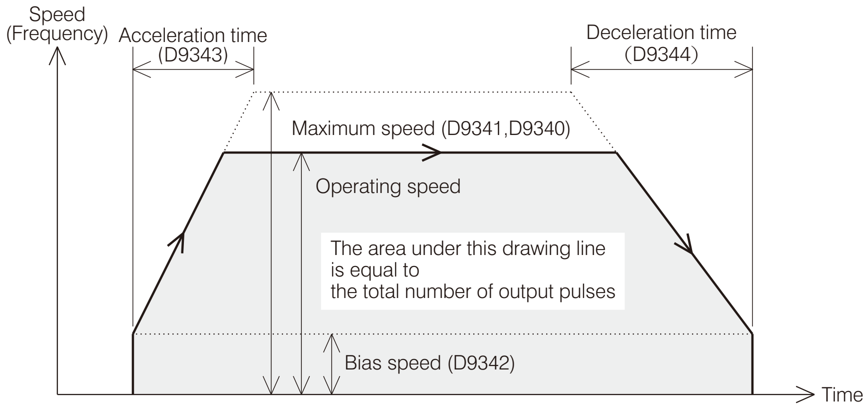
To open the setup page, select the option "Positioning Parameter Setup" under the "Project" of the Ladder Master S. Hereby we illustrate the setting for each item on the page one by one.



Press the "Default" button, that will reset the set contents to the default values about this axis.

8-1-1 Basic Parameters

The VS series PLC provides the positioning control with the acceleration and deceleration functions, as shown in the following diagram. It is necessary to set relevant parameters before to use the positioning control instructions, thus can perform the operation correctly. The parameters in the diagram below uses Y0 output axis as an example.



- Maximum speed

  This value confines the highest limit of the positioning control speed at a certain axis. If its operation speed exceeds the limit of the maximum speed during the action of any positioning control instruction, the instruction will be operated according to the maximum speed.
  The highest output frequency of the VS1 or VS2 series is 50 kHz. The acceptable value range is from 1 to 50 k (Hz). Any value less than 1 is regarded as 1 Hz; more than 50 k is regarded as 50 kHz. The default value is 50 kHz.
  The highest output frequency of the VSM or VS3 series is 200 kHz. The acceptable value range is from 1 to 200 k (Hz). Any value less than 1 is regarded as 1 Hz; more than 200 k is regarded as 200 kHz. The default value is 200 kHz.
  The highest output frequency of the VSM-28ML series is 1 MHz. The acceptable value range is from 1 to 1 M (Hz). Any value less than 1 is regarded as 1 Hz; more than 1 M is regarded as 1 MHz. The default value is 1 MHz.

- Bias speed

  This value confines the lowest limit of the positioning control speed at a certain axis. The main purpose is to avoid the low-frequency resonance area of a step motor. Thus, it is usually set to be 0 for a servo motor.
  If the operation speed is lower than the bias speed during the action of any positioning control instruction, the instruction will be operated according to the bias speed.
  The acceptable value range is from 0 to 20 k (Hz). Any value less than 0 is regarded as 0 Hz; more than 20 k is regarded as 20 kHz. The default value is 0 Hz.

- Acceleration timey

  As shown in the diagram, the acceleration time refers to the time it takes for speeding up from the bias speed to the maximum speed (not the operating speed).
  The acceptable value range is from 0 to 32,000 (ms). Any value less than 0 is regarded as 0 ms; more than 32,000 is regarded as 32,000 ms. The default value is 100 ms.

- Deceleration time

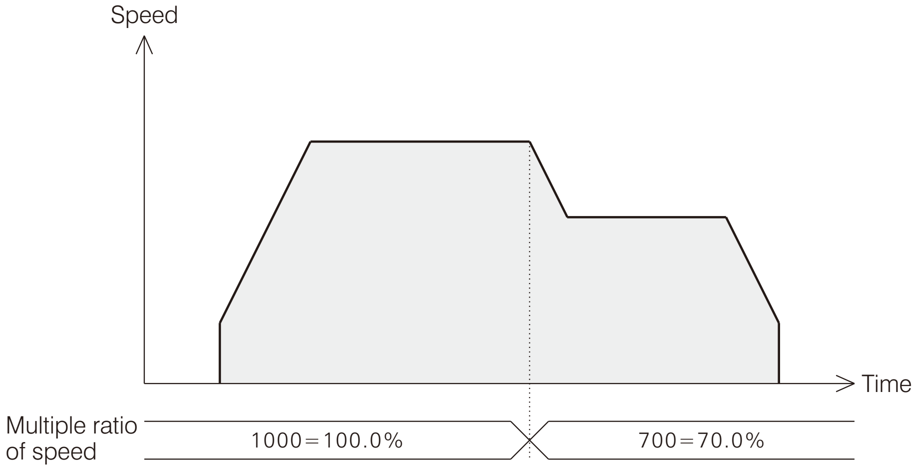
  As shown in the diagram, the deceleration time refers to the time it takes for slowing down from the maximum speed (not the operating speed) to the bias speed.
  The acceptable value range is from 0 to 32,000 (ms). Any value less than 0 is regarded as 0 ms; more than 32,000 is regarded as 32,000 ms. The default value is 100 ms.

- Rotational direction

  Users can select the direction control pattern: “Increase Present Value when forward” or “Increase Present Value when backward”. That will affect to the direction output of the positioning control. The default is “Increase Present Value when forward”.
  If the “Increase Present Value when forward” is selected and the positioning instruction decides to increase its present value, then the direction control point will turn “ON” to drive the motor moving forward.
  If the “Increase Present Value when backward” is selected and the positioning instruction decides to increase its Present Value, then the direction control point will turn “OFF” to drive the motor moving backward.

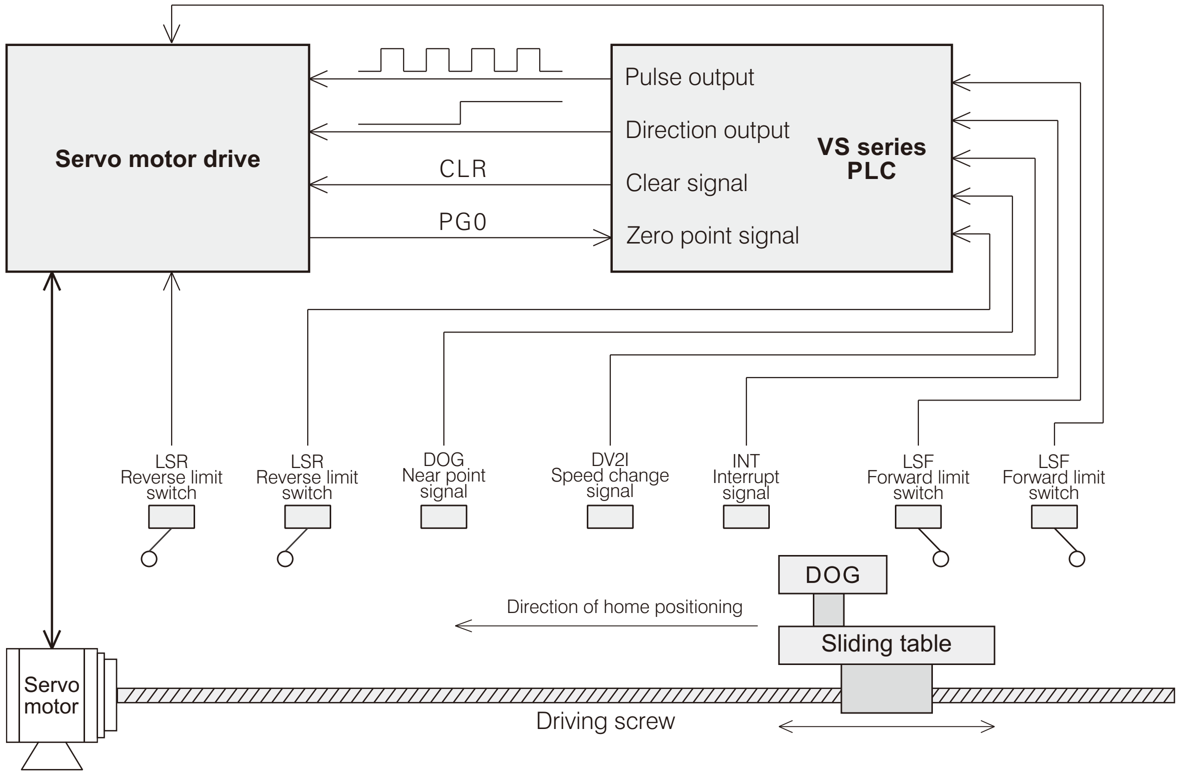
- Multiple ratio of speed:
  This is used to change the operating speed when the positioning control is active. The default value is 1,000 (equal to 100.0%).
  The jog speed, home position return speed, home position return creep speed and operating speed are affected by this parameter.
  If this parameter value is changed during the positioning control instruction is active, the actual operating speed is changed accordingly.

  The available range of the multiple ratio is from 1 to 30,000, in a unit of 0.1% (that means 0.1% ~ 3000.0%).



8-1-2 Positioning Operation Setting Up

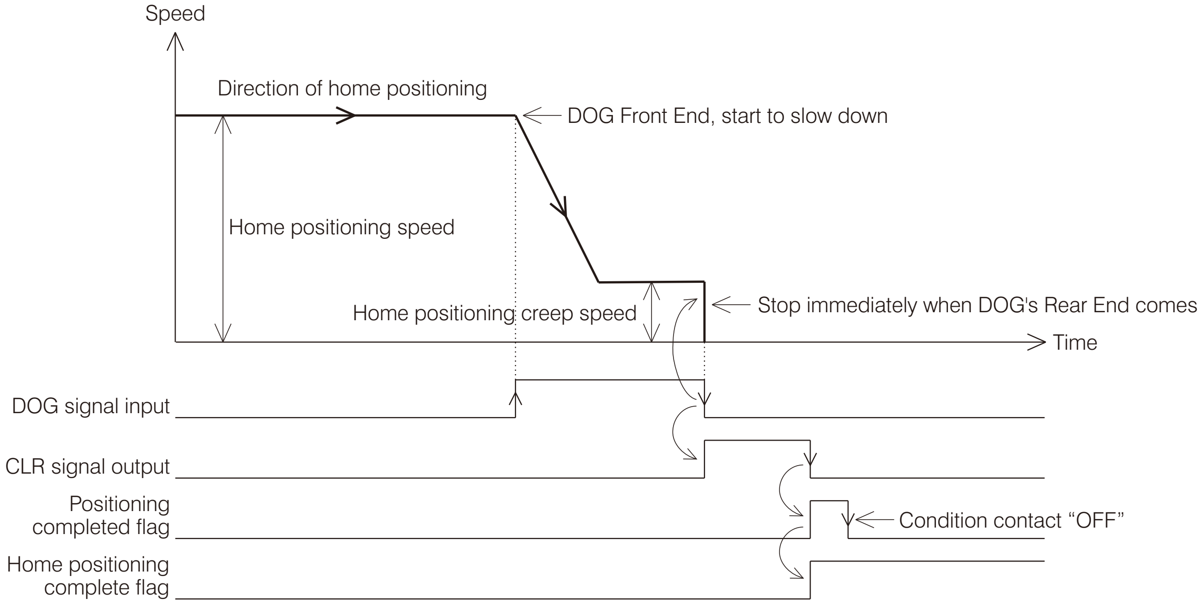
Demonstrated below is the brief configuration of a general positioning control system, which we will use to illustrate the related parameter settings for the positioning control.



- Home Position Return (Zero Return)

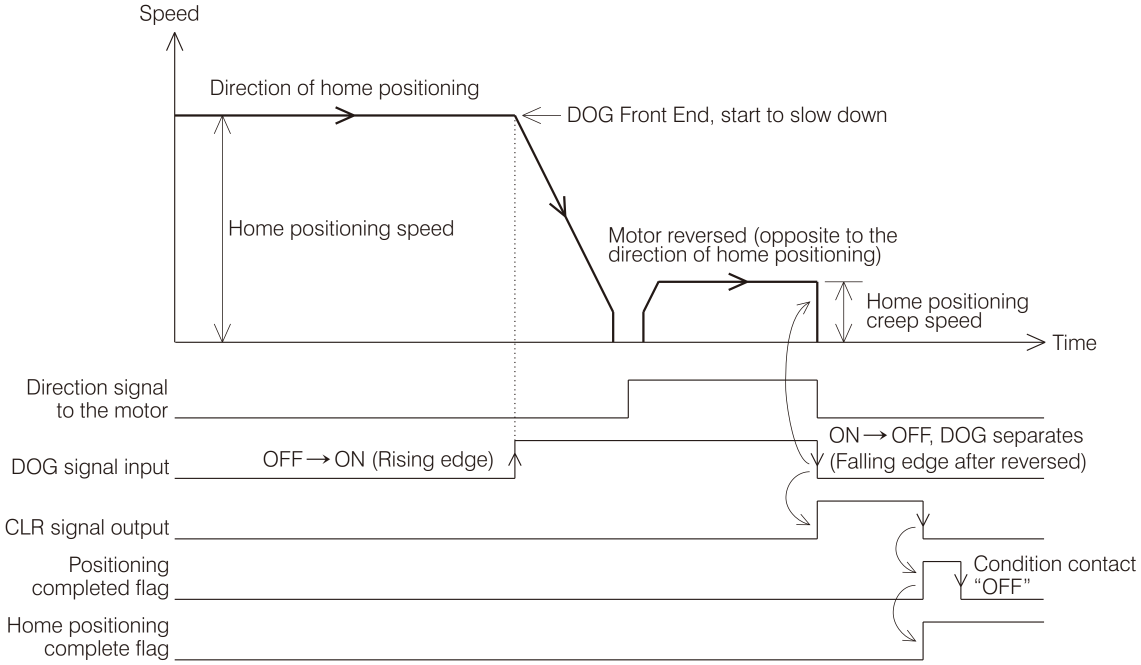
  The VS PLC provides a variety of home position return modes when the ZRN instruction is used, which will be explained one by one below. The parameters in the setup sheet are decided on those different modes. Hence, to easily complete the setting, all the users need to do is to select the appropriate mode, and then fill in the relevant parameters.

  - DOG Rear End home positioning



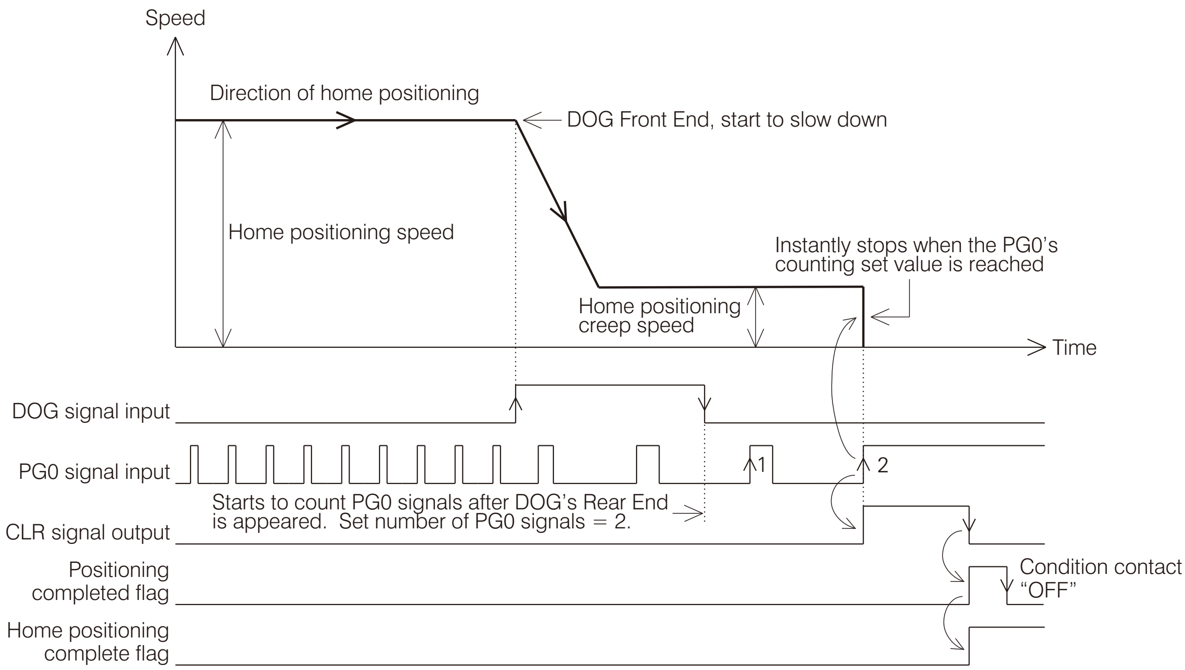
The DOG signal may use a X or M. Usually assigned to the X0~X7, because by the interrupt input point can have the precise home position. On the other hand, if the signal for the DOG is not from X0~X7, the home position after the return will have some inaccurate error.
The CLR signal output can assign to a Y or M, the width of that “ON” signal is ≥ 20ms.

• DOG Front End home positioning



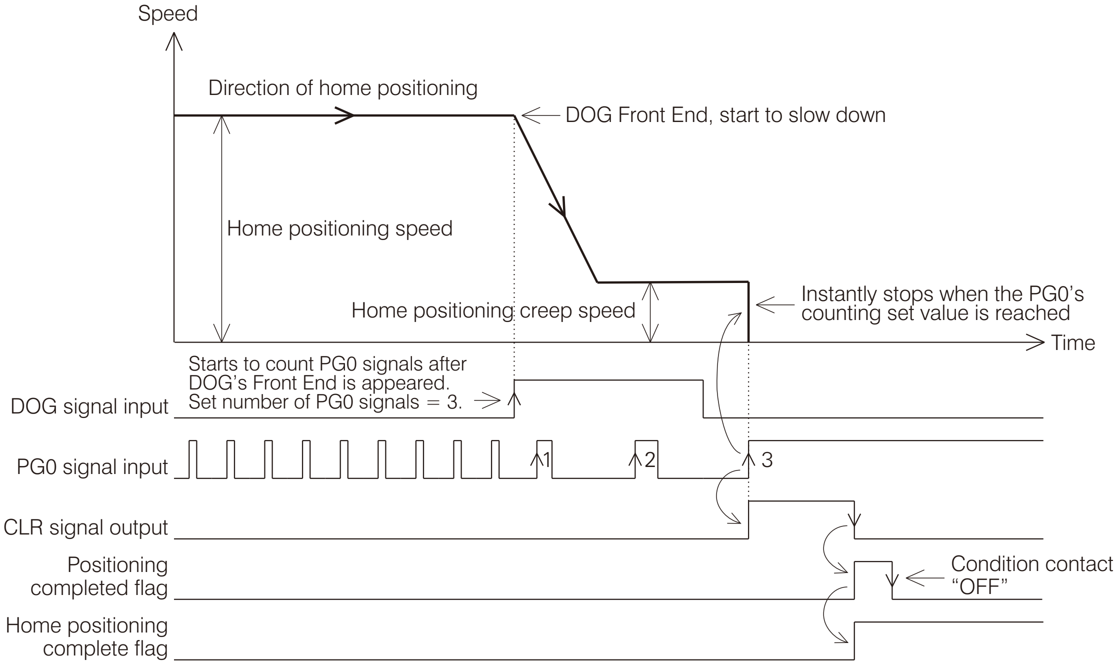
The DOG signal may use a X or M. Usually assigned to the X0~X7, because by the interrupt input point can have the precise home position. On the other hand, if the signal for the DOG is not from X0~X7, the home position after the return will have some inaccurate error.
The CLR signal output can assign to a Y or M, the width of that "ON" signal is ≥ 20ms.

• DOG Rear End with PG0 count home positioning



The DOG signal may use a X or M. Usually assigned to the X0~X7, because the interrupt input point can have the instant response.
Should avoid the DOG's Rear End signal to be too close to the PG0's Front End signal, that may cause the PG0 counting error.
For the PG0 signal input, must use X0~X7. The PLC processes the counting by the way of interrupt.
The CLR signal output can assign to a Y or M, the width of that "ON" signal is ≥ 20ms.

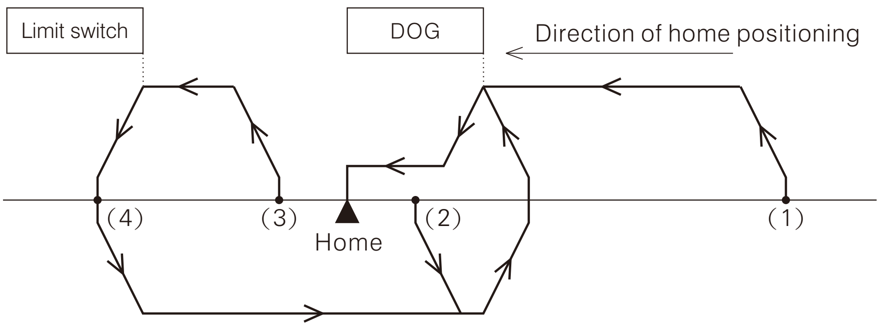
• DOG Front End with PG0 count home positioning



The DOG signal may use a X or M. Usually assigned to the X0~X7, because the interrupt input point can have the instant response.
Should avoid the DOG's Front End signal to be too close to the PG0's Front End signal, that may cause the PG0 counting error.
For the PG0 signal input, must use X0~X7. The PLC processes the counting by the way of interrupt.
The CLR signal output can assign to a Y or M, the width of that "ON" signal is ≥ 20ms.

• Data-set type home return
At this mode, the motor does not rotate because it is no actual moving position. When the function is operated, it will fill the preset value of the home position into the register which stores the Present Value also turn the home positioning complete flag "ON". If the CLR output is activated, the width of the signal pulse is ≥ 20ms.

• Dog search home positioning
When the positioning system has installed with limit switches to provide the limiter signals for the VS PLC, by this mode that will give the automatic search capability for the home positioning. (the examples are using the DOG Rear End home positioning)



The diagram above illustrates the different actions from the starting points (1)~(4) to complete the home positioning.

(1) If the starting point is located on the right of the DOG switch:
The home positioning is moving the sliding table by the home positioning speed and the direction of home positioning. Until the Front End of the DOG is reached, the speed decreases to the creep speed, then to finish the home positioning.

(2) If the starting point had been driving the DOG switch "ON":
The home positioning is moving the sliding table by the home positioning speed and the opposite direction of home positioning. Until the Front End of the DOG is separated (the signal turns from "ON" to "OFF"), the speed slowing down and then stop. After that, it is moving the sliding table by the home positioning speed and the direction of home positioning. Until the Front End of the DOG is reached, the speed decreases to the creep speed, then to finish the home positioning.

(3) If the starting point is located on the right of the DOG switch:
The home positioning is moving the sliding table by the home positioning speed and the direction of home positioning. Until the limit switch is reached, the speed slowing down and then stop. After that, moving the sliding table by the home positioning speed and the opposite direction of home positioning until the Rear End of the DOG is separated (the signal turns from "ON" to "OFF"), the speed slowing down and then stop. Furthermore, it uses the home positioning speed and the direction of home positioning to move the table again until the Rear End of the DOG is reached, the speed decreases to the creep speed then to finish the home positioning.

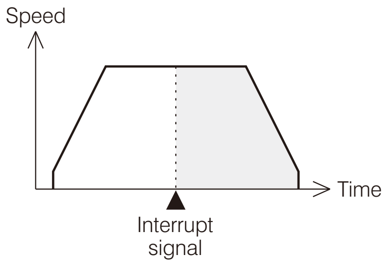
(4) If the starting point has caused the limit switch "ON":
The home positioning is moving the sliding table by the home positioning speed and the opposite direction of home positioning until the Rear End of the DOG is separated (the signal turns from "ON" to "OFF"), the speed slowing down and then stop. Furthermore, it uses the home positioning speed and the direction of home positioning again to move the table until the Front End of the DOG is reached, the speed decreases to the creep speed then to finish the home positioning.

● Limit Switches

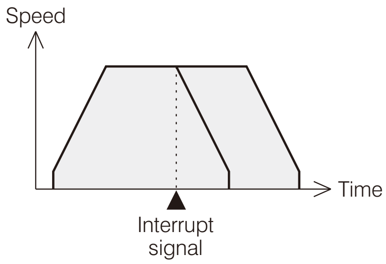
- Forward Limit Switch (LSF)
This is a limit switch for a motor which moves forward. When the forward limit switch is active, that limits all the forward actions. That makes the forward motor to slow down and stop. However, it is not interfering with reversed actions. The imit switch can be assigned to any external input X or internal auxiliary relay M.
- Reverse Limit Switch (LSR)
This is a limit switch for a motor which moves reversed. When the reverse limit switch is active, that limits all the reversed actions. That makes the reversed motor to slow down and stop. However, it is not interfering with forward actions. The imit switch can be assigned to any external input X or internal auxiliary relay M.

