Table of Contents

⚠ WARNING

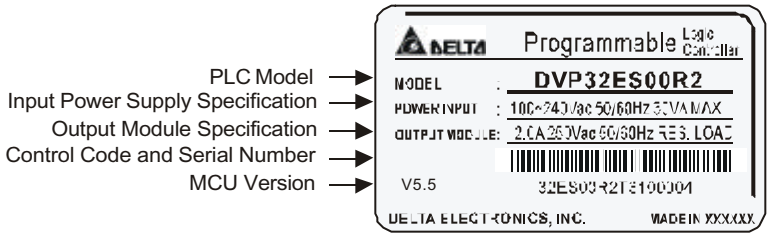
- Always read this manual thoroughly before using DVP PLC.
- AC input power must be disconnected before any maintenance.
- This is an OPEN-TYPE PLC. The PLC must be placed in an enclosure to meet the safety approval of IEC 61131-2 and UL 508.
- The PLC should be kept in an enclosure away from high temperatures, humidity, vibration, corrosive gas, liquid, airborne dust and metallic debrise.
- Do not connect the AC main circuit power supply to any of the input/output terminals, as it will damage the PLC. Ensure all the wiring prior to power up.
- Disconnect all power. Wait one minute for capacitors to discharge before touching internal circuit.
- Some models are equipped with DC power supply output, do not exceed its rated output power.
- Make sure the PLC is properly grounded ⏚, to avoid any electromagnetic noise.

1.1. Model Explanation and Peripherals

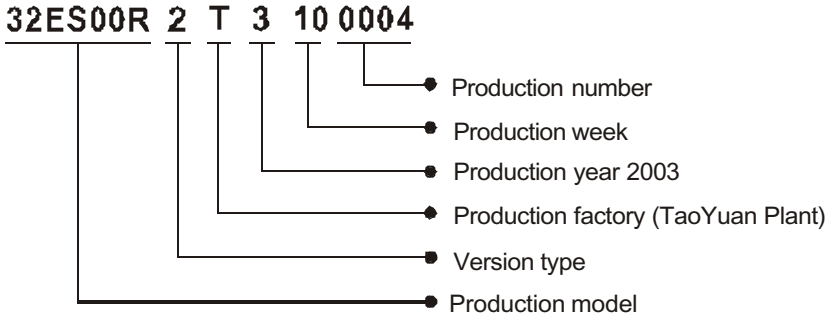
Thank you for choosing DELTA' s PLC DVP Series. The DVP Series has main processing units and extension units. The main processing units offer 14-60 points and the extension units offer 8-32 points. The maximum input/output can be extended up to 128 points. It also can be used on applications according to INPUT/OUTPUT points, power sources, output modules, digital/analog exchanges (A/D & D/A converter). In addition, DVP SS Series has the special modules (AD/DA/PT/TC/XA) used for extending its functions and the maximum special modules can be extended up to 8 units. For more information on the DVP SS Series, refer to the DVP SS Series user manual.

DVP ES/EX/SS MPU is made from improving the functions and specifications of R/T model structure. The additional R2/T2 model has wide improvement in commands type and execution speed. Please refer to the detail information about usable application commands and devices in this manual when using R2/T2 model. The specification in this manual is major for R2/T2 model so that there are some new commands and functions won' t be provided for R/T model.