● Interrupt Signals for Positioning
Other interruption related positioning control instructions are introduced below:

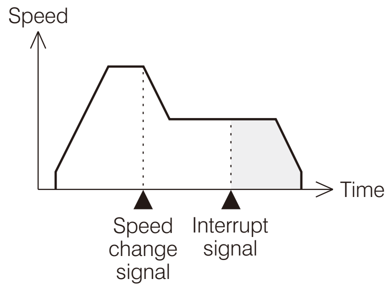
DVIT Interrupt Constant Quantity Positioning



DVSR or DVSA Interrupt to Stop Positioning



DV2I 2 Stages Interrupt Constant Quantity Positioning



- INT interrupt signal
When there is an interrupt signal input, the positioning instructions start pertinent control actions. The interruption signal must have instant response to reach a higher accuracy of positioning. Hence, the interruption signal through the external input X0~X7 is necessary for the PLC to control that in the way of interruption.
- DV2I speed change signal
To operate the DV2I 2 stages interrupt constant quantity positioning instruction, its speed change signal can be assigned to an external input X or internal auxiliary relay M.

8-2 Special Components Related to Positioning Control Instructions

In the tables below, the symbo “ ■ ” represents that the component is not allowed to use an instruction in the program to drive the relay or write data to the register.

Relay ID No.	Description	Series			
Y0 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9340	Y0 axis's status. “OFF” means the Y0 is in the READY status, it is available for a positioning instruction; while “ON” = BUSY, the Y0 has been occupying.	○	○	○	○
■M9341	Y0 axis's pulse output monitor. “ON” means pulse is generating.	○	○	○	○
■M9342	Y0 axis's positioning completed flag.	○	○	○	○
■M9343	Y0 axis's positioning abnormal stop flag.	○	○	○	○
■M9344	Y0 axis's zero home positioning has been completed.	○	○	○	○
M9345	Y0 axis's stop flag (with gradually slow down).	○	○	○	○
M9346	Y0 axis's immediately stop flag.	○	○	○	○
M9347	Y0 axis's table positioning start signal.	○	○	○	○
■M9348	Y0 axis's M-code active flag.	○	○	○	○
M9349	Y0 axis's M-code clear command.	○	○	○	○
M9350	Y0 axis's external interrupt trigger type. When M9350=“OFF”, the interrupt is triggered by a rising edge; when M9350=“ON”, that is triggered by a falling edge.	○	○	○	○
Y1 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9360	Y1 axis's status. “OFF” means the Y1 is in the READY status, it is available for a positioning instruction; while “ON” = BUSY, the Y1 has been occupying.	○	○	○	○
■M9361	Y1 axis's pulse output monitor. “ON” means pulse is generating.	○	○	○	○
■M9362	Y1 axis's positioning completed flag.	○	○	○	○
■M9363	Y1 axis's positioning abnormal stop flag.	○	○	○	○
■M9364	Y1 axis's zero home positioning has been completed.	○	○	○	○
M9365	Y1 axis's stop flag (with gradually slow down).	○	○	○	○
M9366	Y1 axis's immediately stop flag.	○	○	○	○
M9367	Y1 axis's table positioning start signal.	○	○	○	○
■M9368	Y1 axis's M-code active flag.	○	○	○	○
M9369	Y1 axis's M-code clear command.	○	○	○	○
M9370	Y1 axis's external interrupt trigger type. When M9370=“OFF”, the interrupt is triggered by a rising edge; when M9370=“ON”, that is triggered by a falling edge.	○	○	○	○
Y2 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9380	Y2 axis's status. “OFF” means the Y2 is in the READY status, it is available for a positioning instruction; while “ON” = BUSY, the Y2 has been occupying.	○	○	○	○
■M9381	Y2 axis's pulse output monitor. “ON” means pulse is generating.	○	○	○	○
■M9382	Y2 axis's positioning completed flag.	○	○	○	○
■M9383	Y2 axis's positioning abnormal stop flag.	○	○	○	○
■M9384	Y2 axis's zero home positioning has been completed.	○	○	○	○
M9385	Y2 axis's stop flag (with gradually slow down).	○	○	○	○
M9386	Y2 axis's immediately stop flag.	○	○	○	○
M9387	Y2 axis's table positioning start signal.	○	○	○	○
■M9388	Y2 axis's M-code active flag.	○	○	○	○
M9389	Y2 axis's M-code clear command.	○	○	○	○
M9390	Y2 axis's external interrupt trigger type. When M9390=“OFF”, the interrupt is triggered by a rising edge; when M9390=“ON”, that is triggered by a falling edge.	○	○	○	○
Y3 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9400	Y3 axis's status. “OFF” means the Y3 is in the READY status, it is available for a positioning instruction; while “ON” = BUSY, the Y3 has been occupying.	○	○	○	○
■M9401	Y3 axis's pulse output monitor. “ON” means pulse is generating.	○	○	○	○
■M9402	Y3 axis's positioning completed flag.	○	○	○	○
■M9403	Y3 axis's positioning abnormal stop flag.	○	○	○	○
■M9404	Y3 axis's zero home positioning has been completed.	○	○	○	○
M9405	Y3 axis's stop flag (with gradually slow down).	○	○	○	○
M9406	Y3 axis's immediately stop flag.	○	○	○	○
M9407	Y3 axis's table positioning start signal.	○	○	○	○

Relay ID No.	Description	Series			
Y3 Axis's Positioning Control Flag		VS1	VS2	VSM	VS3
■M9408	Y3 axis's M-code active flag.	○	○	○	○
M9409	Y3 axis's M-code clear command.	○	○	○	○
M9410	Y3 axis's external interrupt trigger type. When M9410="OFF", the interrupt is triggered by a rising edge; when M9410="ON", that is triggered by a falling edge.	○	○	○	○

Register ID No.		Description	Series			
Y0 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9340	Lower 16 bits	The Y0's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz.	○	○	○	○
D9341	Upper 16 bits					
D9342	The Y0's bias speed. Available range: 0~20kHz.		○	○	○	○
D9343	The Y0's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9344	The Y0's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9345	The Y0's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.		○	○	○	○
D9346	Lower 16 bits	Tthe Y0's preset value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9347	Upper 16 bits					
D9348	The Y0's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).		○	○	○	○
■D9349	The Y0's M-code register.		○	○	○	○
■D9350	Lower 16 bits	The Y0's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9351	Upper 16 bits					
D9352	Lower 16 bits	The Y0's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9353	Upper 16 bits					
D9354	Lower 16 bits	The Y0's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9355	Upper 16 bits					
Y1 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9360	Lower 16 bits	The Y1's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz.	○	○	○	○
D9361	Upper 16 bits					
D9362	The Y1's bias speed. Available range: 0~20kHz.		○	○	○	○
D9363	The Y1's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9364	The Y1's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9365	The Y1's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.		○	○	○	○
D9366	Lower 16 bits	The Y1's preset value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9367	Upper 16 bits					
D9368	The Y1's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).		○	○	○	○
■D9369	The Y1's M-code register.		○	○	○	○
■D9370	Lower 16 bits	The Y1's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9371	Upper 16 bits					
D9372	Lower 16 bits	The Y1's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9373	Upper 16 bits					
D9374	Lower 16 bits	The Y1's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9375	Upper 16 bits					

Register ID No.		Description	Series			
Y2 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9380	Lower 16 bits	The Y2's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz.	○	○	○	○
D9381	Upper 16 bits					
D9382	The Y2's bias speed. Available range: 0~20kHz.		○	○	○	○
D9383	The Y2's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9384	The Y2's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9385	The Y2's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.		○	○	○	○
D9386	Lower 16 bits	The Y2's preset value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9387	Upper 16 bits					
D9388	The Y2's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).		○	○	○	○
■D9389	The Y2's M-code register.		○	○	○	○
■D9390	Lower 16 bits	The Y2's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9391	Upper 16 bits					
D9392	Lower 16 bits	The Y2's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9393	Upper 16 bits					
D9394	Lower 16 bits	The Y2's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9395	Upper 16 bits					
Y3 Axis's Positioning Control			VS1	VS2	VSM	VS3
D9400	Lower 16 bits	The Y3's maximum speed. Available range: VS1, VS2 is 1Hz~50kHz; VSM, VS3 is 1Hz~200kHz; VSM-28ML is 1Hz~1MHz.	○	○	○	○
D9401	Upper 16 bits					
D9402	The Y3's bias speed. Available range: 0~20kHz.		○	○	○	○
D9403	The Y3's acceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9404	The Y3's deceleration time, range = 0~32,000ms. If <0, then equal to 0; if >32,000, then equal to 32,000.		○	○	○	○
D9405	The Y3's number of PG0 input when to execute the ZRN instruction, range = 1~32,767. If out of the range, then equal to 1.		○	○	○	○
D9406	Lower 16 bits	The Y3's preset value when the ZRN is finished. Available range: −2,147,483,648~+2,147,483,647.	○	○	○	○
D9407	Upper 16 bits					
D9408	The Y3's pulse output speed multiple ratio, ranging from 1~30,000 = 0.1%~3,000.0%. If over the range, then equal to 1,000 (100.0%).		○	○	○	○
■D9409	The Y3's M-code register.		○	○	○	○
■D9410	Lower 16 bits	The Y3's current speed. For the PLSY or PLSR instruction, that displays the current output frequency (unit: Hz).	○	○	○	○
■D9411	Upper 16 bits					
D9412	Lower 16 bits	The Y3's current location (Present Value, PV). The initial value is 0. For the PLSY or PLSR instruction, that is the amount of output pulses.	○	○	○	○
D9413	Upper 16 bits					
D9414	Lower 16 bits	The Y3's current location (Present Value, PV) at positioning. The initial value is 0.	○	○	○	○
D9415	Upper 16 bits					

MEMO

8-3 Positioning Control Instructions

FNC No.	Mnemonic in Ladder Chart	Function Description	Applicable VS 1	2	M	3
300	D ZRN S1 S2 D1 D2	Zero (Home Position) Return	○	○	○	○
301	D JOGF S1 S2 D1 D2	Jog Forward	○	○	○	○
302	D JOGR S1 S2 D1 D2	Jog Reverse	○	○	○	○
303	D DRVR S1 S2 D1 D2	Drive to Relative Position	○	○	○	○
304	D DRVA S1 S2 D1 D2	Drive to Absolute Position	○	○	○	○
305	D DV2R S1 S2 D1 D2	Drive to Relative Position by 2 Stages	○	○	○	○
306	D DV2A S1 S2 D1 D2	Drive to Absolute Position by 2 Stages	○	○	○	○
307	D DVIT S1 S2 D1 D2	Interrupt Constant Quantity Positioning	○	○	○	○
308	D DV2I S1 S2 D1 D2	2 Stages Interrupt Constant Quantity Positioning	○	○	○	○
309	D DVSR S1 S2 D1 D2	Interrupt to Stop or Drive to Relative Position	○	○	○	○
310	D DVSA S1 S2 D1 D2	Interrupt to Stop or Drive to Absolute Position	○	○	○	○
311	D PLSV S D1 D2	Variable Speed Pulse Output	○	○	○	○
312	DTBL S1 S2 D1 D2	Data Table Positioning	○	○	○	○
313	D ABS S D1 D2	Absolute Current Value Read	○	○	○	○
314	MPG S1 S2 D1 D2	Handwheel Positioning	○	○	○	○
315	D LIR S D	Relatively Linear Interpolation	○	○	○	○
316	D LIA S D	Absolutely Linear Interpolation	○	○	○	○

FNC 300	DZRN (S1) (S2) (D1) (D2)	Zero Return (Home positioning)	1	2	M	3
			○	○	○	○

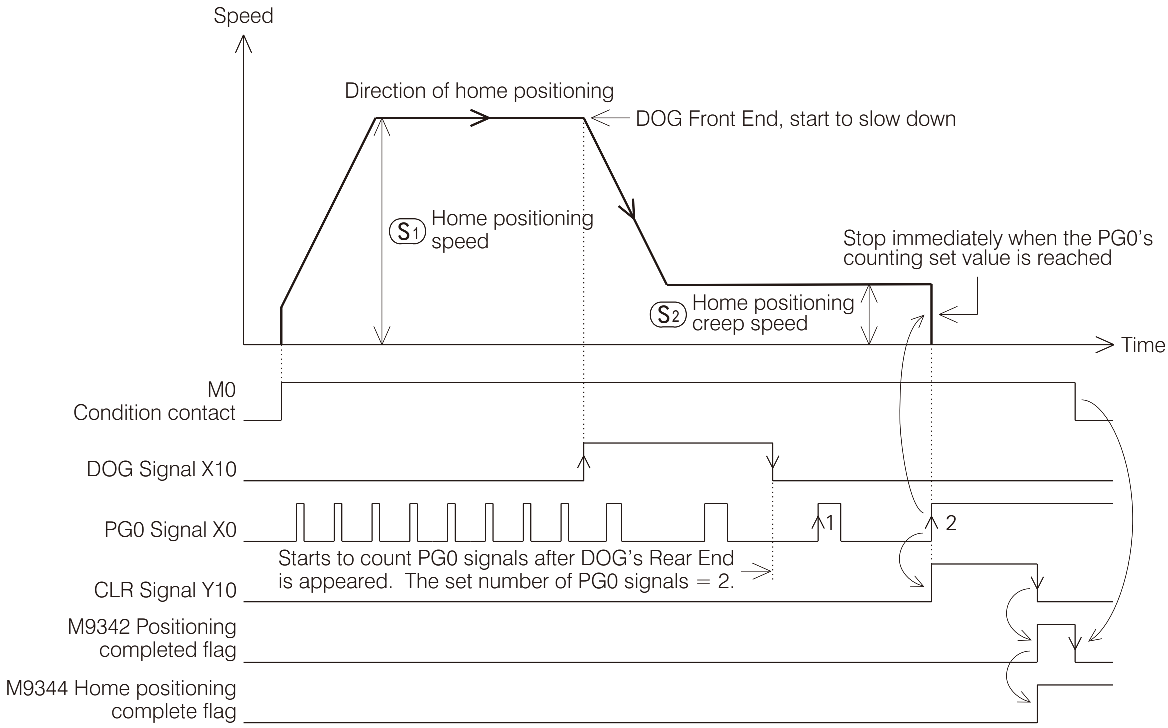
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												

• D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7

```
M0              (S1)     (S2)  (D1) (D2)
─┤├───── DZRN  K50000   K1000  Y0   Y4
```

S1 : the home positioning speed
S2 : the home positioning creep speed
D1 : the output point of generated pulse string
D2 : the output point of direction control signal

- The executive mode of this zero return example is using the “DOG Rear End with PG0 count home positioning” which is appointed by the “Positioning Parameter Setup” under the Ladder Master S. Also, The DOG signal is assigned to X10, PG0 signal to X0 and CLR outputs by Y10.



The DOG signal may use a X or M. Usually assigned to the X0~X7, because the interrupt input point can have the instant response.
Should avoid the DOG's Rear End signal to be too close to the PG0's Front End signal, that may cause the PG0 counting error.
For the PG0 signal input, must use X0~X7. The PLC processes the counting by the way of interrupt.
The CLR signal output can assign to a Y or M, the width of that “ON” signal is ≥ 20ms.

- When M0 = “ON”, the ZRN instruction starts to execute. With the assigned speeds (S1), (S2) and following the selected home return mode, to complete the home positioning action. Then the preset value of home position (D9347, D9346) will be duplicated to the Present Value of Y0 axis (D9355, D9354).
The VS series PLC provides with 5 home return modes (DOG Rear End home positioning / DOG Front End home positioning / DOG Rear End with PG0 count home positioning / DOG Front End with PG0 count home positioning and Data-set type home return). For the details of their actions, please refer to the section 8-1-3 “Positioning Operation Setup”.

(S1) is for to assign the operating speed of home positioning
The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
When (S1) < the bias speed, it will be regarded as the bias speed. Conversely, when (S1) > the maximum speed, it will be regarded as the maximum speed.

S2 is for to assign the creep speed when the DOG near point is triggered.
The available range is from the bias speed (D9342) to 30k Hz.
When S2 < the bias speed, it will be regarded as the bias speed; when S2 > 30k Hz, it will be regarded as 30k Hz.

D1 is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

D2 is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse.
Besides, the "ON"/"OFF" status of the direction control signal is related to both the parameters of the rotational direction (increase Present Value when forward / backward) and the return direction (by the direction of present value increasing / decreasing).

- When the M0 turns from "OFF" to "ON", this ZRN instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), S1 the home positioning speed, S2 the home positioning creep speed, the speed multiple ratio (D9348) and the setup about home positioning to decide the home positioning procedures. During this instruction is in operation, to change any parameter above will be regarded as invalid. Which is different from other positioning instructions and should pay attention to this.
- When the return action is completed, that will let the zero home positioning has been completed flag M9344 = "ON".
- If the condition contact M0 turns "OFF" before it is finished, the movement will be gradually slow down then stop.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag during the unfinished positioning or the ZRN instruction could not find the DOG signal, that will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9344 is the home positioning completed flag of the Y0. (M9364, M9384, M9404)
When the Y0's ZRN instruction has completed, M9344 = "ON".
When the ZRN instruction is operated again, the M9344 will be reset to "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
Before the ZRN instruction operates, the actual operating speed at Y0 positioning can be modified by changing the ratio. But, to change the ratio could not modify the actual operating speed during the ZRN instruction is operating.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.
When this ZRN instruction is operated and completed, the preset value of the home point will be filled in.

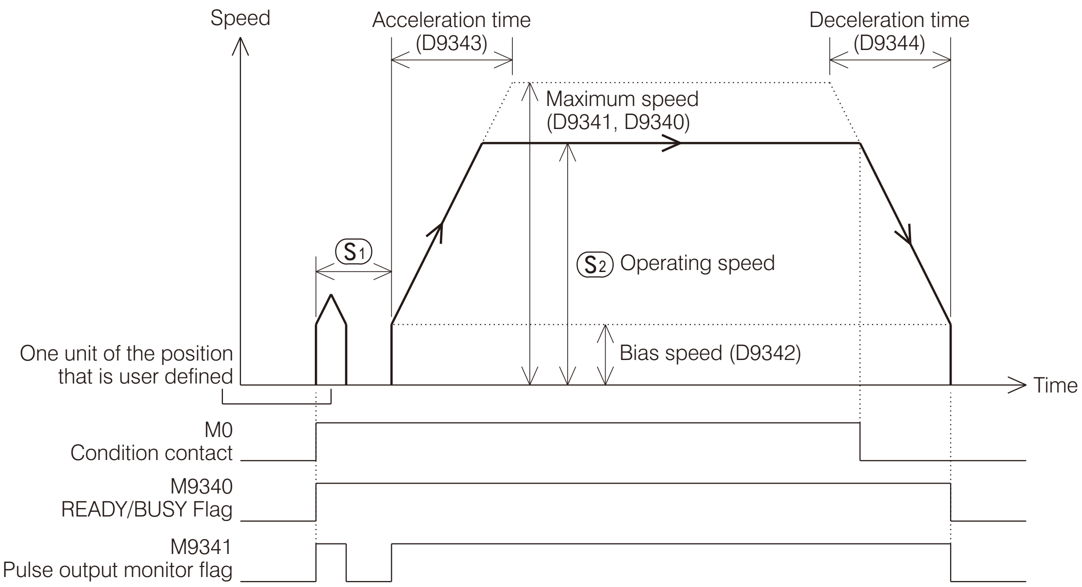
FNC	Instruction	Function	1	2	M	3
FNC 301	D JOGF S1 S2 D1 D2	Jog Forward	○	○	○	○
FNC 302	D JOGR S1 S2 D1 D2	Jog Reverse	○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												

• S1 = 0~5,000 • D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7

```
M0          S1   S2  D1 D2
─┤ ├──── JOGF K300 D20 Y0 Y4
```

S1 : the start delay time (Unit: ms.)
S2 : the operating speed
D1 : the output point of generated pulse string
D2 : the output point of direction control signal



• When M0 = "ON", the Y0 outputs pulse string as the diagram above. During the pulse string generates, the content value of the S2 can be modified in order to change the operating speed. If the condition contact M0 turns "OFF" during the pulse string is output, the movement will be gradually slow down then stop.
If the period of the M0 = "ON" is less than the start delay time S1 or the time needed for one unit of the position, the Y0 will generate the particular number of pulses that is equal to one unit of the position.
If the period of the M0 = "ON" is longer than the start delay time S1 , as the diagram above shows, first the Y0 will generate the particular number of pulses that is equal to one unit of the position. After the start delay time S1 is reached, it begins to generate pulses continually.

S1 is for to assign the start delay time, unit: ms.
The available range is 0~5,000.
When S1 < 0, it will be regarded as 0; when S1 >5,000, it will be regarded as 5,000.

S2 is for to assign the operating speed of JOG.
The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
When S2 < the bias speed, it will be regarded as the bias speed; when S2 > the maximum speed, it will be regarded as the maximum speed.

D1 is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

D2 is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by the JOGF or JOGR instruction is executed.

- When the M0 turns from “OFF” to “ON”, the JOGF or JOGR instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), S1 the start delay time, S2 the operating speed and the speed multiple ratio (D9348) to decide the jog positioning procedures. During the instruction is in operation, could change the value of the S2 operating speed or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.
- These instructions can be used as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction “OFF” could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = “ON”, that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is “ON”, the instruction will not execute.
If the M9340 = “OFF”, that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = “ON”.

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned “ON”, the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned “ON”, the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the “Positioning Parameter Setup” of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from “OFF” to “ON”, the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 303	D DRVR (S1) (S2) (D1) (D2)	Drive to Relative Position	1	2	M	3
			○	○	○	○

Operand	Devices																
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												

• D_1 = Y0~Y3 • If the D_2 is assigned to a Y, it must use Y0~Y7

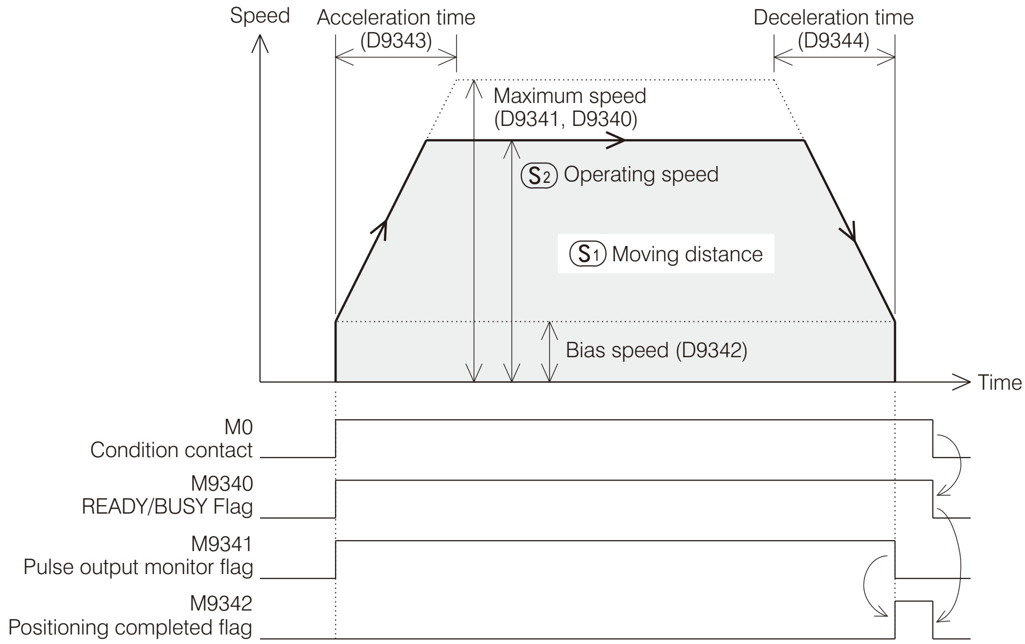
```
        M0                  (S1)  (S2)  (D1)  (D2)
  ──┤ ├─────────────[ DRVR  D10   D20   Y0    Y4 ]
```

S_1 : the moving distance

S_2 : the operating speed

D_1 : the output point of generated pulse string

D_2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, the Y0 by the set operating speed (S2) to generate the corresponding number of pulses. That pulse number comes from the set value of moving distance (S1). When the number of the moving distance is finished, it stops to generate pulses, the positioning action is completed.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the set value of the moving distance (S1) is a positive number, that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the set value of the moving distance is a negative number, that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.

(S1) is for to assign the moving distance.
If the (S1) > 0, it will move in the direction of increasing the Present Value. Conversely, if the (S1) < 0, it will move in the direction of decreasing the Present Value.

(S2) is for to assign the operating speed.
The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
When (S2) < the bias speed, it will be regarded as the bias speed; when (S2) > the maximum speed, it will be regarded as the maximum speed.

(D1) is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

(D2) is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse.
Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/– numerical value of the distance (S1) and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DRVR instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), S1 the moving distance, S2 the operating speed and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the S2 operating speed or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.
- When this instruction completes the S1 specific distance move, that will let the positioning completed flag M9342 = "ON".
- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".
- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

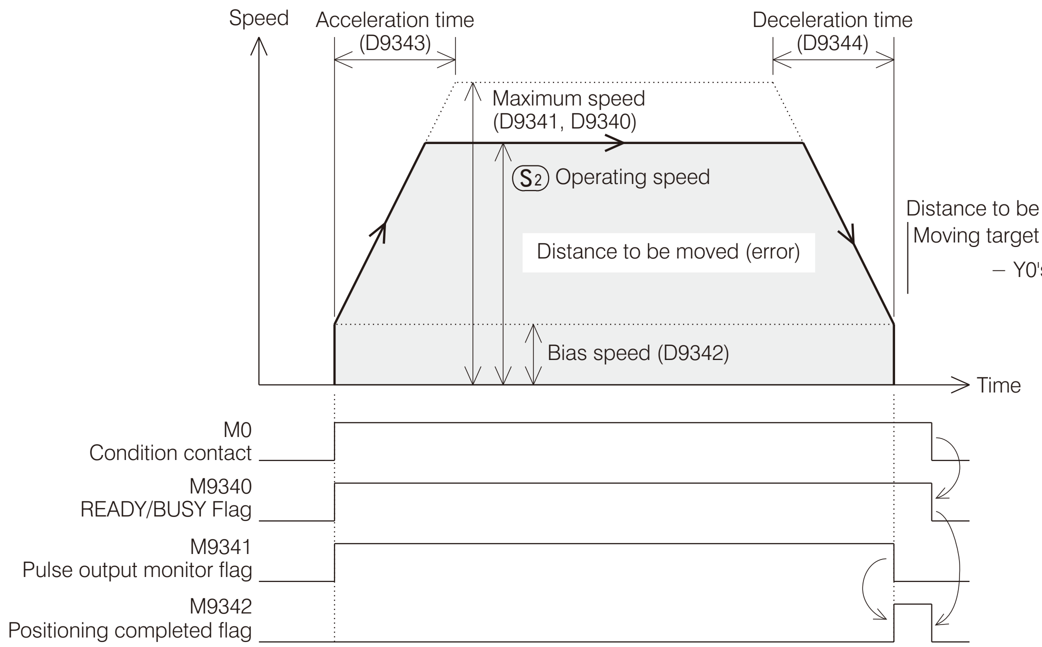
FNC 304	⊣ ⊢ DDRVA (S1) (S2) (D1) (D2)	Drive to Absolute Position	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												

• D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7

```
M0          (S1) (S2) (D1) (D2)
─┤ ├───── DDRVA  D10  D20  Y0  Y4
```

S1 : the moving target
S2 : the operating speed
D1 : the output point of generated pulse string
D2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, the Y0 by the set operating speed (S2) to generate the corresponding number of pulses. That number of pulses is converted from the error (distance to be moved) between the set value of (S1) (D11, D10) and the Present Value of Y0's position (D9355, D9354). When the positioning action is completed (to make the PV at D9355, D9354 = D11, D10), it stops to generate pulses.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the calculated error (distance to be moved) is a positive number (the moving target (S1) > the axis's Present Value), that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the calculated error (distance to be moved) is a negative number (the moving target (S1) < the axis's Present Value), that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.

  (S1) is for to assign the moving target.
  This moving target is an absolute point relative to the zero point. If the calculated error is a positive ((S1) − PV > 0) , it will move in the direction of increasing the Present Value. Conversely, if the calculated error is a negative ((S1) − PV < 0), it will moving in the direction of decreasing the Present Value.

  (S2) is for to assign the operating speed.
  The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
  When (S2) < the bias speed, it will be regarded as the bias speed; when (S2) > the maximum speed, it will be regarded as the maximum speed.

  (D1) is for to assign the output point of generated pulse string.
  It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

  (D2) is for to assign the output point of direction control signal.
  When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/− numerical value of the error ((S1) − PV) and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DRVA instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the moving target S1 , the Present Value (PV, D9355, D9354), the operating speed S2 and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the operating speed S2 or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.

- When this instruction completes the movement (generates pulses to make its Present Value PV equal to the S1), that will let the positioning completed flag M9342 = "ON".

- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".

- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.

- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 305	D DV2R (S1) (S2) (D1) (D2)	Drive to Relative Position by 2 Stages	1	2	M	3
			○	○	○	○

Operand	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●	○				
S2											●	●	●	○				
D1		●																
D2		●	●	●	○	○												

(Devices)

- S1 and S2 occupy 2 components respectively • D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7

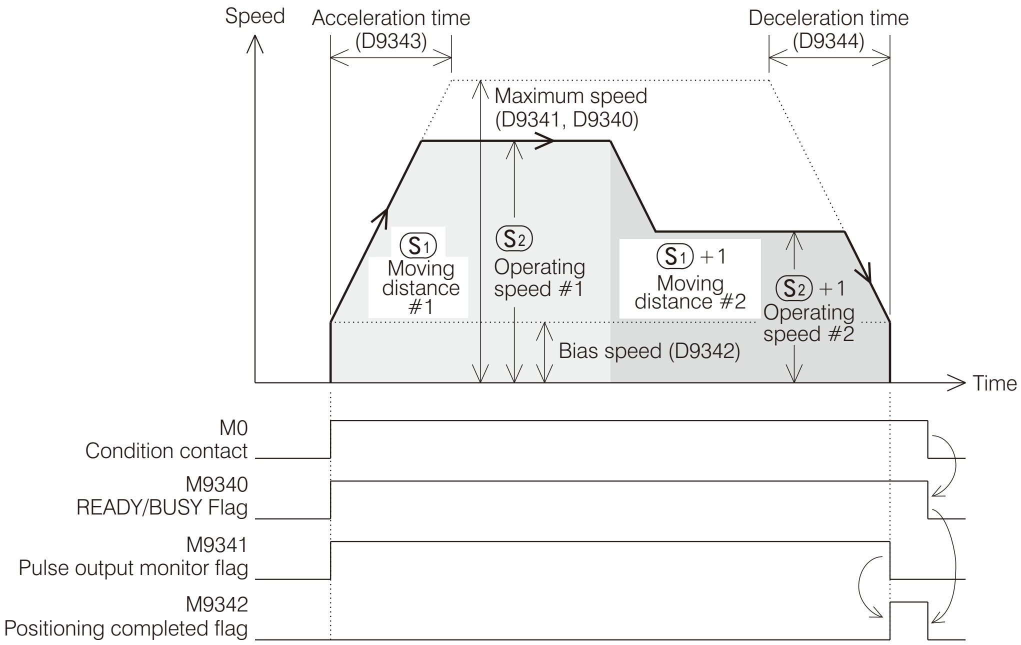
```
        M0                (S1) (S2) (D1) (D2)
  ─┤ ├───────  DV2R  D10  D20  Y0   Y4
```

S1 : the head register of moving distances

S2 : the head register of operating speeds

D1 : the output point of generated pulse string

D2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, at the first stage, the Y0 uses the operating speed #1 (D20) and distance #1 (D10) to generate the corresponding number of pulses. Subsequently at the second stage, it uses the operating speed #2 (D21) and distance #2 (D11) to generate the corresponding number of pulses. When the distance #2 is finished, it stops to generate pulses, the two stages positioning action is completed.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the set value of the moving distance is a positive number, that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the set value of the moving distance is a negative number, that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.

  (S1) is for to assign the moving distance #1 and the following (S1)+1 is for to assign the moving distance #2.
  If the content value of the moving distance > 0, it will move in the direction of increasing the Present Value. Conversely, if the content value of the moving distance < 0, it will move in the direction of decreasing the Present Value.

  (S2) is for to assign the operating speed #1 and the following (S2)+1 is for to assign the operating speed #2.
  The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
  When the content value of the operating speed < the bias speed, it will be regarded as the bias speed; when the content value of the operating speed > the maximum speed, it will be regarded as the maximum speed.

  (D1) is for to assign the output point of generated pulse string.
  It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

  (D2) is for to assign the output point of direction control signal.
  When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/– numerical value of the distance to be moved at the stage and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DV2R instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the moving distances the operating speeds and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the operating speeds or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.

- When this instruction completes the movement of the amount of two specific moving distances, that will let the positioning completed flag M9342 = "ON".

- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".

- If the values of the moving distances #1 and #2 are +/− opposite, the positioning of the stage #1 will be executed first. After that, it will make a brief pause, then carry on the operation of the opposite direction immediately.

- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.

- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

  M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
  Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
  If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
  If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

  M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
  When the Y0 is generating pulses, M9341 = "ON".

  M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
  When the Y0's positioning instruction is completed, M9342 = "ON".
  Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

  M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
  Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
  Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

  M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
  When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
  After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

  M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
  When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
  After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

  Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
  When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
  At this moment, only the moving request of opposite direction can be accepted.

  D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
  The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
  During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
  Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

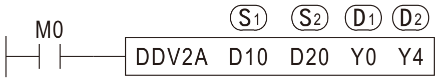
  D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
  When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

  D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
  When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

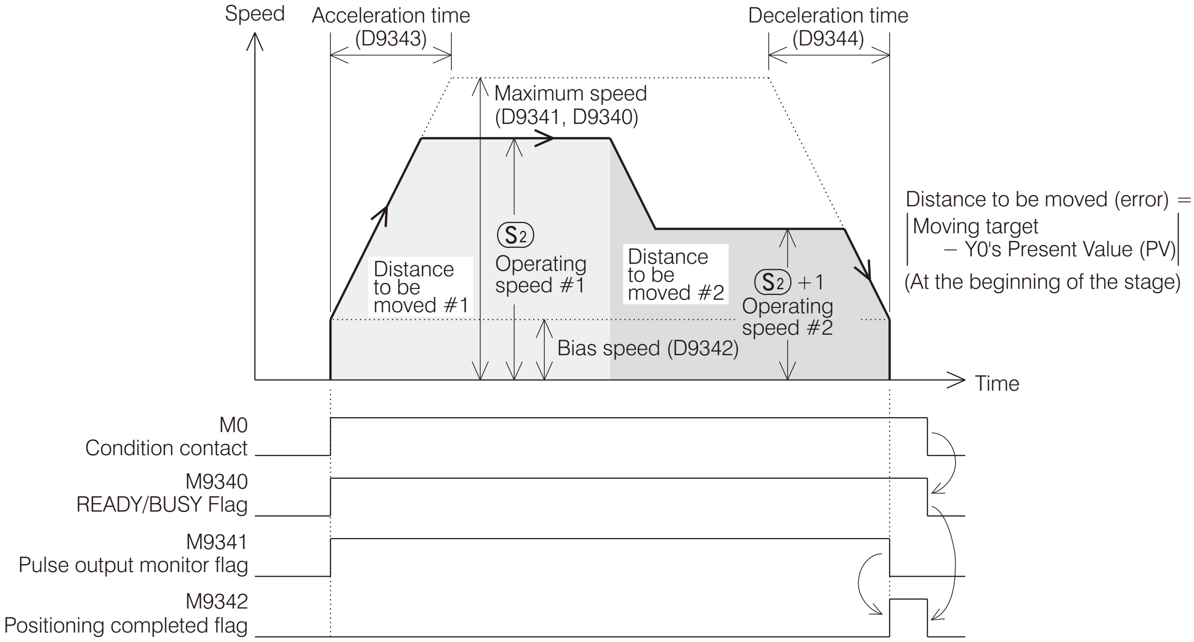
FNC 306	DDV2A S1 S2 D1 D2	Drive to Absolute Position by 2 Stages	1 ○	2 ○	M ○	3 ○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●	○				
S2											●	●	●	○				
D1		●																
D2		●	●	●	○	○												

• S1 and S2 occupy 2 components respectively • D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7



S1 : the head register of moving targets
S2 : the head register of operating speeds
D1 : the output point of generated pulse string
D2 : the output point of direction control signal



- At the first stage, the Y0 uses the operating speed #1 (D21, D20) to generate pulses that will let its Present Value (PV at D9355, D9354) equal to moving target #1 (D11, D10). Subsequently at the second stage, it uses the operating speed #2 (D23, D22) to generate pulses that will let its Present Value (PV at D9355, D9354) equal to moving target #2 (D13, D12). When the target #2 is arrived, it stops to generate pulses, the two stages positioning is completed.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the error (distance to be moved) is a positive number (the moving target > the axis's Present Value), that needs to increase the present value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the error (distance to be moved) is a negative number (the moving target < the axis's Present Value), that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.

  S1 is for to assign the moving target #1 and the following S1+1 is for to assign the moving target #2.
  At the beginning of each stage, if the calculated of the distance to be moved (error) > 0 (moving target > PV), this stage will move in the direction of increasing the Present Value. Conversely, if the calculated distance to be moved (error) < 0 (moving target < PV), this stage will move in the direction of decreasing the Present Value.

  S2 is for to assign the operating speed #1 and the following S2+1 is for to assign the operating speed #2.
  The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
  When the content value of the operating speed < the bias speed, it will be regarded as the bias speed; when the content value of the operating speed > the maximum speed, it will be regarded as the maximum speed.

  D1 is for to assign the output point of generated pulse string.
  It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

  D2 is for to assign the output point of direction control signal.
  When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/− numerical value of the error at the stage and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DV2A instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the moving targets, the Present Value (PV, D9355, D9354), the operating speeds and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the operating speeds or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.
- When this instruction completes the movement (that generates pulses to make its Present Value PV equal to the moving target #1 and #2 sequentially), that will let the positioning completed flag M9342 = "ON".
- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".
- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

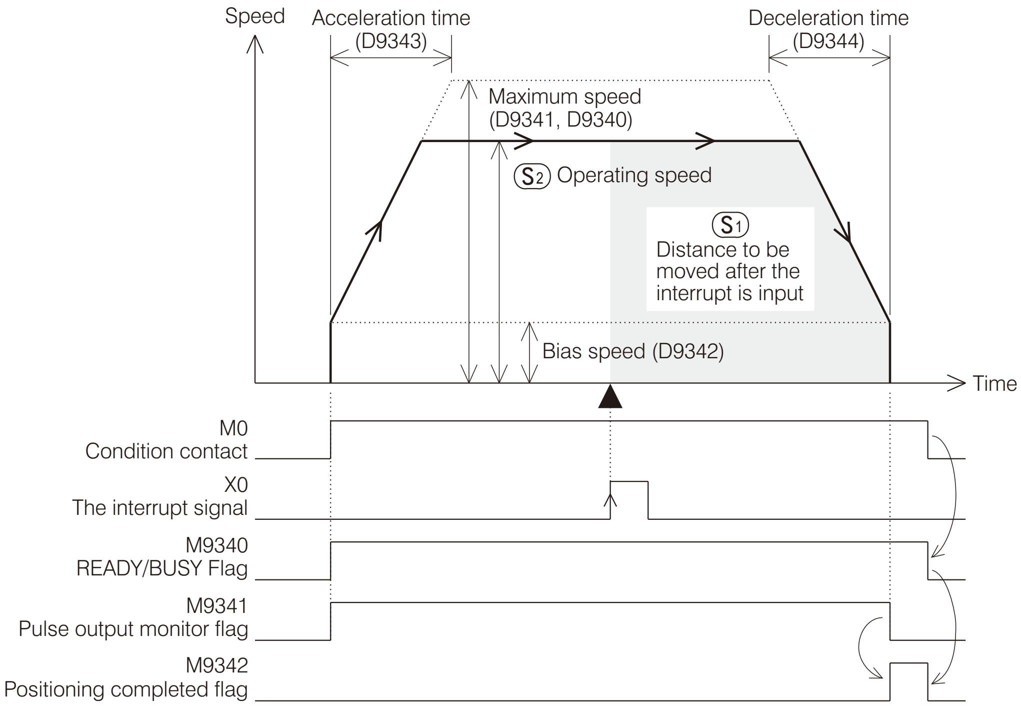
D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 307	DDVIT S1 S2 D1 D2	Interrupt Constant Quantity Positioning	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												
• D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7																		

```
        S1  S2  D1 D2
 M0
─┤├─── DVIT D10 D20 Y0 Y4
```

S1 : the distance to be moved after the interrupt is input
S2 : the operating speed
D1 : the output point of generated pulse string
D2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, the Y0 by the set operating speed S2 to generate pulses. Until the designated interrupt signal X0 is input ("OFF" → "ON"), it will continuously generate the corresponding number of pulses then stop. That pulse number comes from the set value of distance to be moved after the interrupt is input S1 . When the number of distance is finished, the positioning action is completed.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the set value of the distance to be moved after the interrupt S1 is a positive number, that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the set value of the distance to be moved after the interrupt S1 is a negative number, that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.
  The interrupt signal in this instruction is appointed by the "Positioning Parameter Setup" under the Ladder Master S. Also, the signal must input from X0~X7, so the PLC can have an instant interrupt reaction.
  Besides, the M9350 can be used to assign the interrupt trigger mode ("OFF" = rising edge; "ON" = falling edge).

S1 is for to assign the distance to be moved after the interrupt is input.
If the S1 > 0, it will move in the direction of increasing the Present Value. Conversely, if the S1 < 0, it will move in the direction of decreasing the Present Value.

S2 is for to assign the operating speed.
The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
When S2 < the bias speed, it will be regarded as the bias speed; when S2 > the maximum speed, it will be regarded as the maximum speed.

D1 is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

D2 is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/– numerical value of the S1 and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DVIT instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the distance to be moved after the interrupt S1 , the operating speedt S2 and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the S2 operating speed or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.
- When this instruction has an interrupt input after it is active and it completes the S1 specific distance move after the interrupt, that will let the positioning completed flag M9342 = "ON".
- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".
- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9350 is the flag to assign the Y0 interrupt input trigger mode. (M9370, M9390, M9410)
When the M9350 is "OFF", the interrupt will be triggered by the rising edge of the designated input point.
When the M9350 is "ON", the interrupt will be triggered by the falling edge of the designated input point.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

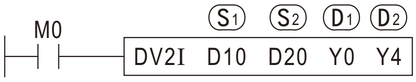
D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

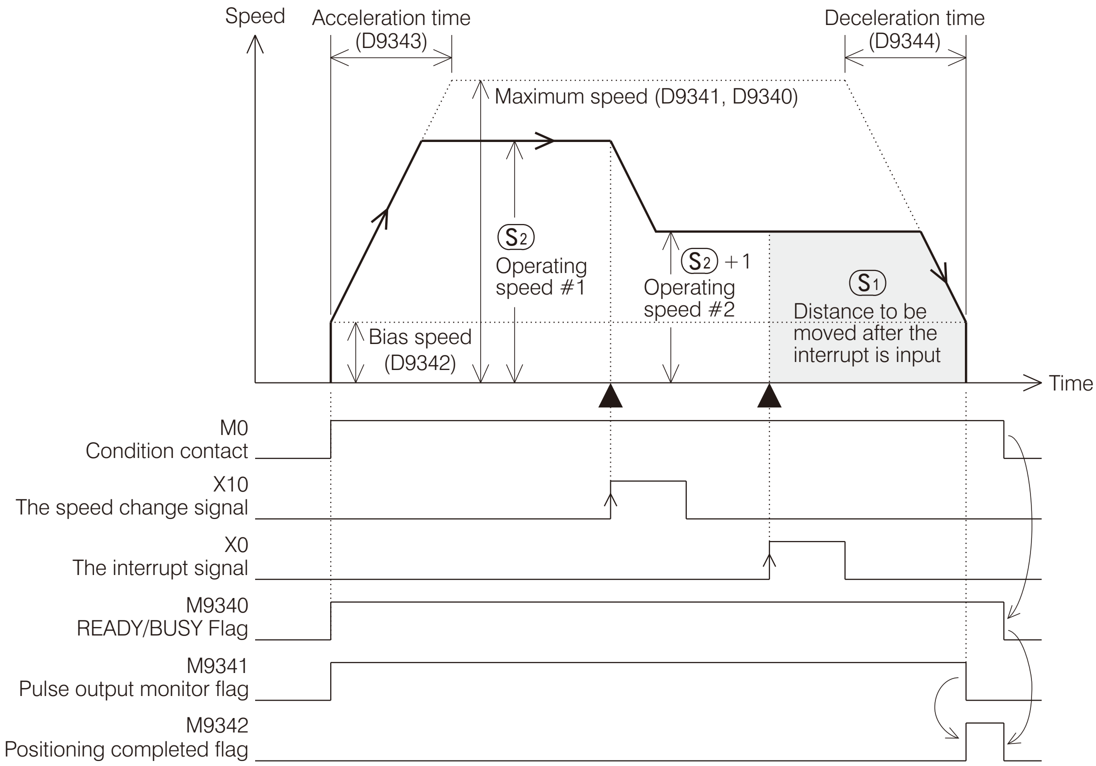
FNC 308	DDV2I (S1) (S2) (D1) (D2)	2 Stages Interrupt Constant Quantity Positioning	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1											●	●	●	○	●	●		
S2											●	●	●	○				
D1		●																
D2		●	●	●	○	○												

• S2 occupies 2 components • D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7



S1 : the distance to be moved after the interrupt is input
S2 : the head register of operating speeds
D1 : the output point of generated pulse string
D2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, at the first stage, the Y0 uses the operating speed #1 (D20) to generate pulses. Since the designate speed change signal X10 is input ("OFF" → "ON"), it will uses the operating speed #2 (D21) to generate pulses. Until the designated interrupt signal X0 is input ("OFF" → "ON"), it will continuously generate the particular number of pulses then stop. That pulse number comes from the set value of distance to be moved after the interrupt is input (S1). When the number of distance is finished, the positioning action is completed. Assume that the rotational mode "Increase Present Value when forward" is selected. If the set value of the distance to be moved after the interrupt (S1) is a positive number, that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the set value of the distance to be moved after the interrupt (S1) is a negative number, that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.
  The interrupt signal and the speed change signal in this instruction are appointed by the "Positioning Parameter Setup" under the Ladder Master S. However, the interrupt signal must use X0~X7 and the M9350 can be used to assign the interrupt trigger mode ("OFF" = rising edge; "ON" = falling edge).

  (S1) is for to assign the distance to be moved after the interrupt is input.
  If the (S1) > 0, it will move in the direction of increasing the Present Value. Conversely, if the (S1) < 0, it will move in the direction of decreasing the Present Value.

  (S2) is for to assign the operating speed #1 and the following (S2)+1 is for to assign the operating speed #2.
  The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
  When the content value of the operating speed < the bias speed, it will be regarded as the bias speed; when the content value of the operating speed > the maximum speed, it will be regarded as the maximum speed.

D1 is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

D2 is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/– numerical value of the distance S1 and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DV2I instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the distance to be moved after the interrupt S1, the operating speeds and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the operating speeds or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.

- When this instruction has an interrupt input after it changes to the operating speed #2 and it completes the S1 specific distance move after the interrupt, that will let the positioning completed flag M9342 = "ON".

- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".

- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.

- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9350 is the flag to assign the Y0 interrupt input trigger mode. (M9370, M9390, M9410)
When the M9350 is "OFF", the interrupt will be triggered by the rising edge of the designated input point.
When the M9350 is "ON", the interrupt will be triggered by the falling edge of the designated input point.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

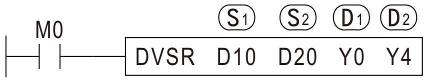
D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

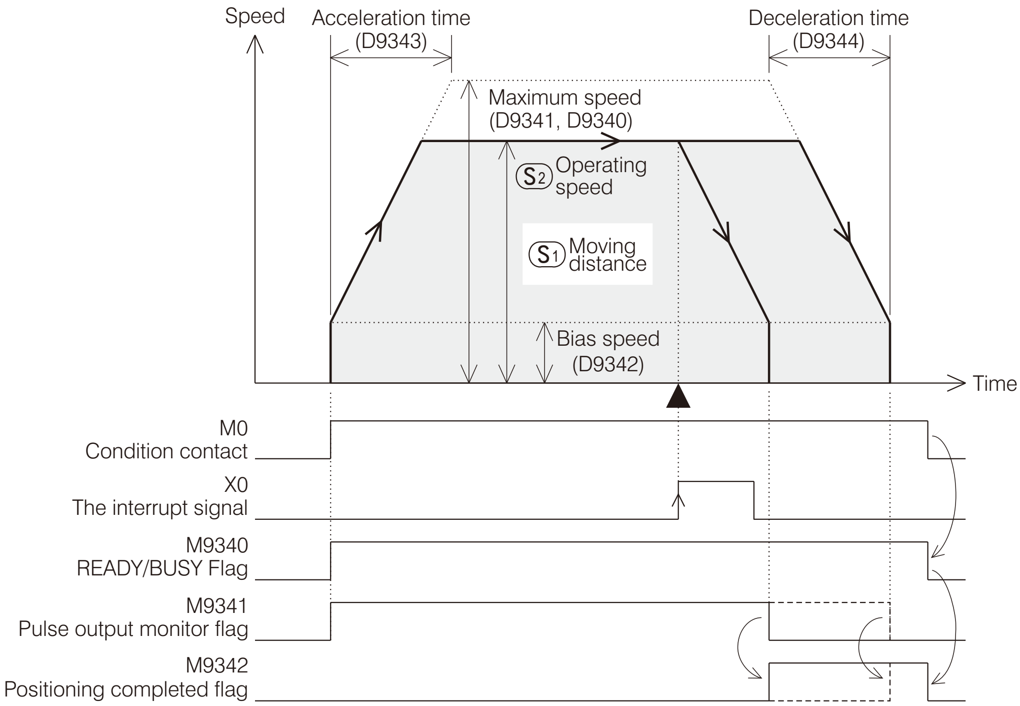
D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is set as 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 309	D DVSR (S1) (S2) (D1) (D2)	Interrupt to Stop or Drive to Relative Position	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S_1							●	●	●	●	●	●	●	○	●	●		
S_2							●	●	●	●	●	●	●	○	●	●		
D_1		●																
D_2		●	●	●	○	○												
• D_1 = Y0~Y3 • If the D_2 is assigned to a Y, it must use Y0~Y7																		



S_1 : the moving distance
S_2 : the operating speed
D_1 : the output point of generated pulse string
D_2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, the Y0 uses operating speed (S2) and the set value of moving distance (S1) to generate the corresponding number of pulses. If the number of distance is finished, it stops to generate pulses, the positioning action is completed. However, the designated interrupt signal X0 is input ("OFF" → "ON") during the instruction generating pulses, it will immediately slow down then stop, also the positioning action is regarded as completed.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the set value of the distance to be moved (S1) is a positive number, that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the set value of the moving distance (S1) is a negative number, that needs to decrease the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.
  The interrupt signal in this instruction is appointed by the "Positioning Parameter Setup" under the Ladder Master S. Also, the interrupt signal must use an input between the X0~X7 so the PLC can have an instant interrupt reaction. Besides, the M9350 can be used to assign the interrupt trigger mode ("OFF" = rising edge; "ON" = falling edge).