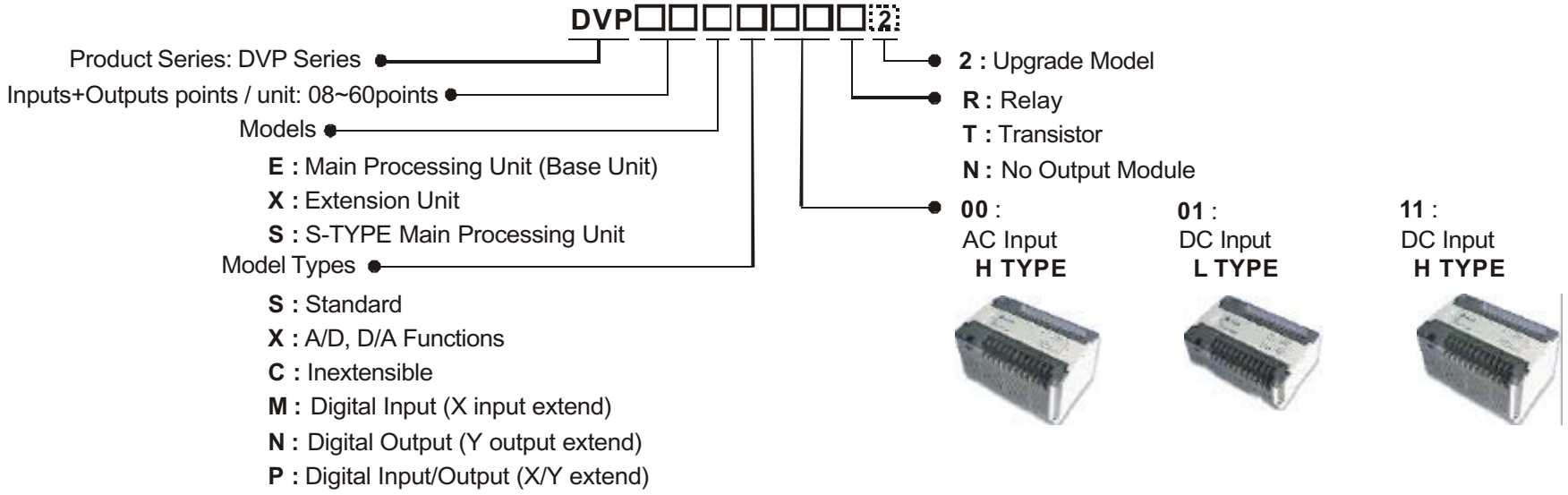
■ Nameplate Explanation



■ Serial Number Explanation



■ Model Explanation



■ Peripheral Equipment

◎ DVPHPP: Handheld Programming Panel
◎ WPLSoft: Windows Ladder Logic Programming Software
◎ DPLSoft: DOS Ladder Logic Programming Software
◎ DVPACAB115: 1.5M Cable (HPP ⇔ PLC, DVPHPP has this cable attached)
◎ DVPACAB215: 1.5M Cable (PC (9 PIN & 25 PIN D-SUB) ⇔ PLC)
◎ DVPACAB230: 3M Cable (PC (9 PIN & 25 PIN D-SUB) ⇔ PLC)
◎ DVPACAB2A30: 3M Cable (PC (9 PIN D-SUB) ⇔ PLC)
◎ DVPACAB230: 3M Cable (PC ⇔ PLC)
◎ DVPACAB315: 1.5M Cable (HPP ⇔ PC)
◎ DVPACAB403: 30cm Cable (MPU-main processing unit⇔Extension Unit or Extension Unit⇔Extension Unit I/O signal extension cable)
◎ DVPAADP01: HPP Power Supply (DVPACAB315 is attached)

1.2. Product Profile and Outline

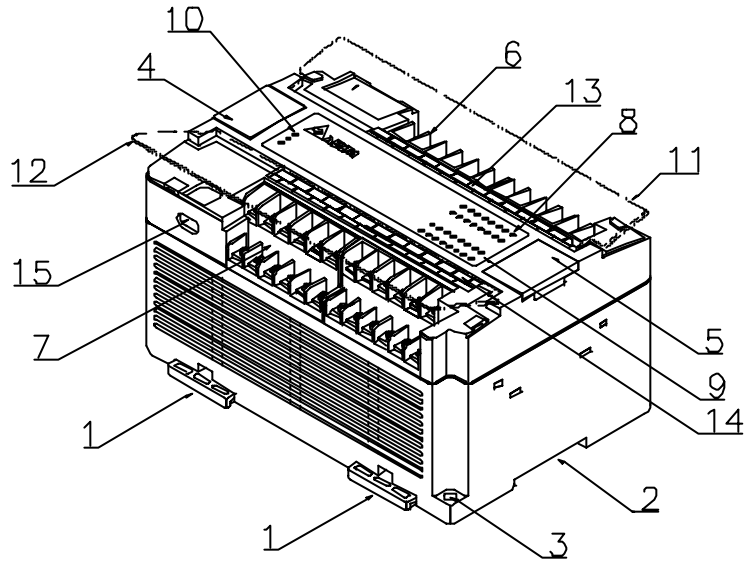


Fig. 1-1: Features of the DVP PLC

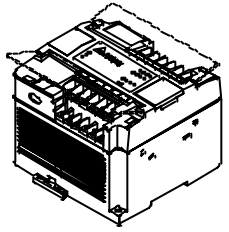
1	DIN rail clip	9	Output indicators
2	DIN rail (35mm)	10	Status indicators, POWER, RUN ERROR
3	Direct mounting holes	11	I/O terminal cover
4	Programming port cover (RS-232)	12	I/O terminal cover
5	Extension port	13	I/O terminal nameplate panel
6	I/O terminals	14	I/O terminal nameplate panel
7	I/O terminals	15	RS-485 Communication port
8	Input indicators		