  (S1) is for to assign the moving distance.
  If the (S1) > 0, it will move in the direction of increasing the Present Value. Conversely, if the (S1) < 0, it will move in the direction of decreasing the Present Value.

  (S2) is for to assign the operating speed.
  The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
  When (S2) < the bias speed, it will be regarded as the bias speed; when (S2) > the maximum speed, it will be regarded as the maximum speed.

  (D1) is for to assign the output point of generated pulse string.
  It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

D2 is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse.
Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/– numerical value of the distance S1 and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DVSR instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the moving distance S1, the operating speed S2 and the speed multiple ratio (D9348) to decide the positioning procedures.
During the instruction is in operation, could change the value of the operating speed S2 or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.
- When this instruction is completed by either the S1 specific moving distance or the designated interrupt signal is input during the process to cause it immediately into the procedure of gradually slow down then stop, that will let the positioning completed flag M9342 = "ON".
- If the condition contact M0 turns "OFF" during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 "ON".
- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource.
Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed by either the S1 specific distance movement is ready or the designated interrupt signal is input to cause it gradually slow down then stop, M9342 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342 "OFF".

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9350 is the flag to assign the Y0 interrupt input trigger mode. (M9370, M9390, M9410)
When the M9350 is "OFF", the interrupt will be triggered by the rising edge of the designated input point.
When the M9350 is "ON", the interrupt will be triggered by the falling edge of the designated input point.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 310	DDVSA S1 S2 D1 D2	Interrupt to Stop or Drive to Absolute Position	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1							●	●	●	●	●	●	●	○	●	●		
S2							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												

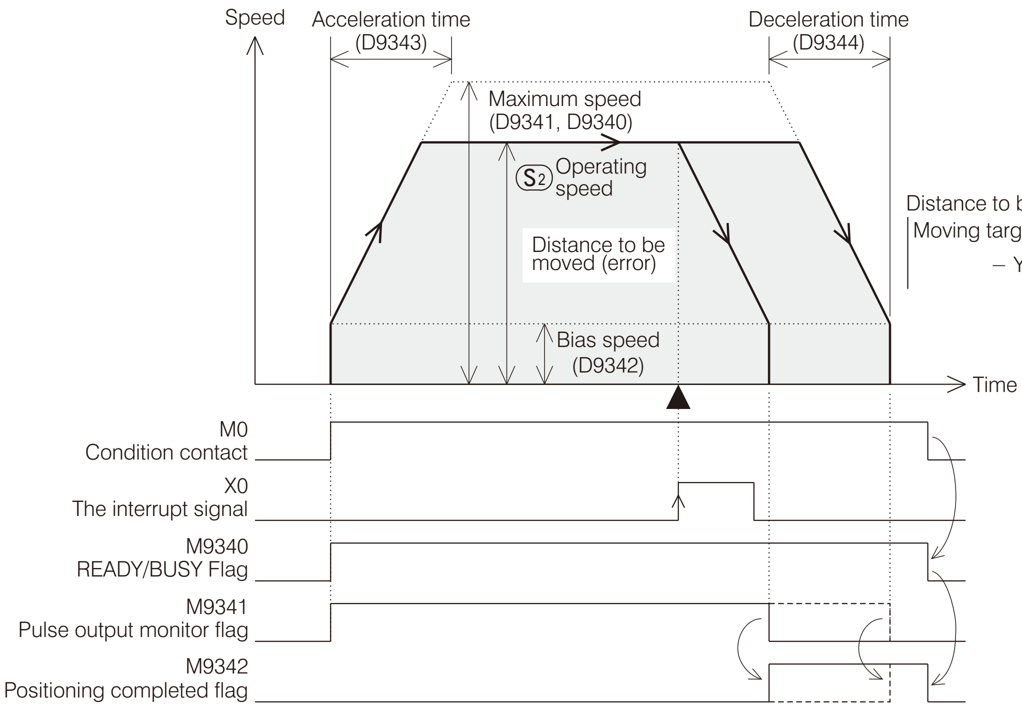
• D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7

S1 : the moving target

S2 : the operating speed

D1 : the output point of generated pulse string

D2 : the output point of direction control signal



• When M0 = "ON", as the diagram above, the Y0 by the set operating speed S2 generates pulses to move the Y0's Present Value (PV) to the set value S1 (D11, D10). When the positioning action is completed (to make the PV at D9355, D9354 = D11, D10), it stops to generate pulses, the positioning action is completed. However, the designated interrupt signal X0 is input ("OFF" → "ON") during the instruction generating pulses, it will immediately slow down then stop, also the positioning action is regarded as completed.
Assume that the rotational mode "Increase Present Value when forward" is selected. If the error (distance to be moved) is a positive number (the moving target > the axis's Present Value), that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. ON the other hand, the error (distance to be moved) is a negative number (the moving target < the axis's Present Value), that needs to decrease the present value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.
The interrupt signal in this instruction is appointed by the "Positioning Parameter Setup" under the Ladder Master S. However, the interrupt signal must use X0~X7 and the M9350 can be used to assign the interrupt trigger mode ("OFF" = rising edge; "ON" = falling edge).

S1 is for to assign the moving target.
This moving target is an absolute point relative to the zero point. If the error is a positive (S1 − PV > 0) , it will move in the direction of increasing the Present Value. Conversely, if the error is a negative (S1 − PV < 0), it will move in the direction of decreasing the Present Value.

S2 is for to assign the operating speed.
The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).
When S2 < the bias speed, it will be regarded as the bias speed; when S2 > the maximum speed, it will be regarded as the maximum speed.

D1 is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

D2 is for to assign the output point of direction control signal.
When the output of direction control signal is “ON”, the motor moves forward; conversely, “OFF” moves reverse.
Besides, the “ON”/“OFF” status of the direction control signal is decided by both the +/− numerical value of the error (S1 − PV) and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from “OFF” to “ON”, this DVSA instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the moving target S1, the Present Value (PV, D9355, D9354), the operating speed S2 and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the operating speed S2 or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.
- When this instruction is completed by either the movement (generates pulses to make its Present Value PV equal to the S1) or the designated interrupt signal is input during the process to cause it immediately into the procedure of gradually slow down then stop, that will let the positioning completed flag M9342 = “ON”.
- If the condition contact M0 turns “OFF” during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 “ON”.
- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction “OFF” could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = “ON”, that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is “ON”, the instruction will not execute.
If the M9340 = “OFF”, that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = “ON”.

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When the Y0's positioning instruction is completed by either the move target is arrived (Present Value PV equal to the S1) or the designated interrupt signal is input to cause it gradually slow down then stop, M9342 = “ON”.
Turn “OFF” the condition contact of the completed instruction will lead the flag M9342 “OFF”.

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = “ON” to indicate this stop is abnormal.
Turn “OFF” the condition contact of the activated instruction will lead the flag M9343 “OFF”.

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned “ON”, the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned “ON”, the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9350 is the flag to assign the Y0 interrupt input trigger mode. (M9370, M9390, M9410)
When the M9350 is “OFF”, the interrupt will be triggered by the rising edge of the designated input point.
When the M9350 is “ON”, the interrupt will be triggered by the falling edge of the designated input point.

Forward / Reverse limit switch (under the “Positioning parameter setting” of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from “OFF” to “ON”, the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

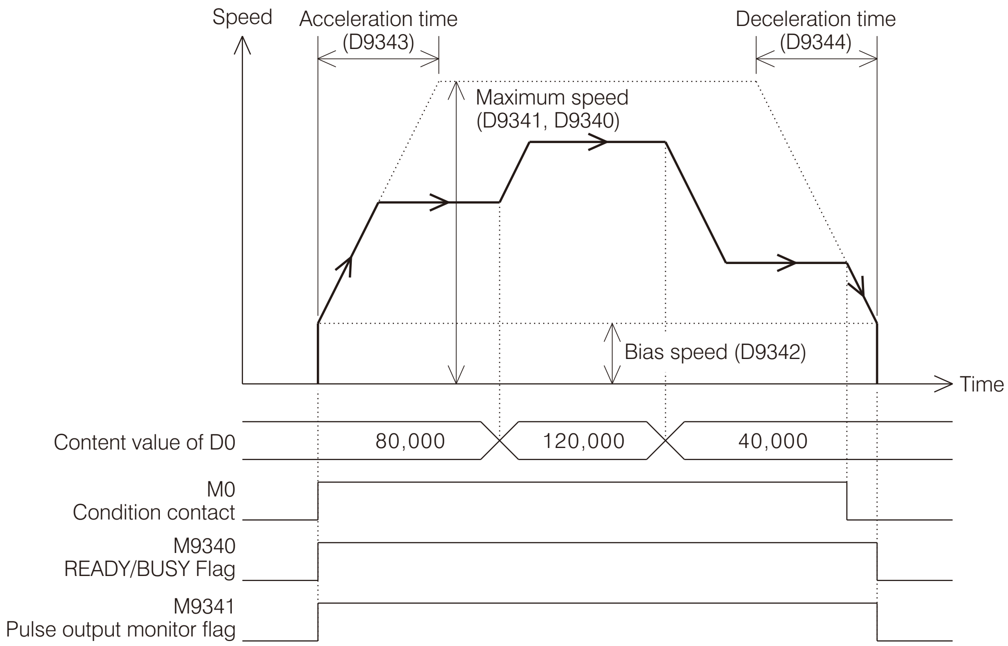
FNC 311	DPLSV (S) (D1) (D2)	Variable Speed Pulse Output	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S							●	●	●	●	●	●	●	○	●	●		
D1		●																
D2		●	●	●	○	○												
• D1 = Y0~Y3	• If the D2 is assigned to a Y, it must use Y0~Y7																	

S : the operating speed

D_1 : the output point of generated pulse string

D_2 : the output point of direction control signal



- When M0 = "ON", as the diagram above, the Y0 uses operating speed (S) to generate pulses. The real output operation speed is available by changing the value of (S) to modify during the positioning. However, the condition contact M0 turns form "ON" to "OFF", it will slow down then stop.
  Assume that the rotational mode "Increase Present Value when forward" is selected. If the operating speed (S) is a positive number, that needs to increase the Present Value, thus the instruction will turn the Y4 "ON" to drive motor moves forward. On the other hand, the operating speed (S) is a negative number, that needs decreasing the Present Value, thus the instruction will turn the Y4 "OFF" to drive motor moves backward.

  (S) is for to assign the operating speed.
  The serviceable range of the absolute value of (S) is from the bias speed (D9342) to the maximum speed (D9341, D9340).
  If the content value of this operating speed (S) > 0, it will move in the direction of increasing the Present Value.
  Conversely, if the content value of this operating speed (S) < 0, it will move in the direction of decreasing the Present Value.
  When the content absolute value of the operating speed (S) < the bias speed, it will be regarded as the bias speed; when the content absolute value of the operating speed (S) > the maximum speed, it will be regarded as the maximum speed.

  (D1) is for to assign the output point of generated pulse string.
  It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

  (D2) is for to assign the output point of direction control signal.
  When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse.
  Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/– numerical value of the operating speed (S) and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- When the M0 turns from "OFF" to "ON", this DLSV instruction uses the parameters of the bias speed (D9342), the maximum speed (D9341, D9340), the acceleration time (D9343), the deceleration time (D9344), the operating speed (S) and the speed multiple ratio (D9348) to decide the positioning procedures. During the instruction is in operation, could change the value of the operating speed (S) or the speed multiple ratio (D9348) to modify the real operating speed, but to change other parameters above will be regarded as invalid.

- If the operating speed (S) changes and that +/– numerical value through this change is opposite, it will gradually slow down then stop. After a brief pause, carry on the operation of the opposite direction immediately.

- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.

- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

  M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
  Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
  If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
  If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

  M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
  When the Y0 is generating pulses, M9341 = "ON".

  M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
  Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0 positioning function, will cause Y0 stops to generate pulses and let the M9343 = "ON" to indicate this stop is abnormal.
  Turn "OFF" the condition contact of the activated instruction will lead the flag M9343 "OFF".

  M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
  When the M9345 is turned "ON", the pulse generating at the Y0 will gradually slow down then stop.
  After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

  M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
  When the M9346 is turned "ON", the pulse generating at the Y0 will stop imminently (without to slow down).
  After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

  Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
  When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
  At this moment, only the moving request of opposite direction can be accepted.

  D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)
  The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
  During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
  Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

  D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
  When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

  D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
  When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 312	DTBL S1 S2 D1 D2	Data Table Positioning	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1	Using a Table Code Q0~Q31 (V, Z index modifiable) or a Table Nickname (by user-defined 16 English characters)																	
S2													○					
D1		●																
D2		●	●	●	○	○												

• D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7 • The S2 occupies 4 components

```
 M0                S1               S2    D1  D2
─┤ ├────[ DTBL  Y0_PROCEDURE_TBL  D100  Y0  Y4 ]
```

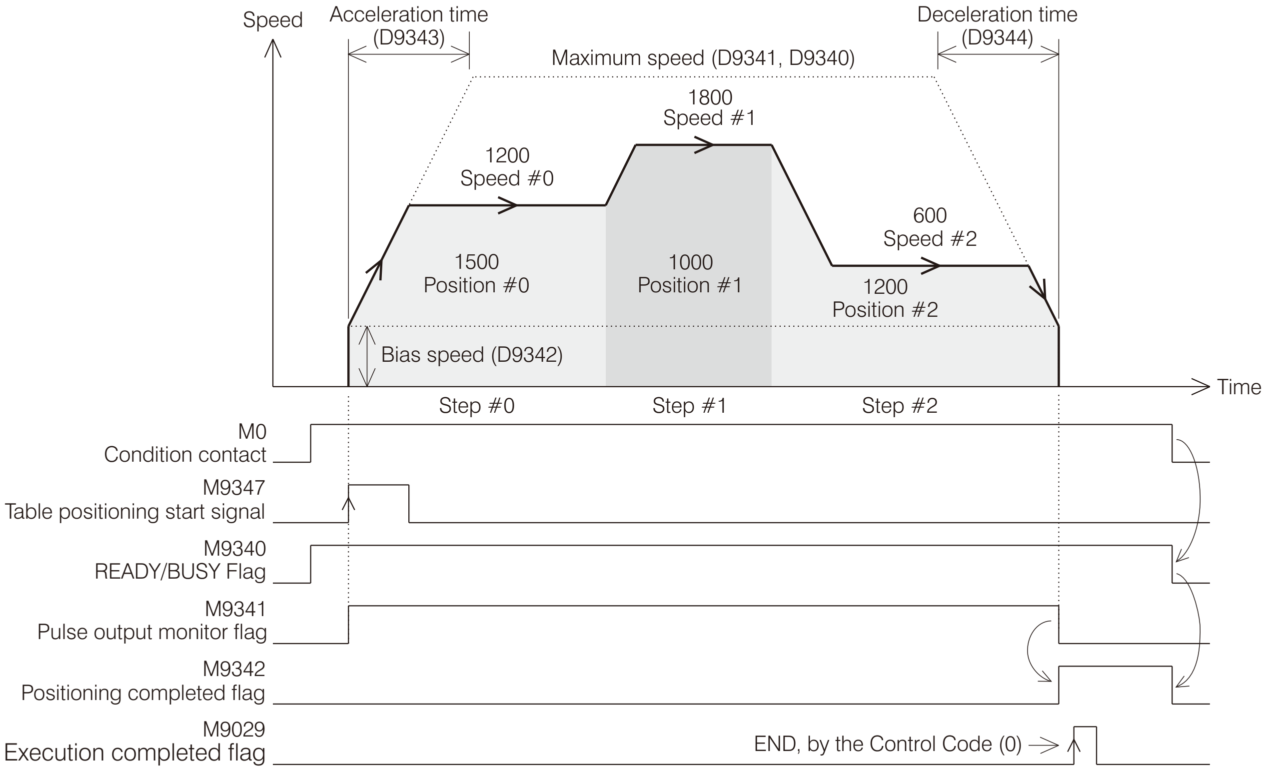
S1 : the procedure table to describe the positioning orders

S2 : the working area for the instruction

D1 : the output point of generated pulse string

D2 : the output point of direction control signal

• This DTBL instruction is to perform a continuous multi-speed positioning control and various multi-step positioning control.



• When the M0 = "ON", the DTBL instruction starts to sequentially control the Y0 by the prepared positioning table "Y0_PROCEDURE_TBL". At the very beginning, that M-code register D9349 will be set to –1. Also the working area D100~D103 will be occupied to store the state of instruction execution.

S1 is the procedure table to describe the positioning orders.

The example table below is a series of procedures for the data table positioning, that is provided by the programming tool Ladder Master S.

Step No.	Control Code	Position Data	Speed Data	M-code Data
	Could use K, D or R (by a 16-bit data)	Could use K, D or R (by a 32-bit data)	Could use K, D or R (by a 32-bit data)	Could use K, D or R (by a 16-bit data)
0	Continuous relative positioning Control Code (3)	Position #0 1,500	Speed #0 1,200	–1
1	Continuous relative positioning Control Code (3)	Position #1 1,000	Speed #1 1,800	–1
2	Continuous relative positioning Control Code (3)	Position #2 1,200	Speed #2 600	–1
3	END, Control Code (0)	–1	–1	–1

Control Code:

This code is to assign a positioning function for each step, which has 6 different functions to choose from.

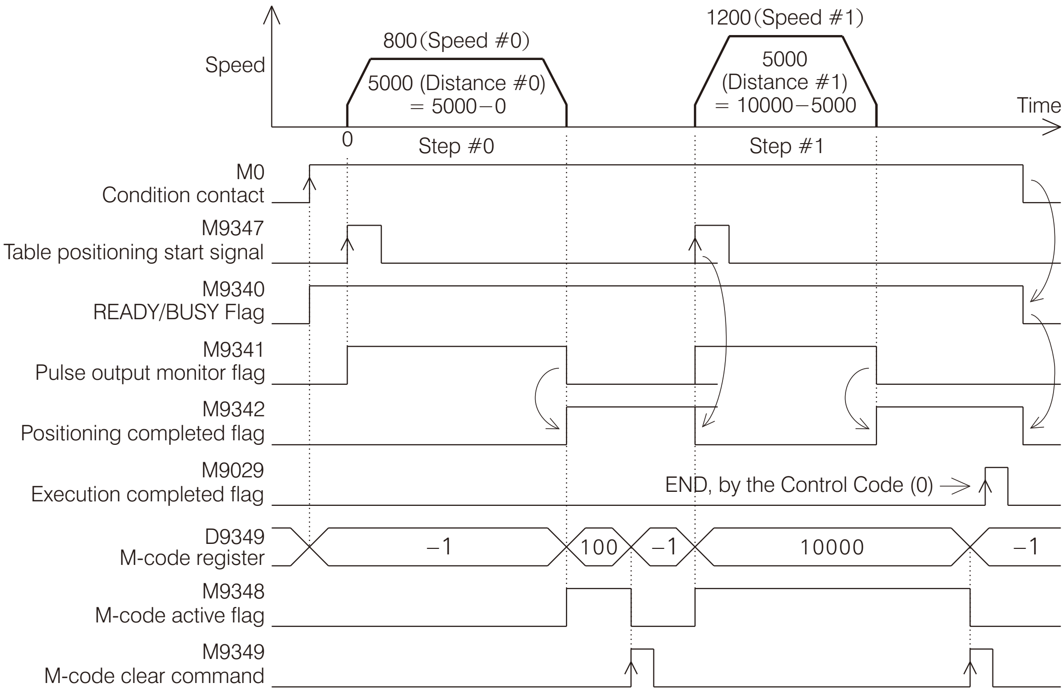
Control Code	Control Function	Description
0	END	One procedure table can store many positioning steps (each step needs 4 components to describe), then the END is the finish code to close this table. When the END code is executed, the Execution Completed Flag M9029 will "ON" for a Scan Time to indicate all the steps in the procedure table are finished.
1	Single relative positioning	This relative positioning will stand-by when the step is completed, that moving distance is defined by the content value at the Position Data and its +/– numerical value decides the moving direction. During the instruction operates, could change its Speed Data to modify the operating speed, but to change Position Data is invalid.
2	Single absolute positioning	This absolute positioning will stand-by when the step's target is arrived, the target is defined by the content value at the Position Data. Since the target is an absolute point relative to the zero point, the moving distance = (Position Data – PV) and that calculated +/– numerical value decides the direction. During the instruction operates, could change its Speed Data to modify the operating speed, but to change Position Data is invalid.
3	Continuous relative positioning	This relative positioning will continue to the next step when the current step is completed. Each step has its own Position Data and Speed Data, to orderly perform a series of steps can become a multiple speed positioning. If a break is required during the continuous relative positioning, it will into stand-by when this step is finished, then waiting for the restart condition to perform the next step.
4	Continuous absolute positioning	This absolute positioning will continue to the next step when the target of current step is arrived. Each step has its own Position Data and Speed Data, to orderly perform a series of steps can become a multiple speed positioning. If a break is required during the continuous absolute positioning, it will into stand-by when this step is finished, then waiting for the restart condition to perform the next step.
5	JMP	Jump to the specific step to perform. At this function, its Position Data is used to appoint the destination step number.

To start a new step (not by continuous), it requires the M-code register D9349 = –1 (OFF) at the time. And at the beginning, should trigger the Table positioning start signal M9347 = "OFF" → "ON".

The Single Relative Positioning and Single Absolute Positioning Functions

Here is using two single relative positioning and suppose its Present Value (PV) is 0 at the beginning.

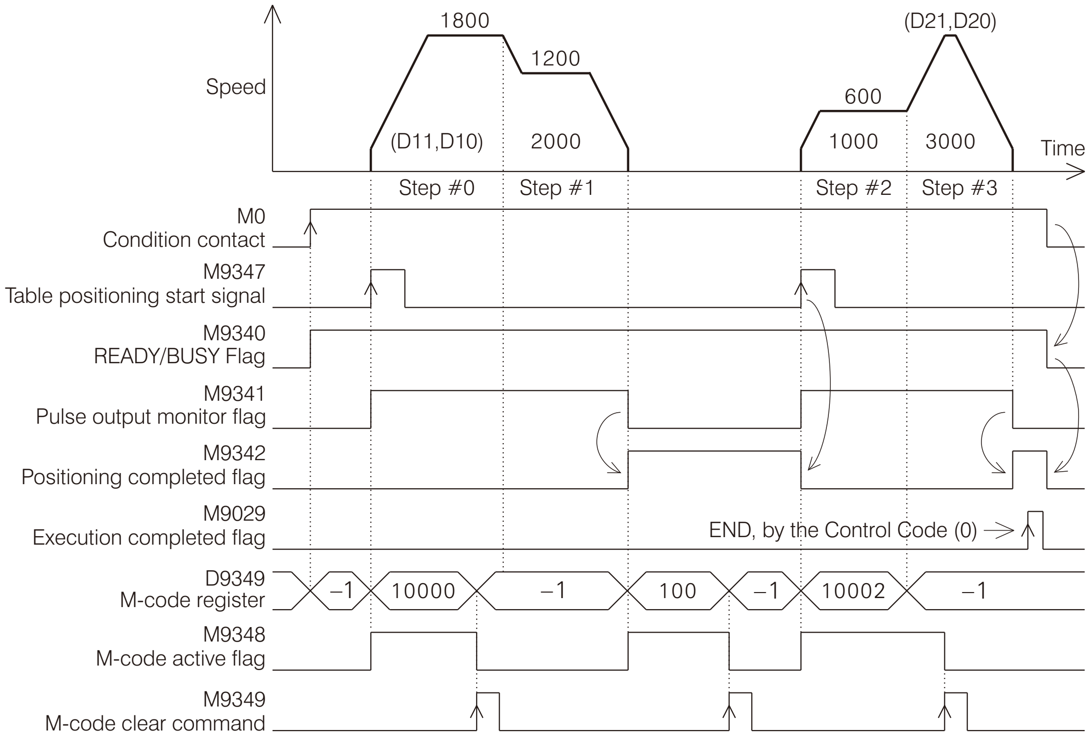
Step No.	Control Code	Position Data	Speed Data	M-code Data
0	Single relative positioning(2)	5,000	800	100
1	Single relative positioning(2)	10,000	1,200	10,000
2	END(0)	–1	–1	–1



The Continuous Relative Positioning and Continuous Absolute Positioning Functions

This example is using the continuous relative positioning

Step No.	Control Code	Position Data	Speed Data	M-code Data
0	Continuous relative positioning (3)	D10	1,800	10,000
1	Continuous relative positioning (3)	2,000	1,200	100※
2	Continuous relative positioning (3)	1,000	600	10,002
3	Continuous relative positioning (3)	3,000	D20	–1
4	END(0)	–1	–1	–1



※ When a step's M-code is assigned to the AFTER mode and it is finished, the positioning process is into the stand-by, will not go to the next step continuously. Must use the M-code clear command to clear its related active flag and register, then trigger the start signal again to go to the next step. Please refer to the M-code data on the next page for more details.

Position Data:

This data is related to the assigned Control Code. It is the moving distance for a relative positioning or the moving target for an absolute positioning.

At the relative positioning, it is used to assign the moving distance. If the moving distance > 0, will control the direction to increase the Present Value. If the moving distance < 0, will control the direction to decrease the Present Value.

At the absolute positioning, it is used to assign the moving target. This target is an absolute point relative to the zero point, thus the error = (moving target – Present Value PV). If the error > 0, will control the direction to increase the Present Value. If the error < 0, will control the direction to decrease the Present Value.

At the jump (JMP) function, this Position Data is used to appoint the step number of the jumping destination.

This Position Data is a 32-bit data, can use a constant number K or register D, R. When the register is used, that will occupy a pair of D or R. During the instruction is in operation, to change this data will be regarded as invalid.

Speed Data:

It is used to assign the operating speed. The available range is from the bias speed (D9342) to the maximum speed (D9341, D9340).

When the operating speed < the bias speed, it will be regarded as the bias speed; when the operating speed > the maximum speed, it will be regarded as the maximum speed.

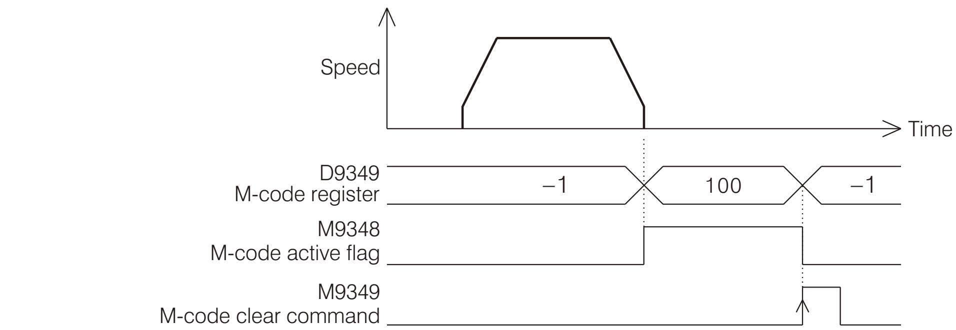
This Speed Data is a 32-bit data, can use a constant number K or register D, R. When the register is used, that will occupy a pair of D or R. During the instruction is in operation, to change this data could modify the real operating speed.

M-code Data:

The M-code Data is the source of the M-code register, that mechanism can be used to connect the executive status of the DTBL instruction with other applications. The available range at the M-code Data is –1～32,767. The DTBL instruction can give a number into the M-code register and turn “ON” the M-code active flag to tell a new event happening. By this mechanism, the user program can do the related control. After the step is finished and stand-by, should trigger the M-code clear command “OFF”→“ON” to clear the M-code register and the M-code active flag, hence the instruction is able to execute the next step.
In order to meet the actual requirements, the timing to give the M-code register and active flag can be separated to the AFTER and WITH modes.

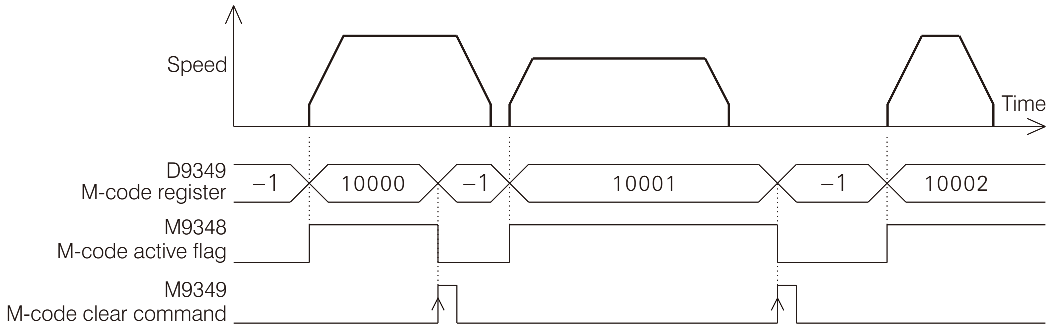
M-code Data	Description
–1	This step will let the M-code register = –1 (OFF). The mechanism of the M-code is disabled.
0～9,999	The step is using the AFTER mode. When this step is finished, will copy this M-code Data to the M-code register and turn “ON” the M-code active flag, then stand-by for a new start signal.
10,000～32,767	The step is using the WITH mode. When the step is started, will copy this M-code Data to the M-code register and turn “ON” the M-code active flag.

AFTER mode (M-code Data = 0~9,999)

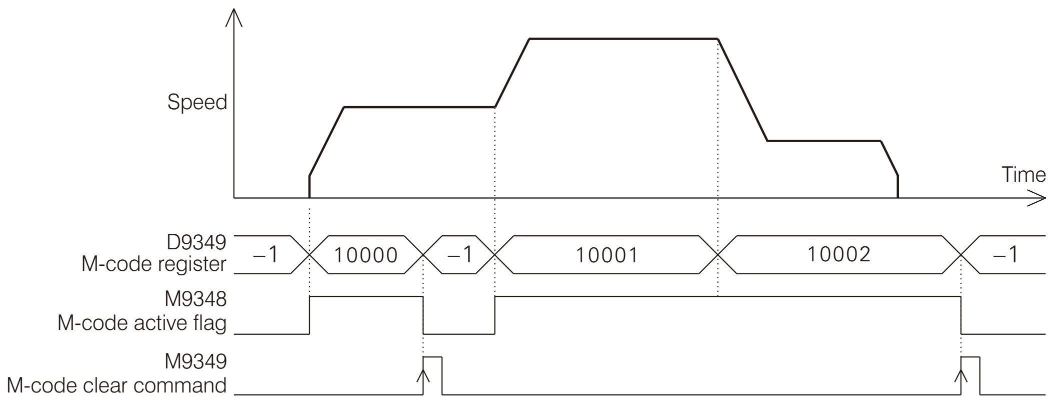


Assume that M-code Data is 100 (by the AFTER mode). When the step is finished, the M-code register changes from –1 to 100 and the M-code active flag turns “ON”.

WITH mode M-code (M-code Data = 10,000~32,767)



Assume the M-code Data for those 3 individual steps are 10,000, 10,001 and 10,002 (all by the WITH mode). When every step is started, the M-code register changes from –1 to the step's M-code Data and the M-code active flag turns “ON”. Besides, trigger the M-code clear command “OFF”→“ON” can clear the M-code register to –1 (OFF) and the M-code active flag “OFF”.



The picture above, those steps are continuous positioning and all by the WITH mode. The M-code register will automatically follow the steps to change its content at the beginning.

(S2) is the working area for the instruction.

(S2)	Description
D100	To appoint the starting step number of the DTBL instruction.
D101	Displays a step number at the table of the DTBL instruction which is being performed.
D102～D103	The working area is required for the system when this instruction is performed.

(D1) is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

(D2) is for to assign the output point of direction control signal.
When the output of direction control signal is “ON”, the motor moves forward; conversely, “OFF” moves reverse. Besides, the “ON”/“OFF” status of the direction control signal is decided by both the +/– numerical value of the distance to be move and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

- During the instruction is in operation, could change the value of the speed multiple ratio (D9348) to modify the real operating speed.
- When a single step or a series of continuous steps is finished, the Positioning completed flag M9342 will “ON”. After that, trigger the start signal M9347 = “OFF” → “ON” or turn “OFF” the instruction's condition contact will lead the completed flag M9342 “OFF”.
- If the condition contact M0 turns “OFF” during the pulse is output, the movement will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 “ON”.
- When the Control Code END (0) is executed, the Execution Completed Flag M9029 will “ON” for a Scan Time to indicate all the steps in this procedure table are finished. Therefore, the programmer can use the M9029 to turn “OFF” the condition contact of the DTBL instruction, then the output point could be used by another instruction.
- This instruction can be used as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction “OFF” could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = “ON”, that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is “ON”, the instruction will not execute.
If the M9340 = “OFF”, that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = “ON”.

M9342 is the positioning completed flag of the Y0. (M9362, M9382, M9402)
When a single step or a series of continuous steps is finished, the Positioning completed flag M9342 will “ON”.
After that, trigger the start signal M9347 = “OFF” → “ON” or turn “OFF” the instruction's condition contact will lead the completed flag M9342 “OFF” .

M9343 is the positioning abnormal stop flag of the Y0. (M9363, M9383, M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch before the Control Code END (0) is executed, that will cause Y0 stops to generate pulses and let the M9343 = “ON” to indicate this stop is abnormal. Turn “OFF” the condition contact of the activated instruction will lead the flag M9343 “OFF”.

M9345 is the flag to stop the Y0 generating pulse string (by gradually slow down). (M9365, M9385, M9405)
When the M9345 is turned “ON”, the pulse generating at the Y0 will gradually slow down then stop.
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9346 is the flag to imminently stop the Y0 generating pulse string. (M9366, M9386, M9406)
When the M9346 is turned “ON”, the pulse generating at the Y0 will stop imminently (without to slow down).
After the stop signal is discontinued, to make it move must release the positioning instruction then restart.

M9347 is the start signal for the Y0 to process the table positioning. (M9368, M9388, M9408)
To start a new (not by the continuous) step at the procedure table must trigger the Table positioning start signal M9347 = “OFF” → “ON”.

M9348 is the M-code active flag for to indicate the Y0's M-code is effective. (M9367, M9387, M9407)
When a step's designated event is happening, the axis's M-code active flag will turn “ON”.
Therefore, should trigger the relative M-code clear command “OFF” → “ON” to clear the M-code active flag “OFF”.

M9349 is the M-code clear command of the Y0. (M9369, M9389, M9409)

Trigger the M-code clear command "OFF"→"ON" can reset the M-code register become –1 (OFF) and the M-code active flag "OFF".

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).

When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9348 is the multiple ratio of the Y0 positioning speed. (D9368, D9388, D9408)

The actual operating speed at Y0 positioning is equal to the operating speed designated by the positioning instruction multiplied by the value of the multiple ratio.
During the instruction is operating, to change the ratio could modify the Y0's actual operating speed.
Its available value range to be set is 1~30,000 (Unit: 0.1%), therefore the speed multiple ratio is 0.1%~3,000.0%.

D9349 is the M-code register of the Y0. (D9369, D9389, D9409)

When the DTBL instruction is"OFF"→"ON", the M-code register will become –1 to indicate there is not have a M-code.
When a step has a M-code Data and its indicated event is happening, the M-code Data will copy to the M-code register D9349, hence that provides a useful symbol for the related control.
After the additional control is finished and this instruction is stand-by, should trigger the relative M-code clear command "OFF"→"ON" to clear the M-code register to –1 (OFF), then a new step is allowed.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)

When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

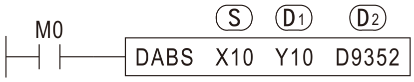
D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)

When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

FNC 313	DABS (S) (D1) (D2)	Absolute Current Value Read	1	2	M	3
			○	○	○	○

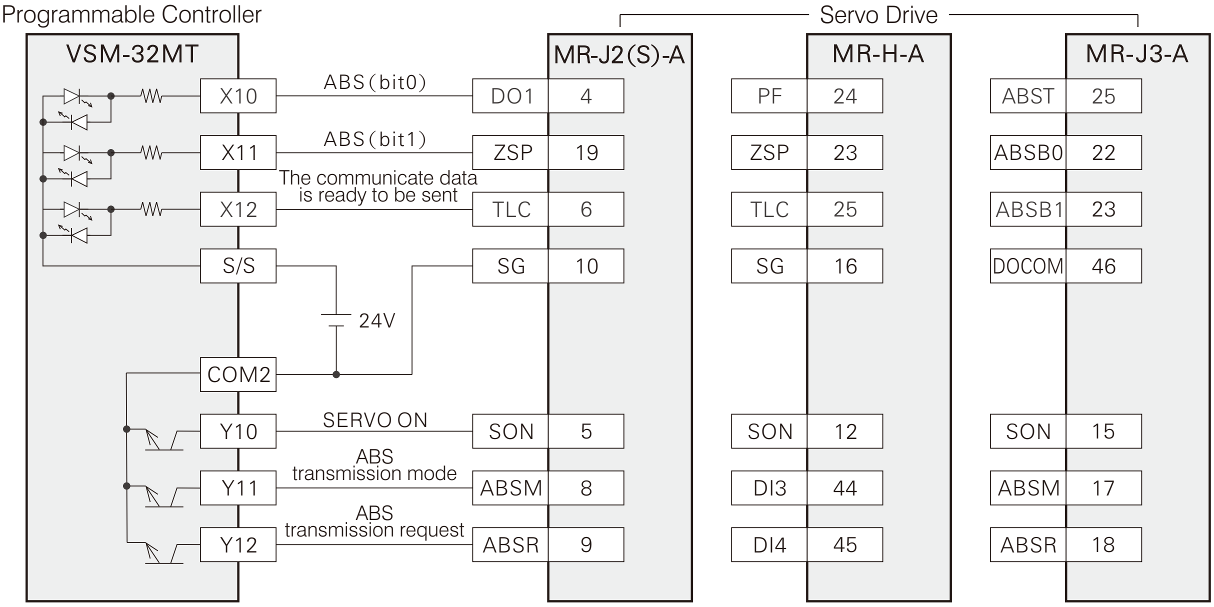
Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S	●	●	●	●	○	○												
D1		●	●	●	○	○												
D2								●	●	●	●	●	●	○	●			

• S occupies 3 components • D1 occupies 3 components • D2 occupies 2 components

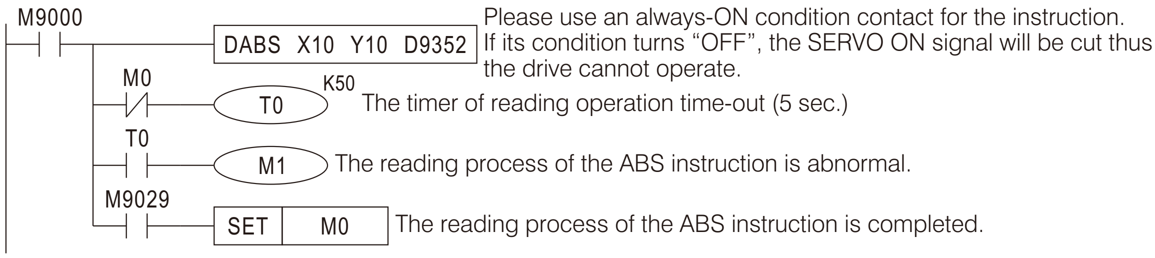


S : the points to receive the signal from the servo drive
D1 : the points to send the request signal to the servo drive
D2 : the device to store the decoded data (32-bit)

- The actual wiring for the example above can be illustrated with the diagram below:



- This instruction is for to read the Present Value of the built-in absolute positioning facility from a Mitsubishi servo drive.
- When the M0 turns from "OFF" to "ON", this ABS instruction starts to process the Present Value reading operation. When the operation is completed, the M9029 will turn "ON" for a Scan Time. If the M0 turns "OFF" during the operation, that will discontinue the reading action.
- The reading result of this instruction is from the current position register (by the pulse unit) in the servo drive, which is corresponded to the Present Value of the connected axis (by the pulse unit). Therefore, the (D2) is often assigned to the D9352 (PV of Y0 axis), D9372 (PV of Y1 axis), D9392 (PV of Y2 axis) or D9412 (PV of Y3 axis).
  When the (D2) is assigned to one of the PV special registers above, it will fill the read value into the appointed special register. As the example above, that (D2) D9352 is the lower 16-bit of the Y0's PV register. When the instruction is completed, the 32-bit position data read from the connected servo drive will be stored into (D9353, D9352) and (D9355, D9354). After that, the position data between the PLC's Y0 and the servo drive will be equal.
- This instruction is a 32-bit instruction. Therefore, be sure to input DABS in the program.
- This instruction can only be used once in the program.
- Program example: (assume this servo drive is connected with the Y0 axis)



FNC 314	MPG (S1) (S2) (D1) (D2)	Handwheel Positioning	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S1																○		
S2													○					
D1		●																
D2		●	●	●	○	○												

• S1 = 0~3 • S2 occupies 3 components • D1 = Y0~Y3 • If the D2 is assigned to a Y, it must use Y0~Y7

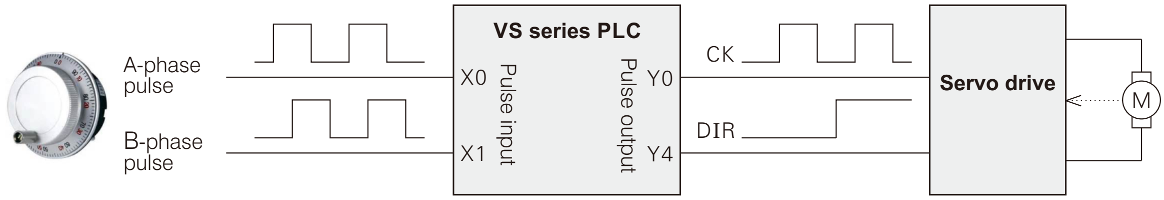
```
M0          (S1) (S2)  (D1) (D2)
-| |-----[ MPG  K0  D100  Y0  Y4 ]
```

S1 : to assign the combination of pulse input points

S2 : to assign work parameters, occupying 3 registers in total

D1 : the output point of generated pulse string

D2 : the output point of direction control signal



• When M0 = "ON", the instruction is activated. As the diagram above, the A/B phase pulses from an electronic handwheel are connected to a pair of the PLC's high speed input points (X0~X7). When the handwheel is rotated, this instruction is based on the input pulses' speed and quantity to multiply with the electronic gear ratio then generates a pulse string from the output point in proportion.

Number of pulses output = Number of pulses input × (Electronic gear ratio's numerator divided by its denominator)
= (Number of pulses input × D101) / D102

(S1) is for to assign the combination of pulse input points. It can be appoint to a number between 0~3.

(S1)	K0	K1	K2	K3
A-phase pulses input point	X0	X3	X2	X6
B-phase pulses input point	X1	X4	X5	X7

(S2) is the head register to assign the operational parameters of the instruction.

(S2)	Description
D100	The response time of the handwheel inputs. (Unit: ms.)
D101	The numerator of the electronic gear ratio
D102	The denominator of the electronic gear ratio

The response time: Can be appointed to 1~500 ms.; if the set value is exceeded, it will be regarded as 5 ms.

The response time refers to the delay between pulse input and pulse output. If the response time is set too short, it may cause machinery vibration. In this case, can adjust it to a longer response time to improve that.

The electronic gear ratio:

Number of pulses output = Number of pulses input × Electronic gear ratio

$$= \text{Number of pulses input} \times \frac{\text{The numerator of the electronic gear ratio}}{\text{The denominator of the electronic gear ratio}}$$

The available range of the numerator of the electronic gear ratio is 1~32,767; it will be regarded as 1 if the value exceeds the range.

The available range of the denominator of the electronic gear ratio is 1~32,767; it will be regarded as 1 if the value exceeds the range.

(D1) is for to assign the output point of generated pulse string.
It can only appoint to a point between Y0~Y3, and must use a transistor or line driver output Main Unit.

(D2) is for to assign the output point of direction control signal.
When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by the input order of A/B phase signals. If the A-phase is input earlier than the B-phase, the direction control signal turns "ON" to make the motor forward.

- The pulse output frequency of this instruction is proportionate to the frequency of pulses input and the electronic gear ratio only, other parameters are ineffectual. The positioning speed multiple ratio will not affect to the output frequency.
- The number of output pulses of this instruction is proportionate to the number of input pulses and the electronic gear ratio only, other parameters are ineffectual.
- When the related forward / reverse limit switch is activated during the MPG instruction generates pulses, that will cause the imminently stop.
- The condition contact turns "OFF" during the MPG instruction generates pulses, it will slow down then stop.
- This instruction can be written as many times as required and the points Y0~Y3 can be used to generate pulses individually at the same time.
- The example below uses Y0 output axis for the description of this instruction since the output axes Y1~Y3 have the same function. The three devices between parentheses ( ) indicate those are corresponding to Y1~Y3 respectively.

M9340 indicates the Y0 is occupied by a positioning instruction. (M9360, M9380, M9400)
Any instruction uses the Y0 to generate pulse string and it is already activated, that will occupy the resource of Y0 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource. Thus, before to activate this instruction should confirm the state of M9340.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9340 = "OFF", that indicates the Y0 is in the READY status (its resource is free to use), the Y0 can be used for this instruction.

M9341 is the pulse output monitor of the Y0. (M9361, M9381, M9401)
When the Y0 is generating pulses, M9341 = "ON".

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When the forward / reverse limit switch is active, that limits all the forward / reverse actions. That makes the forward / reverse motor to slow down and stop.
At this moment, only the moving request of opposite direction can be accepted.

D9351, D9350 display the positioning current speed of Y0. (DD9370, DD9390, DD9410)
When the instruction is active, this 32-bit special register shows the Present Value of operating speed.

D9355, D9354 display the positioning location (Present Value PV) of Y0. (DD9374, DD9394, DD9414)
When the power is turned from "OFF" to "ON", the initial value of this 32-bit special register is set as 0. Afterward, to operate any related positioning control instruction will affect to its content, it shows the current location of Y0.

- Program example:

  This example uses an electronic handwheel which with an X/Y axis switch and an x1 / x10 / x100 switch to perform the 2-axis handwheel control.

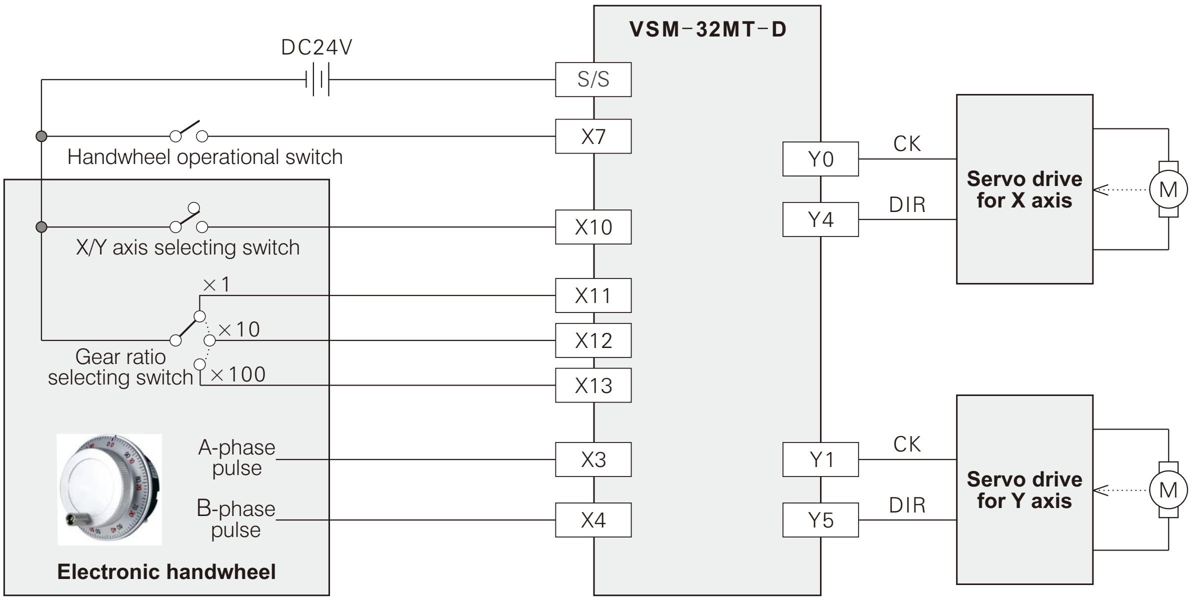
  Connect the MPG control operational switch to the input X7.