1.3. Model Numbers

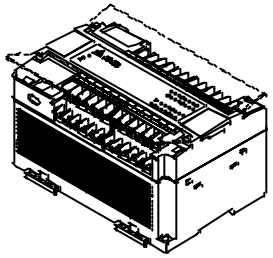
◎ Standard MPU-00

Model	Input / Output					Profile
	Power	Input Unit		Output Unit		
		Point	Type	Point	Type	
DVP14ES00R2	100~240VAC	8	DC Sink or Source	6	Relay	❶
DVP24ES00R2		16		8		❷
DVP32ES00R2		16		16		
DVP60ES00R2		36		24		❺
DVP14ES00T2		8		6	Transistor	❶
DVP24ES00T2		16		8		❷
DVP32ES00T2		16		16		
DVP60ES00T2		36		24		❺

Profile reference



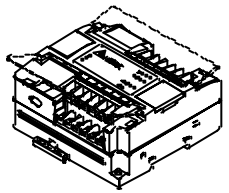
❶



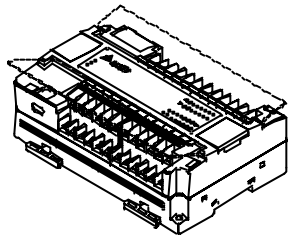
❷

◎ Standard MPU-01

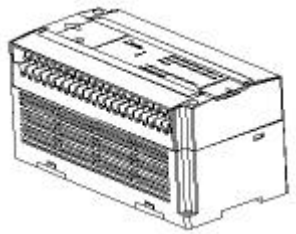
Model	Input / Output					Profile
	Power	Input Unit		Output Unit		
		Point	Type	Point	Type	
DVP14ES01R2	24VDC	8	DC Sink or Source	6	Relay	❸
DVP24ES01R2		16		8		❹
DVP32ES01R2		16		16		
DVP14ES01T2		8		6	Transistor	❸
DVP24ES01T2		16		8		❹
DVP32ES01T2		16		16		



❸



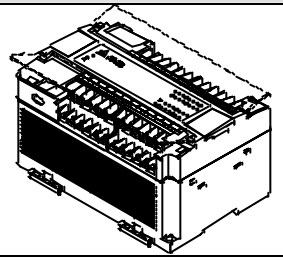
❹



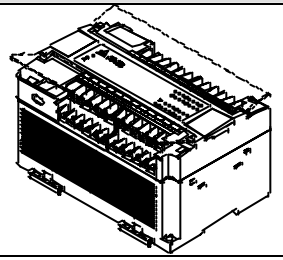
❺

1. Sink or Source connections. Please refer to Chapter 4 Installation and Wiring.
2. Please refer to Chapter 2 Standard Specifications for detailed electrical specifications.

◎ Special Function MPU–00

Model	Input / Output									Profile reference
	Power	Input Unit				Output Unit				
		Point		Type		Point		Type		
		DI	AI	DI	AI	DO	AO	DO	AO	
DVP20EX00R2	100~240VAC	8	4	DC Sink or Source	-20mA~20mA -10V ~ +10 V	6	2	Relay	0~20mA or -10V ~ +10 V (for R/T model) 0V ~ +10 V (for R2/T2 model)	
DVP20EX00T2		8	4			6	2	Transistor		

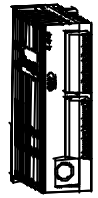
◎ Analog/Digital MPU–11

Model	Input / Output									Profile reference
	Power	Input Unit				Output Unit				
		Point		Type		Point		Type		
		DI	AI	DI	AI	DO	AO	DO	AO	
DVP20EX11R2	24VDC	8	4	DC Sink or Source	-20mA~20mA -10V ~ +10 V	6	2	Relay	0~20mA 0V ~ +10 V	
DVP20EX11T2		8	4			6	2	Transistor		

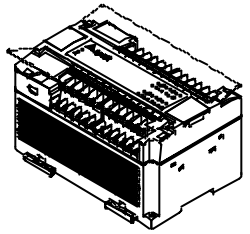
DI（Digital Input） DO（Digital Output）

AI（Analog Input） AO（Analog Output）

◎ Module Standard MPU–11

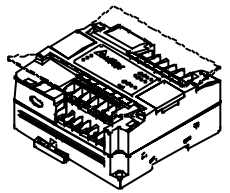
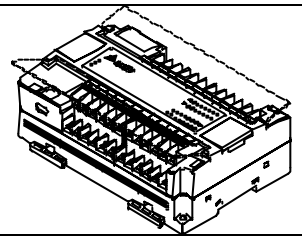
Model	Input / Output					Profile reference
	Power	Input Unit		Output Unit		
		Point	Type	Point	Type	
DVP14SS11R2	100~240VAC	8	DC Sink or Source	6	Relay Transistor	
DVP14SS11T2		8		6		

◎ Digital I/O Extension Unit-00

Model	Input / Output					Profile reference
	Power	Input Unit		Output Unit		
		Point	Type	Point	Type	
DVP24XN00R	100~240VAC	0	DC Sink or Source	24	Relay	
DVP24XP00R		16		8		
DVP32XP00R		16		16		
DVP24XP00T		16		8	Transistor	
DVP24XN00T		0		24		
DVP32XP00T		16		16		

1. Sink or Source connections. Please refer to Chapter 4 Installation and Wiring.
2. Please refer to Chapter 2 Standard Specifications for detailed electrical specifications.