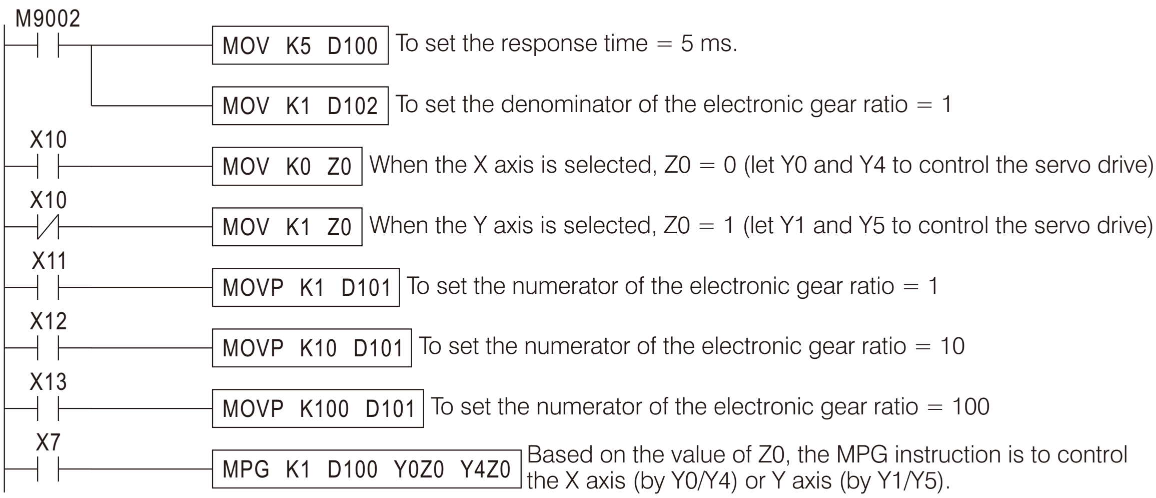
  Connect the A-phase signal of the handwheel to the input X3.

  Connect the B-phase signal of the handwheel to the input X4.

  Connect the X or Y axis selecting switch to the input X10. If the X10 = ON, then the MPG instruction is used to drive the X axis, else the X10 = OFF is to drive the Y axis.

  Connect the x1 / x10 / x100 gear ratio selecting switch to inputs X11, X12 and X13.





FNC 315	D LIR (S) (D)	Relative Linear Interpolation	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●					
D																○		

- The 16-bit instruction, S occupies 9 components
- The 32-bit instruction, S occupies 18 components
- D = 0 or 1 (it uses Y0, Y1, Y4 and Y5 if D = 0; it uses Y2, Y3, Y6 and Y7 if D = 1)

```
 M0          (S)    (D)
|—| |—————[LIR  D1000  K0]
```

S : the head register of the parameter block

D : the parameter to set the executing output points

- This instruction is for to reach the linear interpolation positioning between related two axes at a transistor or line driver PLC. When the instruction starts to operate, the two axes start to generate pulses at the same time then those will respectively produce the correspond numbers of pulses to achieve the axes' distance to be moved. After the instruction reaches the target and stops the pulse generating, this positioning is completed.
- When the M0 turns “ON”, this relative instruction will move the position from the start-up point (X^0, Y^0) to the target point (X^0+D1003, Y^0+D1004). Thus, the output points Y0 (pulse string for X axis) and Y1 (pulse string for Y axis) separately use the parameters of D1000 (Composite Initial speed), D1001 (Composite Operating speed) and D1002 (Acceleration / Deceleration time) to generate pulses simultaneously. Also, the points Y4 (direction signal for X axis) and Y5 (direction signal for Y axis) will control the moving directions of the axes.

Description of the parameter block (S) :

16-bit Instruction	32-bit Instruction	Description	Input Method
(S)	(S)+1, (S)	Composite initial speed Available range: 0 ~ 20k (Hz). If the set value < 0, will be regarded as 0; if the set value > 20k, will be regarded as 20k.	Designated by user program
(S)+1	(S)+3, (S)+2	Composite operating speed Available range: 10 (Hz) ~ the maximum speed. If the set value < 10, will be regarded as 10Hz; if the set value > the maximum speed, will be regarded as the maximum speed. The maximum speed at the VS1/VS2 is 50kHz; VSM/VS3 is 200kHz; VSM-28ML is 1MHz.	
(S)+2	(S)+5, (S)+4	Acceleration / Deceleration time (Available range: 0~32,000 ms.)	
(S)+3	(S)+7, (S)+6	Distance to be moved at the X-axis	
(S)+4	(S)+9, (S)+8	Distance to be moved at the Y-axis	
(S)+5	(S)+11, (S)+10	Bias speed at the X-axis	Result storage area, produced by the LIR instruction
(S)+6	(S)+13, (S)+12	Operating speed at the X-axis	
(S)+7	(S)+15, (S)+14	Bias speed at the Y-axis	
(S)+8	(S)+17, (S)+16	Operating speed at the Y-axis	

The designated composite initial speed must less than or equal to the composite operating speed.

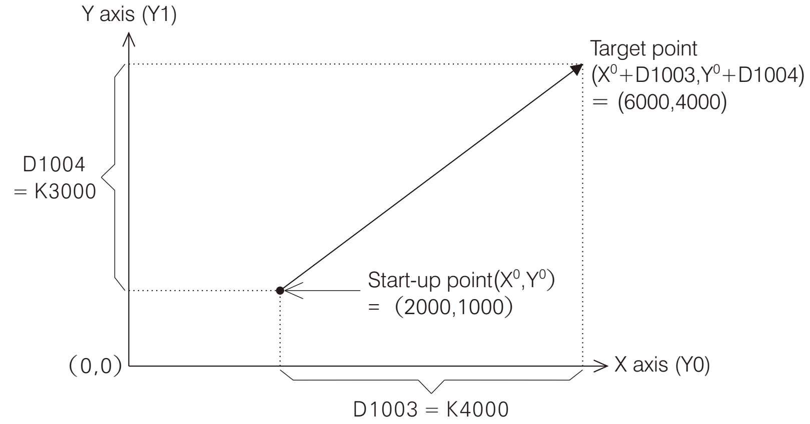
The axis will not generate pulse if that calculated operating speed is less than 1 Hz.

Description of the parameter (D) that is used to set the executing output points:

(D)	X axis		Y axis	
	Pulse string output	Direction signal output	Pulse string output	Direction signal output
K0	Y0	Y4	Y1	Y5
K1	Y2	Y6	Y3	Y7

When the output of direction control signal is “ON”, the motor moves forward; conversely, “OFF” moves reverse. Besides, the “ON”/“OFF” status of the direction control signal is decided by both the +/ − numerical value of the axis's distance to be moved and the parameter of the rotational direction (increase Present Value when forward / backward)

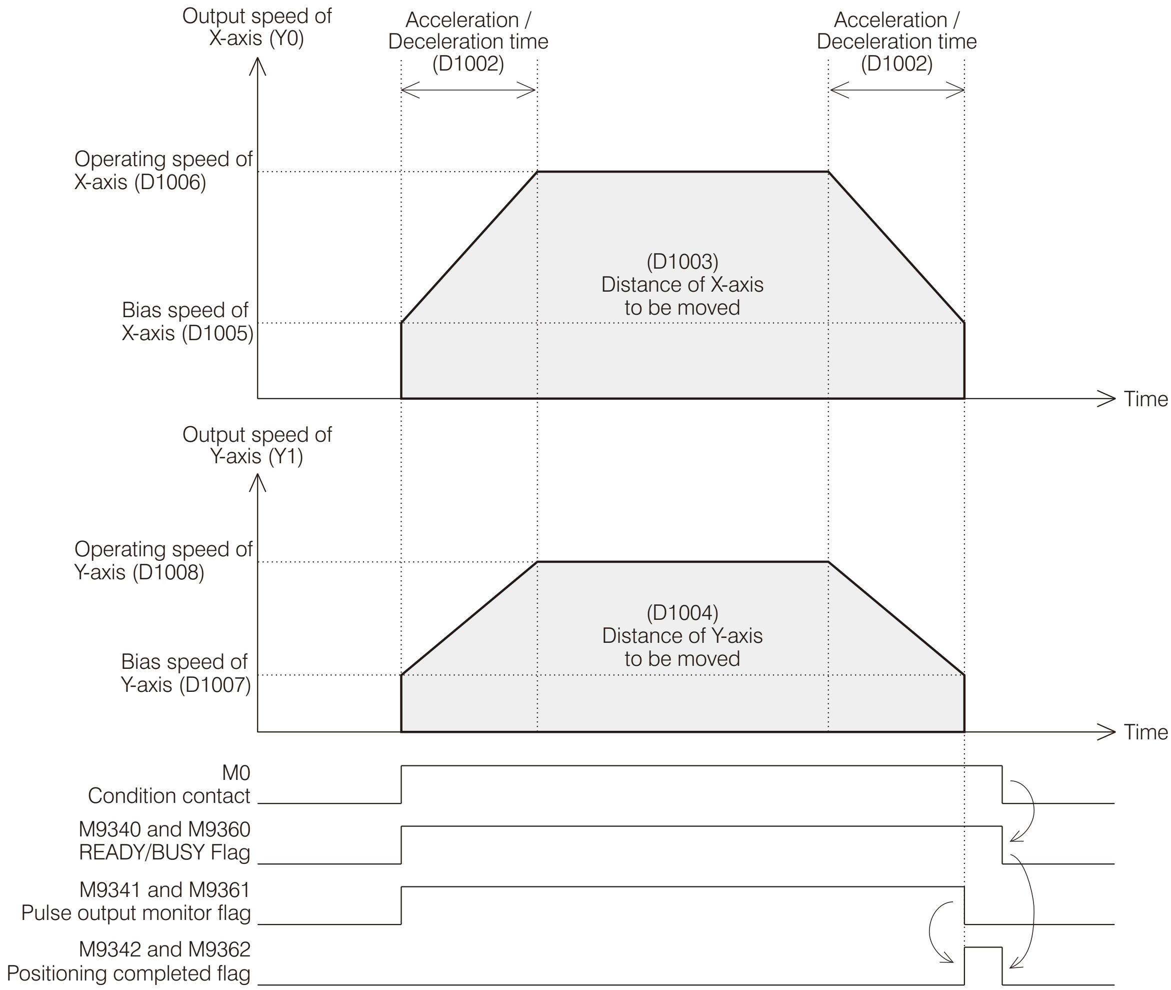
- For example, before the instruction executes, the current position CP of start-up point is $(X^0, Y^0) = (2000,1000)$, D1000 = K1000 (Composite initial speed Hz), D1001 = K3000 (Composite operating speed Hz), D1002 = K300 (Acceleration / Deceleration time ms.), D1003 = K4000 (Distance to be moved at the X-axis) and D1004 = K3000 (Distance to be moved at the Y-axis). When the M0 turns from "OFF" to "ON", this LIR instruction will compute the further parameters and fill D1005~D1008 up, then all the related signals of X-axis and Y-axis will start to output. Those points use the data at the parameter block to move the positioning path from (2000,1000) to (6000,4000), as shown below.



The formulas to get the data at D1005~D1008 and the relationship diagram between the 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 the instruction is in operation, to change any related parameter will be regarded as invalid. Therefore, all the necessary data at the parameter block S should be set up completely before it starts.
- When this instruction completes the movement of specific distances at both axes, that will let the positioning completed flag M9342 and M9362 both = "ON".
- If the condition contact M0 turns "OFF" during the pulses are output, the movement of both axes will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9342 or M9362 "ON" neither.
- The operation of this instruction will not be affected by the content values of the D9340~D9344, D9348, D9360~D9364, D9368, D9380~D9384, D9388, D9400~D9404 and D9408.
- When the distance to be moved at the X-axis and Y-axis are both equal to zero, this instruction will not execute.
- The instruction will occupy two pulse output points at the same time (D = K0 uses Y0+Y1; D = K1 uses Y2+Y3). The example below uses (Y0+Y1) output axes for the description of this instruction. Since the output axes (Y2+Y3) have the same function, the devices between parentheses () indicate those are corresponding to (Y2+Y3).

M9340/M9360 indicates that Y0/Y1 is occupied by a positioning instruction. (M9380/M9400)
Any instruction uses the Y0/Y1 to generate pulse string and it is already activated, that will occupy the resource of Y0/Y1 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource.
Thus, before to activate this instruction should confirm the states of M9340 and M9360 both.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9360 = "ON", that indicates the Y1 is BUSY to generate pulse string or other instruction still occupies the resource of the Y1. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If both the M9340 and M9360 = "OFF", that indicates the Y0 and Y1 are in the READY status (these resources are free to use), the Y0+Y1 can be used for this instruction.

M9341/M9361 is the pulse output monitor of the Y0/Y1. (M9381/M9401)
When the Y0/Y1 is generating pulses, both of the M9341/M9361 = "ON".

M9342/M9362 is the positioning completed flag of the Y0/Y1. (M9382/M9402)
When the positioning instruction is completed, both of the M9342/M9362 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342/M9362 "OFF".

M9343/M9363 is the positioning abnormal stop flag of the Y0/Y1. (M9383/M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0+Y1 positioning function, will cause both the Y0 and Y1 stop to generate pulses and let the M9343/M9363 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343/M9363 "OFF".

M9345/M9365 is the flag to stop the Y0 and Y1 generating pulse string (by gradually slow down). (M9385/M9405)
When either the M9345 or M9365 is turned "ON", the pulse generating at both the Y0 and Y1 will gradually slow down then stop.
After the stop signal is discontinued, to make the axes move must release the positioning instruction then restart.

M9346/M9366 is the flag to imminently stop the Y0 and Y1 generating pulse string. (M9386/M9406)
When either the M9346 or M9366 is turned "ON", the pulse generating at both the Y0 and Y1 will stop imminently (without to slow down).
After the stop signal is discontinued, to make the axes move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When either one of the related forward / reverse limit switch is activated, that limits both the actions of the Y0 and Y1 to make the motors slow down and stop.

DD9350/DD9370 display the positioning current speed of the Y0/Y1. (DD9390/DD9410)
When the instruction is active, those 32-bit special registers show the Present Value of operating speeds.

DD9354/DD9374 display the positioning location (Present Value PV) of the Y0/Y1. (DD9394/DD9414)
When the power is turned from "OFF" to "ON", the initial value of those 32-bit special registers are 0.
Afterward, to operate any related positioning control instruction will affect to the contents, they indicate the location of the Y0 and Y1.

FNC 316	D LIA (S) (D)	Absolute Linear Interpolation	1	2	M	3
			○	○	○	○

Operand	Devices																	
	X	Y	M	S	D.b	R.b	KnX	KnY	KnM	KnS	T	C	D,R	V,Z	UnG	K,H	E	"$"
S													●					
D																○		
• The 16-bit instruction, S occupies 9 components • The 32-bit instruction, S occupies 18 components																		
• D = 0 or 1 (it uses Y0, Y1, Y4 and Y5 if D = 0; it uses Y2, Y3, Y6 and Y7 if D = 1)																		

M0 —| |— [LIA D1000 K1] (S) (D)

S : the head register of the parameter block

D : the parameter to set the executing output points

- This instruction is for to reach the linear interpolation positioning between related two axes at a transistor or line driver PLC. When the instruction starts to operate, the two axes start to generate pulses at the same time then those will respectively produce the correspond numbers of pulses to achieve the X/Y axes location (Present Values) equal to target point. After the instruction reaches the target and stops the pulse generating, this positioning is completed.
- When the M0 turns "ON", this absolute instruction will move the position from the start-up point (X^0, Y^0) to the target point (D1003, D1004). Thus, the points Y2 (pulse string for X axis) and Y3 (pulse string for Y axis) separately use the parameters of D1000 (Composite Initial speed), D1001 (Composite Operating speed) and D1002 (Acceleration / Deceleration time) to generate pulses simultaneously. Also, the points Y6 (direction signal for X axis) and Y7 (direction signal for Y axis) will control the moving directions of the axes.

Description of the parameter block (S) :

16-bit Instruction	32-bit Instruction	Description	Input Method
(S)	(S)+1, (S)	Composite initial speed Available range: 0 ~ 20k (Hz). If the set value < 0, will be regarded as 0; if the set value > 20k, will be regarded as 20kHz.	Designated by user program
(S)+1	(S)+3, (S)+2	Composite operating speed Available range: 10 (Hz) ~ the maximum speed. If the set value < 10, will be regarded as 10Hz; if the set value > the maximum speed, will be regarded as the maximum speed. The maximum speed at the VS1/VS2 is 50kHz; VSM/VS3 is 200kHz; VSM-28ML is 1MHz.	
(S)+2	(S)+5, (S)+4	Acceleration / Deceleration time (Available range: 0~32,000 ms.)	
(S)+3	(S)+7, (S)+6	Target point at the X-axis The target is an absolute position that is related to the zero point.	
(S)+4	(S)+9, (S)+8	Target point at the Y-axis The target is an absolute position that is related to the zero point.	
(S)+5	(S)+11, (S)+10	Bias speed at the X-axis	Result storage area, produced by the LIA instruction
(S)+6	(S)+13, (S)+12	Operating speed at the X-axis	
(S)+7	(S)+15, (S)+14	Bias speed at the Y-axis	
(S)+8	(S)+17, (S)+16	Operating speed at the Y-axis	

The designated composite initial speed must less than or equal to the composite operating speed.

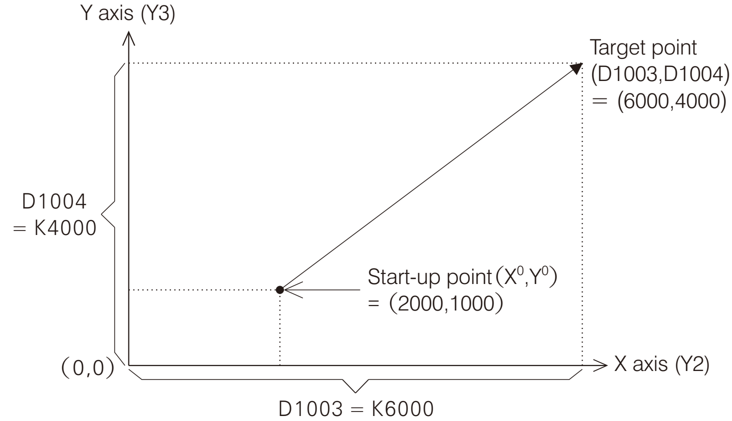
The axis will not generate pulse if that calculated operating speed is less than 1 Hz.

Description of the parameter (D) that is used to set the executing output points:

(D)	X axis		Y axis	
	Pulse string output	Direction signal output	Pulse string output	Direction signal output
K0	Y0	Y4	Y1	Y5
K1	Y2	Y6	Y3	Y7

When the output of direction control signal is "ON", the motor moves forward; conversely, "OFF" moves reverse. Besides, the "ON"/"OFF" status of the direction control signal is decided by both the +/ — numerical value of the axis's distance to be moved (Target point — Present value, PV) and the parameter of the rotational direction (increase Present Value when forward / backward) are executed.

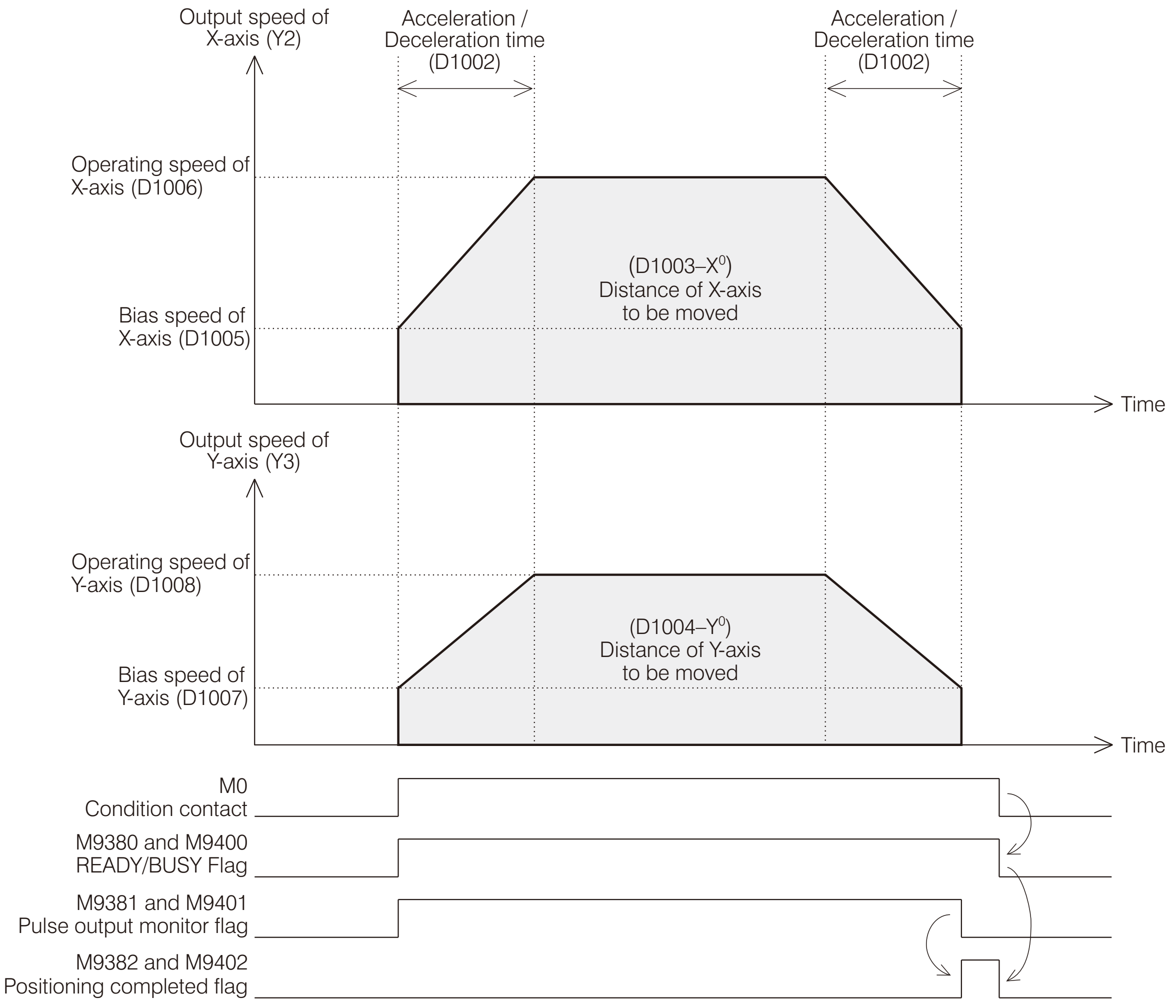
- For example, before the instruction executes, the current position CP of start-up point is $(X^0, Y^0) = (2000,1000)$, D1000 = K1000 (Composite initial speed Hz), D1001 = K3000 (Composite operating speed Hz), D1002 = K300 (Acceleration / Deceleration time ms.), D1003 = K6000 (Target point at the X-axis) and D1004 = K4000 (Target point at the Y-axis). When the M0 turns from “OFF” to “ON”, this LIA instruction will compute the further parameters and fill D1005~D1008 up, then all the related signals of X-axis and Y-axis will start to output. Those points use the data at the parameter block to move the positioning path from (2000,1000) to (6000,4000), as shown below.



The formulas to get the data at D1005~D1008 and the relationship diagram between the 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 the instruction is in operation, to change any related parameter will be regarded as invalid. Therefore, all the necessary data at the parameter block (S) should be set up completely before it starts.
- When this instruction completes the movement to the target point at both axes, that will let the positioning completed flag M9382 and M9402 both = "ON".
- If the condition contact M0 turns "OFF" during the pulses are output, the movement of both axes will be gradually slow down then stop. Thus, that will not to turn the positioning completed flag M9382 or M9402 "ON" neither.
- The operation of this instruction will not be affected by the content values of the D9380~D9384, D9388, D9400~D9404, D9408, D9420~D9424, D9428, D9440~D9444 and D9448.
- When the distance to be moved at the X-axis and Y-axis are both equal to zero (Start-up point = Target point), this instruction will not execute.
- The instruction will occupy two pulse output points at the same time ((D) = K0 uses Y0+Y1; (D) = K1 uses Y2+Y3). The example below uses (Y0+Y1) output axes for the description of this instruction. Since the output axes (Y2+Y3) have the same function, the devices between parentheses () indicate those are corresponding to (Y2+Y3).

M9340/M9360 indicates that Y0/Y1 is occupied by a positioning instruction. (M9380/M9400)
Any instruction uses the Y0/Y1 to generate pulse string and it is already activated, that will occupy the resource of Y0/Y1 axis consistently. However, turn the condition contact of this activated instruction "OFF" could release the resource.
Thus, before to activate this instruction should confirm the states of M9340 and M9360 both.
If the M9340 = "ON", that indicates the Y0 is BUSY to generate pulse string or other instruction still occupies the resource of the Y0. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If the M9360 = "ON", that indicates the Y1 is BUSY to generate pulse string or other instruction still occupies the resource of the Y1. Even though, the condition contact of this instruction is "ON", the instruction will not execute.
If both the M9340 and M9360 = "OFF", that indicates the Y0 and Y1 are in the READY status (these resources are free to use), the Y0+Y1 can be used for this instruction.

M9341/M9361 is the pulse output monitor of the Y0/Y1. (M9381/M9401)
When the Y0/Y1 is generating pulses, both of the M9341/M9361 = "ON".

M9342/M9362 is the positioning completed flag of the Y0/Y1. (M9382/M9402)
When the positioning instruction is completed, both of the M9342/M9362 = "ON".
Turn "OFF" the condition contact of the completed instruction will lead the flag M9342/M9362 "OFF".

M9343/M9363 is the positioning abnormal stop flag of the Y0/Y1. (M9383/M9403)
Trigger any related compulsive stop flag or the forward / reverse limit switch during the unfinished Y0+Y1 positioning function, will cause both the Y0 and Y1 stop to generate pulses and let the M9343/M9363 = "ON" to indicate this stop is abnormal.
Turn "OFF" the condition contact of the activated instruction will lead the flag M9343/M9363 "OFF".

M9345/M9365 is the flag to stop the Y0 and Y1 generating pulse string (by gradually slow down). (M9385/M9405)
When either the M9345 or M9365 is turned "ON", the pulse generating at both the Y0 and Y1 will gradually slow down then stop.
After the stop signal is discontinued, to make the axes move must release the positioning instruction then restart.

M9346/M9366 is the flag to imminently stop the Y0 and Y1 generating pulse string. (M9386/M9406)
When either the M9346 or M9366 is turned "ON", the pulse generating at both the Y0 and Y1 will stop imminently (without to slow down).
After the stop signal is discontinued, to make the axes move must release the positioning instruction then restart.

Forward / Reverse limit switch (under the "Positioning Parameter Setup" of the Ladder Master S).
When either one of the related forward / reverse limit switch is activated, that limits both the actions of the Y0 and Y1 to make the motors slow down and stop.

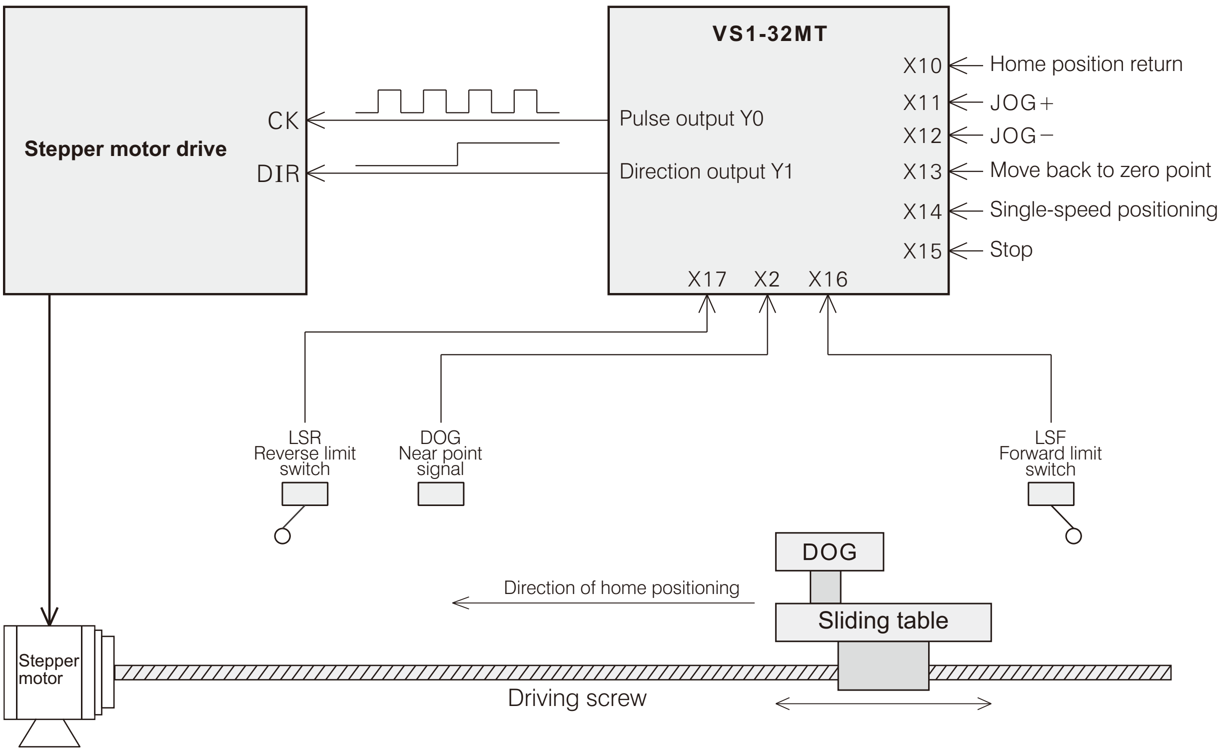
DD9350/DD9370 display the positioning current speed of the Y0/Y1. (DD9390/DD9410)
When the instruction is active, those 32-bit special registers show the Present Value of operating speeds.

DD9354/DD9374 display the positioning location (Present Value PV) of the Y0/Y1. (DD9394/DD9414)
When the power is turned from "OFF" to "ON", the initial value of those 32-bit special registers are 0.
Afterward, to operate any related positioning control instruction will affect to the contents, they indicate the location of the Y0 and Y1.

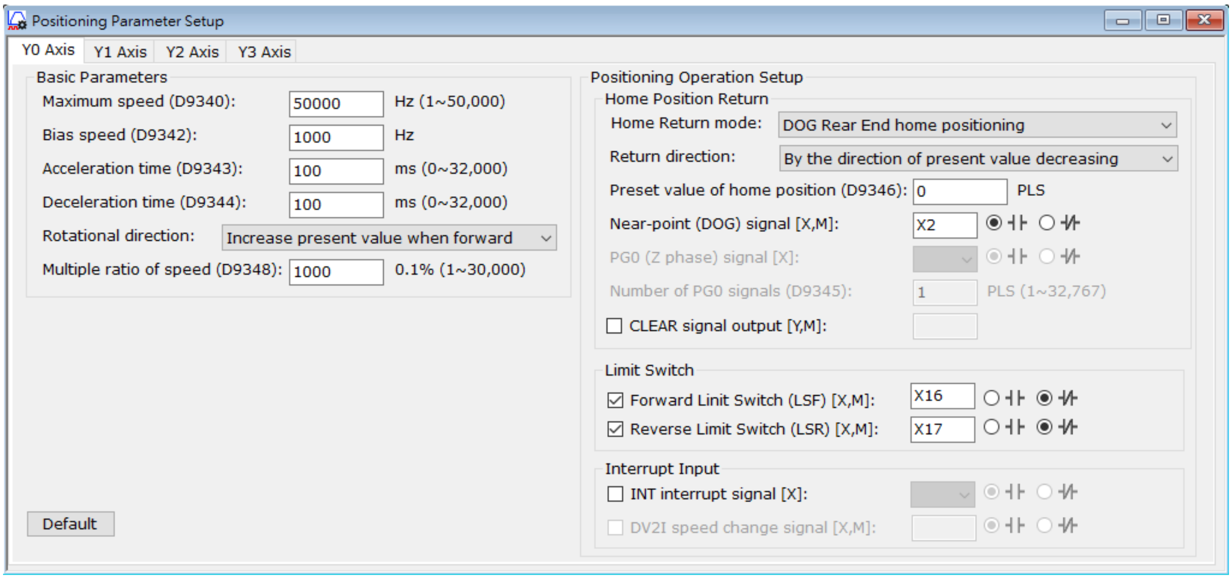
8-4 Positioning Program Example

8-4-1 Positioning Program Example for the VS1 or VS2 Series PLC

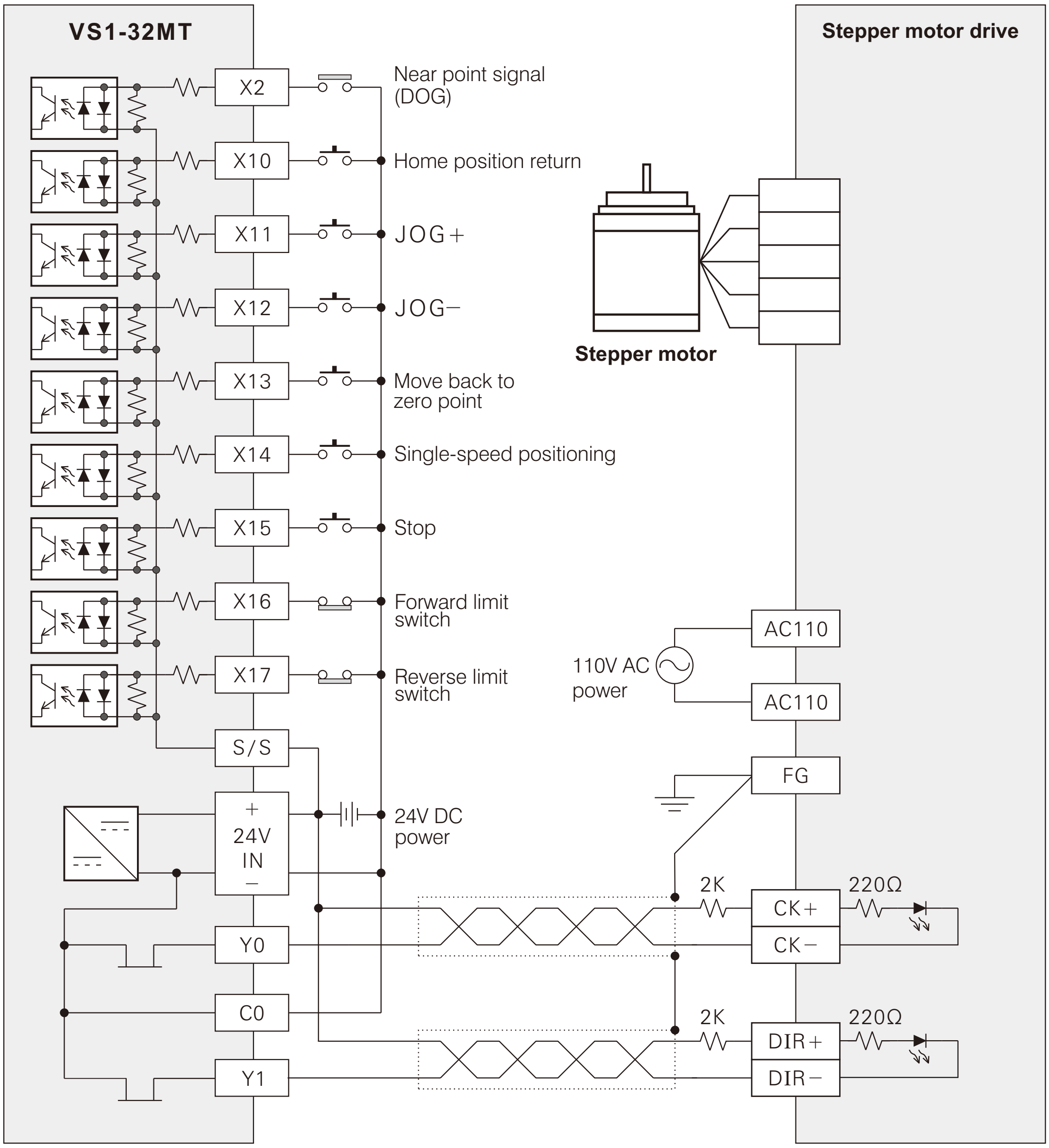
This example uses the combination of the VS1 or VS2 transistor Main Unit and a stepper motor drive to complete a positioning control system. This control example carries out the home position return, forward JOG, reverse JOG and single-speed positioning functions. The brief diagram of the system is shown below.

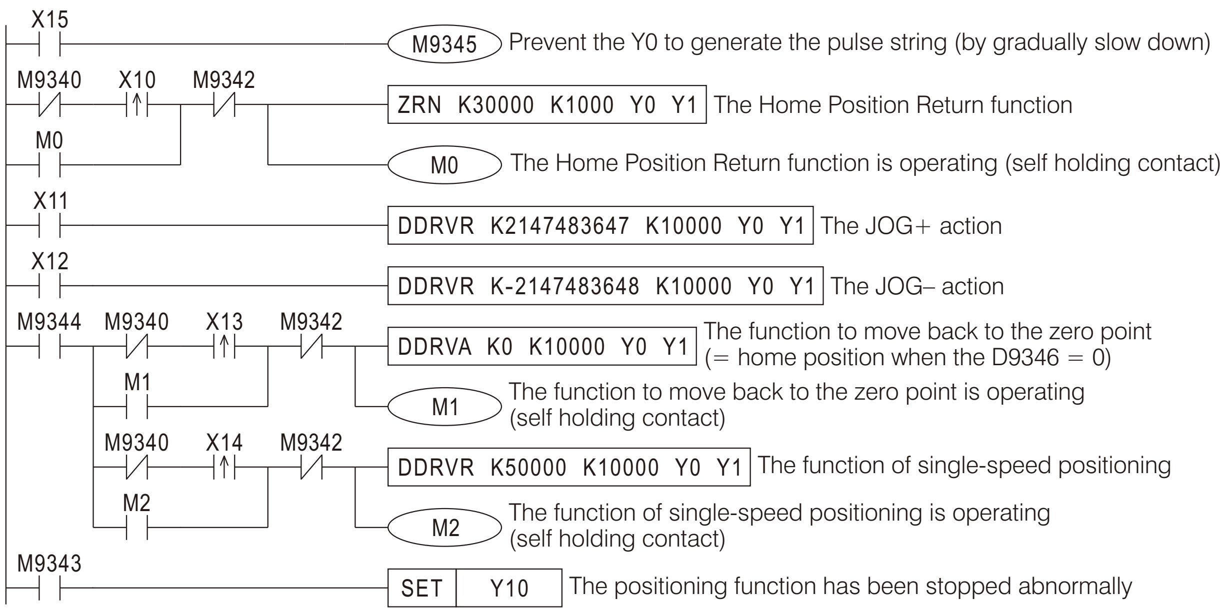


Use the "Positioning Parameter Setup" function which is provided by the programming tools Ladder Master S to set the relevant parameters about the Y0 axis. Select the "DOG Rear End home positioning" as the home return mode. Then the settings will be written onto the PLC with the project.



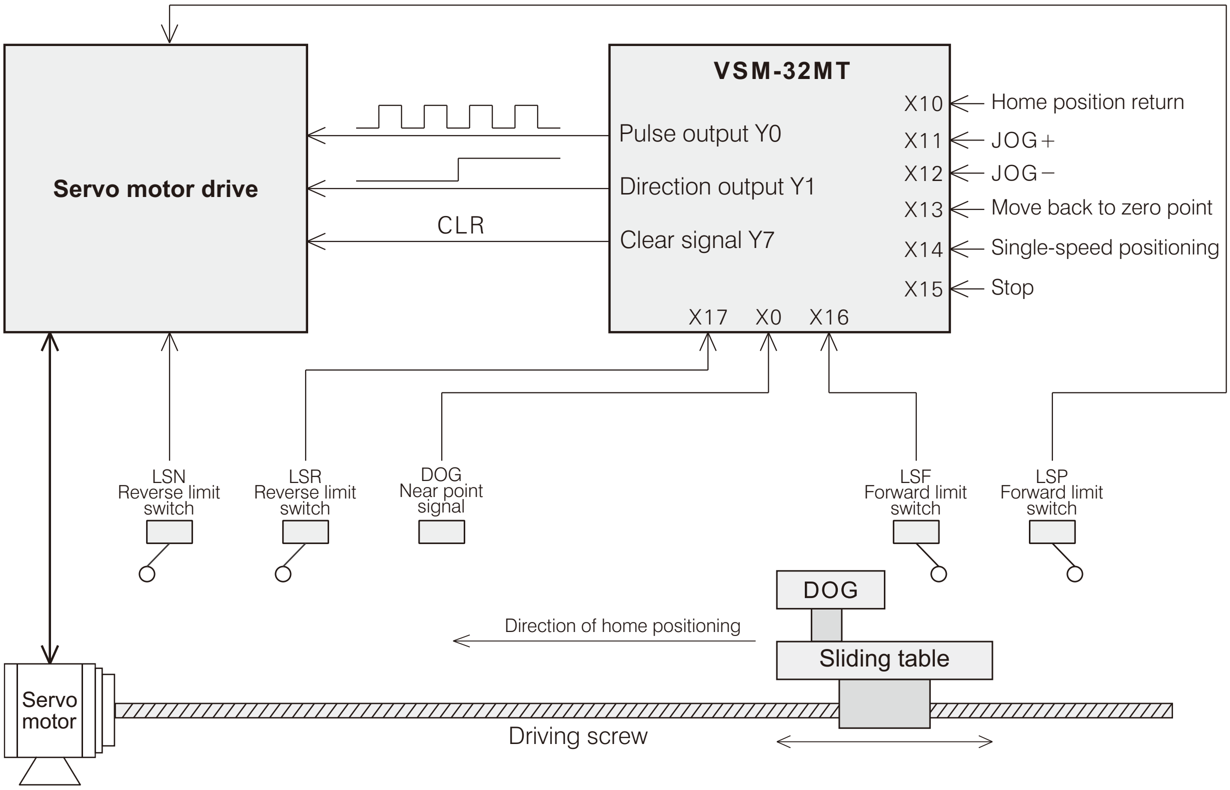
The wiring example between the VS1 transistor Main Unit and a stepper servo drive



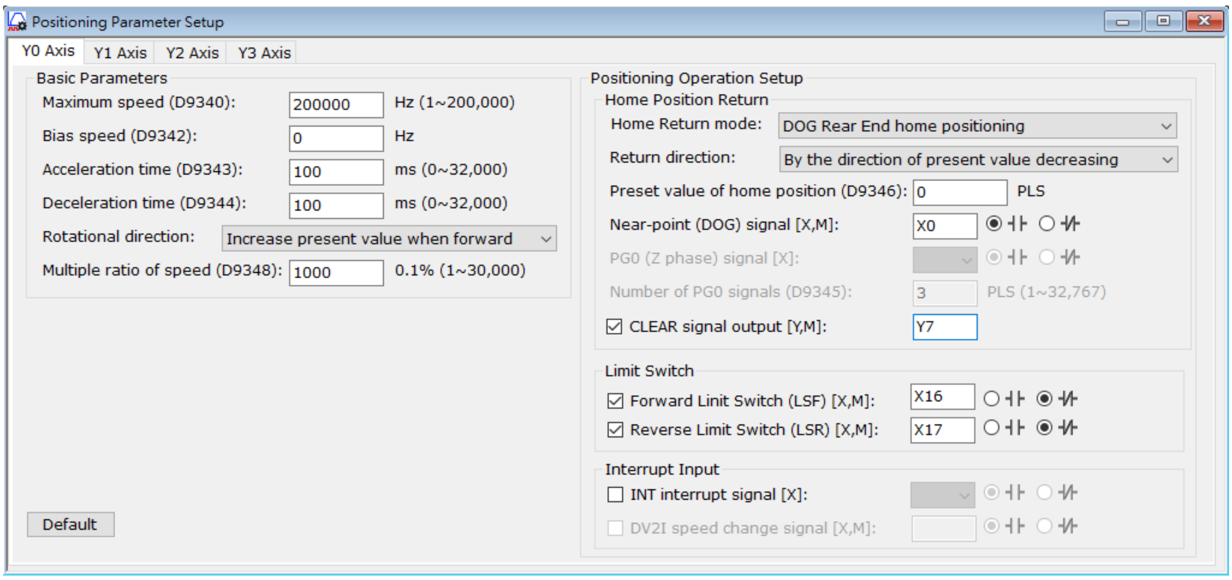


8-4-2 Positioning Program Example for the VSM or VS3 Series PLC

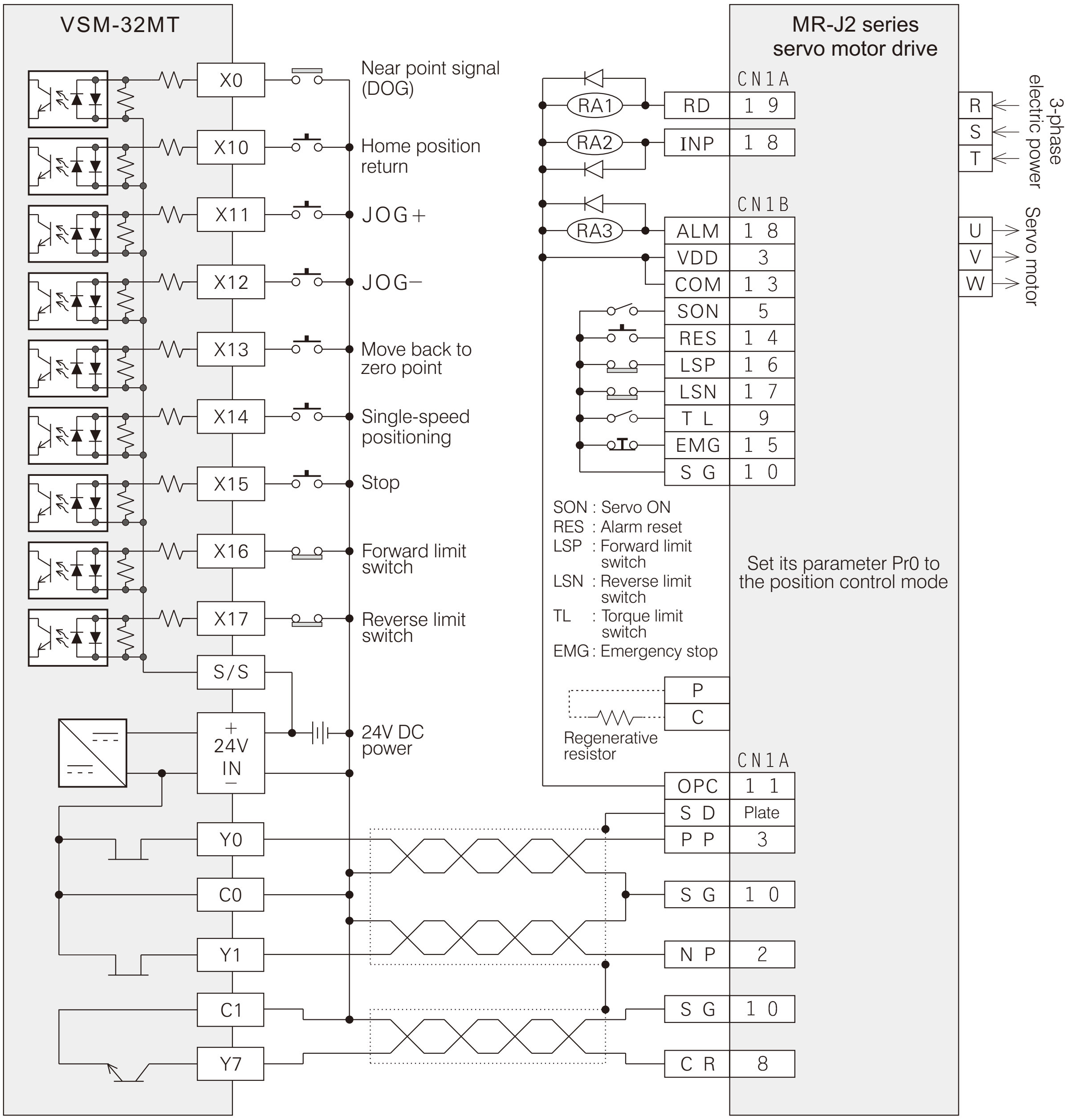
This example uses the combination of the VSM or VS3 transistor Main Unit and a Mitsubishi servo drive (MR-J2) to complete a positioning control system. This control example carries out the home position return, forward JOG, reverse JOG and single-speed positioning functions. The brief diagram of the system is shown below.

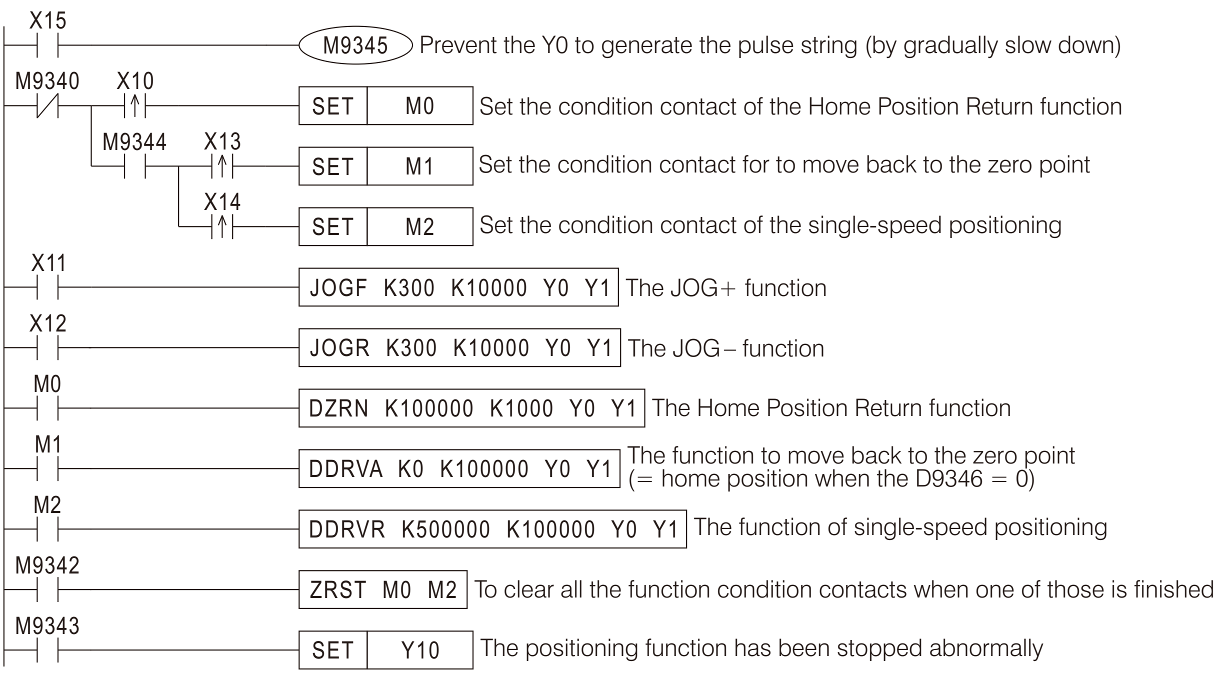


Use the "Positioning Parameter Setup" function which is provided by the programming tools Ladder Master S to set the relevant parameters about the Y0 axis. Select the "DOG Rear End home positioning" as the home return mode. Then the settings will be written onto the PLC with the project.



The wiring example between the VSM transistor Main Unit & Mitsubishi servo drive (MR-J2)





8-4-3 8 Axes Positioning Program Example

One VS series transistor or line driver Main Unit can support 4 axes position control. If the control is required more than 4 axes, one unit is not enough. However, we can use the character of the CPU LINK communication to immediately transfer data to other PLCs. At the following example, that links two VS series PLCs to perform an 8 axes positioning. In the same way, to control more axes are available.

The master station #0 is the main PLC to handle the communication procedure and make the positioning command, the Station No. 1 become the real output PLC to control #5~#8 axes.

There are 3 sections to explame the control system.

Section 1: To plan the components which will be used for the data transfer between PLCs.
At the project of the master Station No. 0 to establish the CPU LINK communication table. The table is to designate components to send the data from Station No. 0 to 1 and some are about to send data from No. 1 to 0. This example uses the D100~D114 to send the data from Station No. 0 to 1 for the #5 axis, and the D115~D119 to send the feedback data from Station No. 1 to 0. The list below shows the related components will be used by the #5~#8 axes.

The configuration for #5~#8 axes:

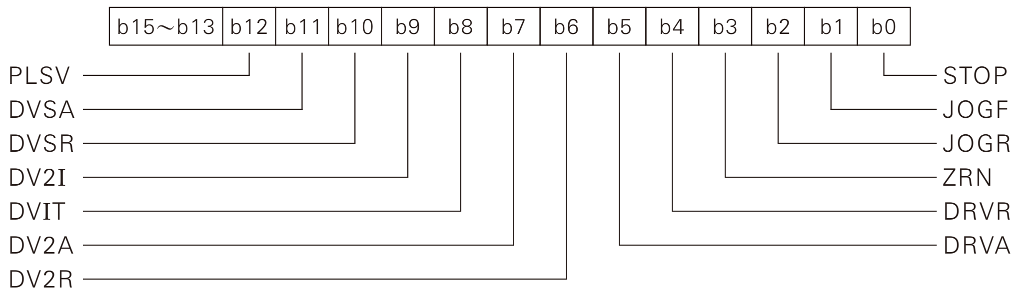
Axis No.	From No. 0 to 1	From No. 1 to 0
#5 axis	D100~D114	D115~D119
#6 axis	D120~D134	D135~D139
#7 axis	D140~D154	D155~D159
#8 axis	D160~D174	D175~D179

By the purposes to establish the communication table “CPUL0”:

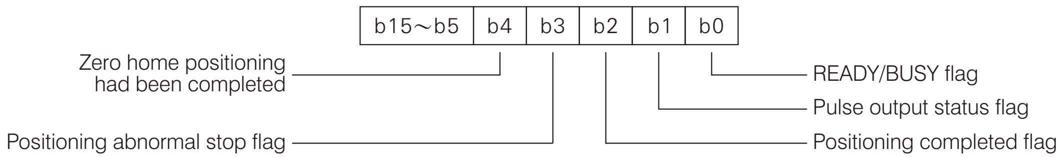
Item No.	Station No.	Device Range	Word / Bit	Disable Contact
1	0	D100 – D114	—	—
2	1	D115 – D119	—	—
3	0	D120 – D134	—	—
4	1	D135 – D139	—	—
5	0	D140 – D154	—	—
6	1	D155 – D159	—	—
7	0	D160 – D174	—	—
8	1	D175 – D179	—	—

The transfer purposes of related components: (below are for the #5 axis only, the rest may be inferred by analogy)
(D101, D100) — Set JOG speed
(D103, D102) — Set ZRN (Home Positioning) speed
(D105, D104) — Set ZRN (Home Positioning) creep speed
(D107, D106) — Set target position #1
(D109, D108) — Set target position #2
(D111, D110) — Set operation speed #1
(D113, D112) — Set operation speed #2

D114 — Operation command

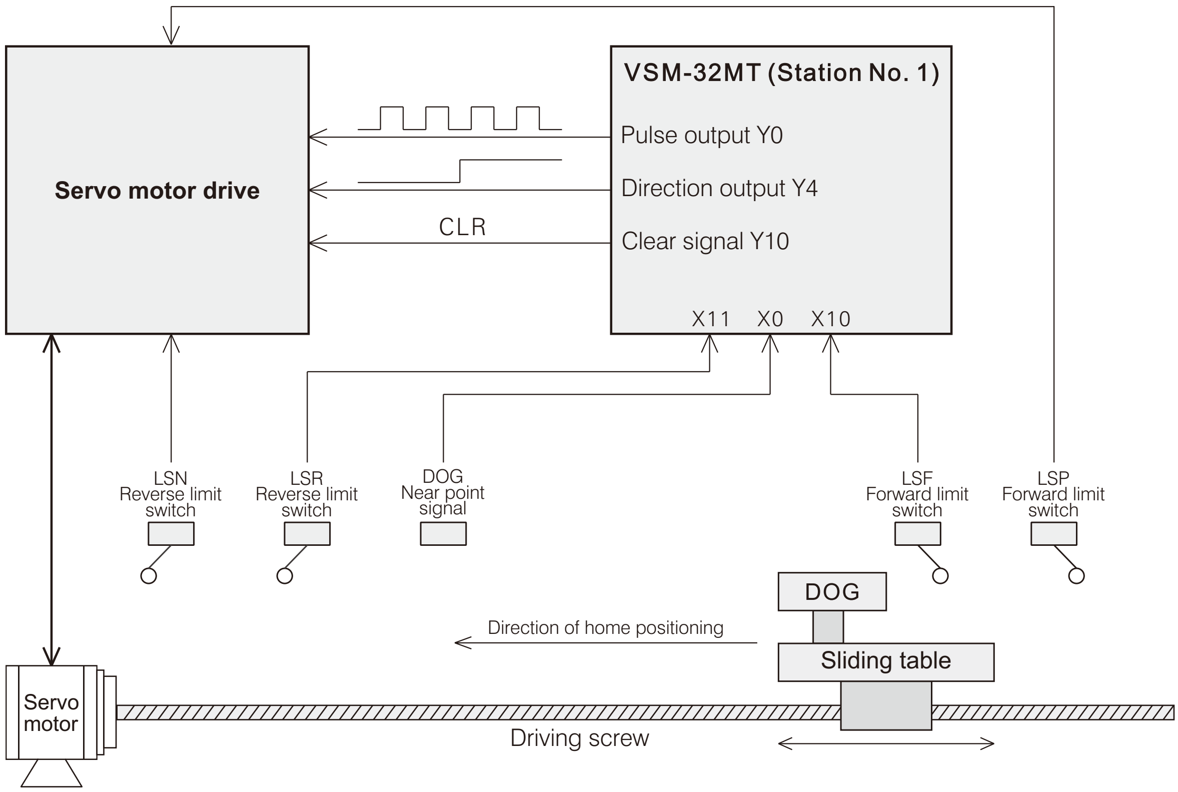


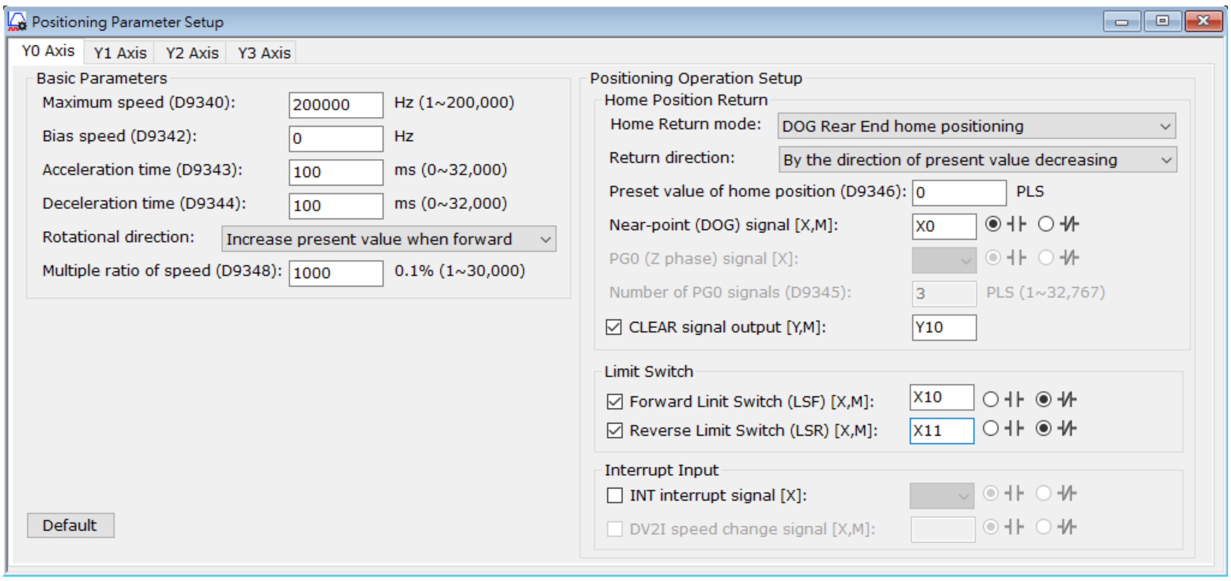
D115 — Operation status feedback



(D117, D116) — Current speed feedback
(D119, D118) — Current location feedback

Section 2: To complete the related setting at the Station No. 1 for the #5~#8 axes.
According to the positioning mission to fill in the parameters of the Y0~Y3 (#5~#8 axes).
The diagram and picture are the brief wiring and setup example for the Y0 (#5 axis).





And then, edit the user program to control each axis.
Since the control procedure is according to the command from the Station No. 0, the program in this Station No. 1 is to receive the command then generates the real control pulses. Also, the construct for all the Y0~Y3 are the same, the tiny differences are the allocated components (please refer to the previous Section 1). This brief positioning example is for to control the Y0 (#5 axis) only, but user can use the same shape to finish other positioning axis.

Y0 uses D100~D119
DD100=JOG spd; DD102=ZRN spd; DD104=ZRN creep spd; DD106=Target#1; DD108=Target#2;
DD110=Spd#1; DD112=Spd#2; D114=Command; D115=Status; DD116=Current spd; DD118=Current location

D115 delivers the operation status of Y0
b0=READY/BUSY flag; b1=Pulse output status flag; b2=Positioning completed flag;
b3=Positioning abnormal stop flag; b4=Zero home positioning had been completed

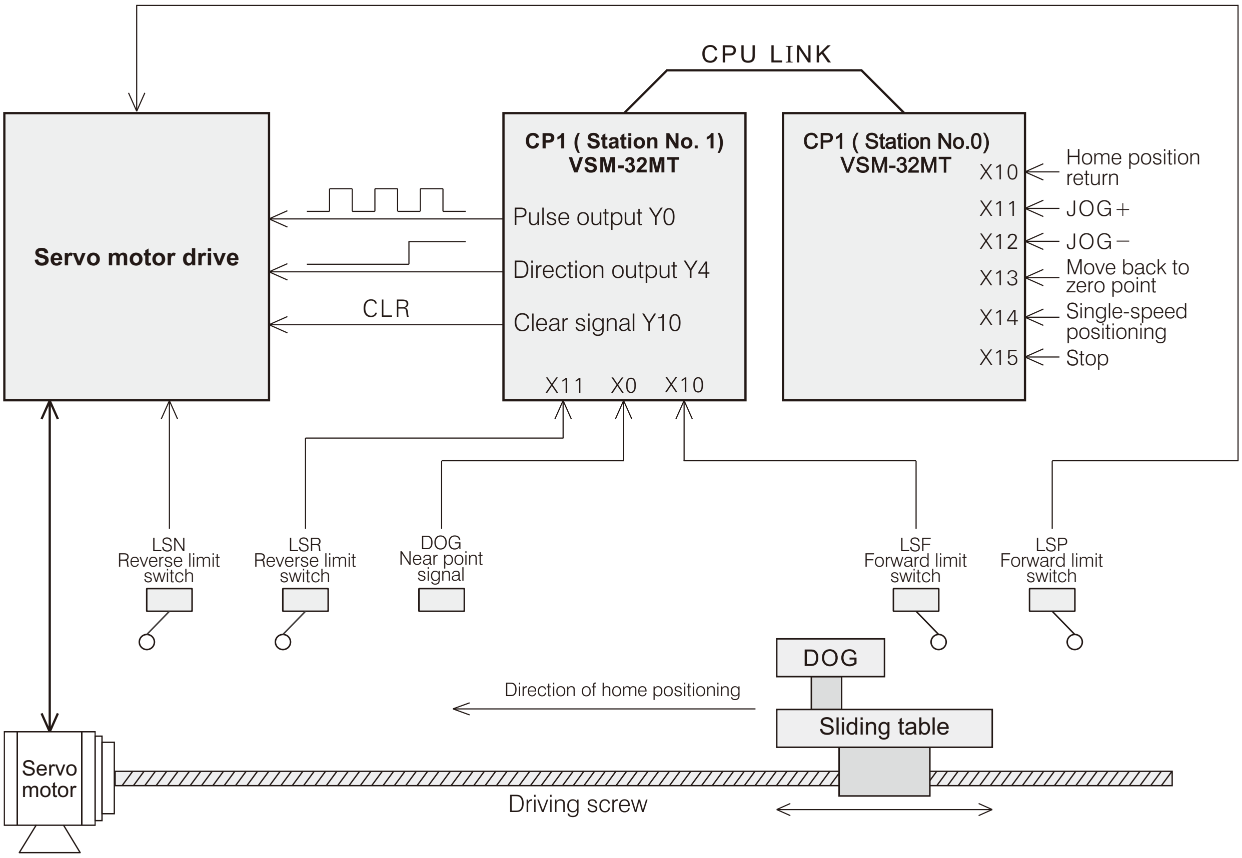
D114 transfers the operation command for the Y0 positioning
b0=STOP; b1=JOGF; b2=JOGR; b3=ZRN; b4=DRVR; b5=DRVA; b6=DV2R; b7=DV2A;
b8=DVIT; b9=DV2I; b10=DVSR; b11=DVSA; b12=PLSV

M9340	D115.0	For to deliver the READY/BUSY flag feedback to the master station
M9341	D115.1	For to deliver the pulse output status flag feedback to the master station
M9342	D115.2	For to deliver the positioning completed flag feedback to the master station
M9343	D115.3	For to deliver the positioning abnormal stop flag feedback to the master station
M9344	D115.4	For to deliver the ZRN completed flag feedback to the master station
M9000	MOV D114 K4M100	Transfer the commands from the master station to individual function contacts
M100	M9346	Force the Y0 axis immediately stop
M101	DJOGF K300 D100 Y0 Y4	Use the JOG speed from the master to perform the JOG+ operation
M102	DJOGR K300 D100 Y0 Y4	Use the JOG speed from the master to perform the JOG− operation
M103	DZRN D102 D104 Y0 Y4	Use the ZRN speed & ZRN creep speed from the master to perform the ZRN operation
M104	DDRVR D106 D110 Y0 Y4	Use the target #1 and speed #1 from the master to perform the relative positioning
M105	DDRVA D106 D110 Y0 Y4	Use the target #1 and speed #1 from the master to perform the absolute positioning
M106	DDV2R D106 D110 Y0 Y4	Use the targets and speeds from the master to perform the 2 stages relative positioning
M107	DDV2A D106 D110 Y0 Y4	Use the targets and speeds from the master to perform the 2 stages absolute positioning
M108	DDVIT D106 D110 Y0 Y4	Use the target #1 and speed #1 from the master to perform the interrupt constant quantity positioning
M109	DDV2I D106 D110 Y0 Y4	Use the target #1 and speeds from the master to perform the 2 stages interrupt constant quantity positioning
M110	DDVSR D106 D110 Y0 Y4	Use the target #1 and speed #1 from the master to perform the interrupt to stop or drive to relative positioning
M111	DDVSA D106 D110 Y0 Y4	Use the target #1 and speed #1 from the master to perform the interrupt to stop or drive to absolute positioning
M112	DPLSV D110 Y0 Y4	Use the speed #1 from the master to perform the variable speed pulse output
M9000	DMOV D9350 D116	For to deliver the current speed feedback to the master station
	DMOV D9354 D118	For to deliver the current location feedback to the master station

Section 3: To edit the related program and setting at the master Station No. 0.
Create the communication table "CPUL0" for the CPU LINK and add the CPUL instruction into the program of the master Station No. 0, thus the station No. 0 and No. 1 will transfer the related data to each other automatically.

```
#5 axis uses D100~119
DD100=JOG spd; DD102=ZRN spd; DD104=ZRN creep spd; DD106=Target#1; DD108=Target#2
DD110=Spd#1; DD112=Spd#2; D114=Command; D115=Status; DD116=Current spd; DD118=Current location

#6 axis uses D120~139
DD120=JOG spd; DD122=ZRN spd; DD124=ZRN creep spd; DD126=Target#1; DD128=Target#2
DD130=Spd#1; DD132=Spd#2; D134=Command; D135=Status; DD136=Current spd; DD138=Current location

#7 axis uses D140~159
DD140=JOG spd; DD142=ZRN spd; DD144=ZRN creep spd; DD146=Target#1; DD148=Target#2
DD150=Spd#1; DD152=Spd#2; D154=Command; D155=Status; DD156=Current spd; DD158=Current location

#8 axis uses D160~179
DD160=JOG spd; DD162=ZRN spd; DD164=ZRN creep spd; DD166=Target#1; DD168=Target#2
DD170=Spd#1; DD172=Spd#2; D174=Command; D175=Status; DD176=Current spd; DD178=Current location

M9000
--| |------------[CPUL CPUL0 D0 K1]
```

Next, edit the control program for the #5 axis. The following example program carries out the home position return, forward JOG, reverse JOG and single-speed positioning functions.
User can try to edit a complex positioning program for study and test, also can use the same shape to add up control program for the #6~#8 axes.
The brief diagram of the #5 axes is shown below.

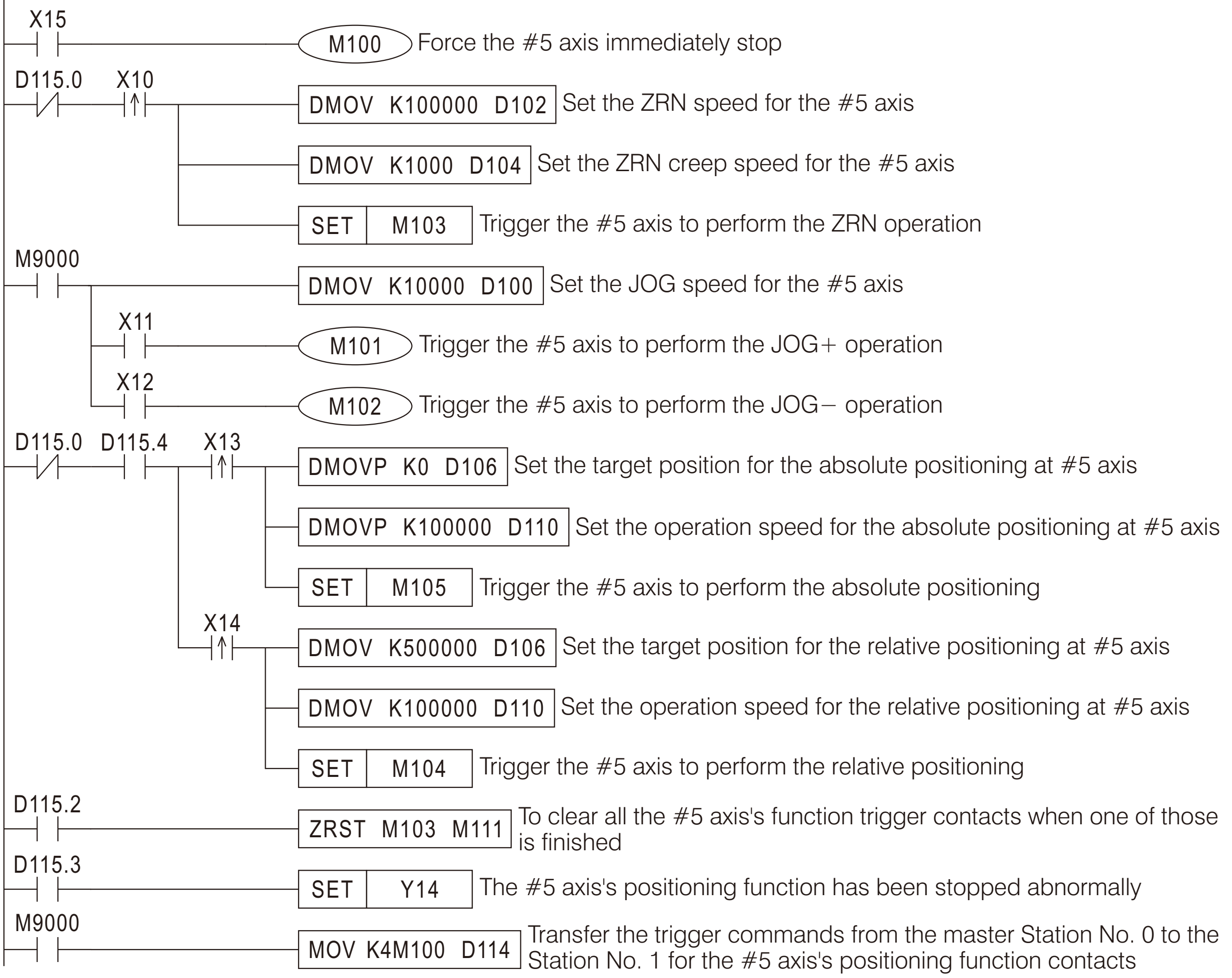


This brief positioning example is in the master Station No. 0 to control the #5 axis at the Station No. 1.

#5 axis uses D100~119
DD100=JOG spd; DD102=ZRN spd; DD104=ZRN creep spd; DD106=Target#1; DD108=Target#2;
DD110=Spd#1; DD112=Spd#2; D114=Command; D115=Status; DD116=Current spd; DD118=Current location

D115 delivers the operation status of #5
b0=READY/BUSY flag; b1=Pulse output status flag; b2=Positioning completed flag;
b3=Positioning abnormal stop flag; b4=Zero home positioning had been completed

D114 transfers the operation command for the #5 positioning
b0=STOP; b1=JOGF; b2=JOGR; b3=ZRN; b4=DRVR; b5=DRVA; b6=DV2R; b7=DV2A;
b8=DVIT; b9=DV2I; b10=DVSR; b11=DVSA; b12=PLSV



MEMO

Index of Application Instructions