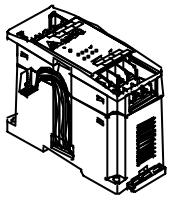
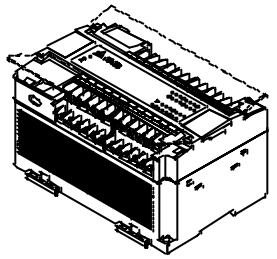
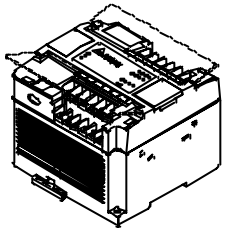
◎ Digital I/O Extension–01 (L-Type)

Model	Power	Input Unit		Output Unit		Profile reference	
		Point	Type	Point	Type		
DVP16XM01N	24VDC	16	DC Sink or Source	0	None	❶	 ❶
DVP16XN01R		0		16	Relay	❷	
DVP24XN01R		0		24			
DVP24XP01R		16		8			
DVP32XP01R		16		16			
DVP16XN01T		0		16	Transistor		 ❷
DVP24XN01T		0		24			
DVP24XP01T		16		8			
DVP32XP01T		16		16			

1. Sink or Source connections. Please refer to Chapter 4 Installation and Wiring.

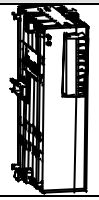
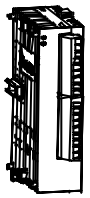
2. Please refer to Chapter 2 Standard Specifications for detailed electrical specifications.

◎ Digital I/O Extension–11 (H-Type)

Model	Power	Input Unit		Output Unit		Profile reference		
		Point	Type	Point	Type			
DVP08XM11N	24VDC	8	DC Sink or Source	0	None	❶	 ❶	 ❸
DVP16XM11N		16		0		❷		
DVP08XN11R		0		8	Relay	❶		
DVP16XN11R		0		16		❸		
DVP24XN11R		0		24				
DVP08XP11R		4		4		❶		
DVP24XP11R		16		8		❸		
DVP32XP11R		16		16				
DVP08XN11T		0		8	Transistor	❶	 ❷	
DVP16XN11T		0		16		❸		
DVP24XN11T		0		24				
DVP08XP11T		4		4		❶		
DVP24XP11T		16		8		❸		
DVP32XP11T		16		16				

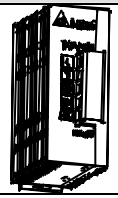
1. Sink or Source connections. Refer to Chapter 4 Installation and Wiring for more information.
2. Refer to Chapter 2 Standard Specifications for detailed electrical specifications.
3. When connects to DVP08XP11R/T, recommend to place at the end of extension units. Refer to Chapter 8 EX MPU and I/O Extension Units for more information.

◎ Digital I/O Extension Units

Model	Power	Input Unit		Output Unit		Profile reference
		Point	Type	Point	Type	
DVP08SM11N	24VDC	8	DC Sink or Source	0	None	
DVP08SN11R		0		8	Relay	
DVP08SN11T		0		8	Transistor	
DVP08SP11R		4		4	Relay	
DVP16SP11R		8		8		
DVP08SP11T		4		4	Transistor	
DVP16SP11T		8		8		

◎ Please refer to each user manual of extension model for the detail of SS special extension module.

◎ Power Output Module

Model	Input Power	Output Power	Profile reference
DVPPS01	100~240VAC （50/60Hz）	Output Voltage: 24VDC Max. Output Current: 1A	

■ General Specifications

Items		Specifications	Remarks
Control Method		Stored program, cyclic scan system	
I/O Processing Method		Batch I/O (refresh)	Direct I/O instruction available
Execution Speed		Basic commands (several μs)	Application commands (10~hundreds μs)
Program Language		Ladder Logic, Instruction, SFC	Including the Step commands
Program Capacity		3792 STEPS	Built-in EEPROM
Instructions		32 Basic sequential commands (Including STL / RET)	97 Application commands (145 including the 32-bit commands)
Auxiliary Relays	General	512+232 Points	M000~M511+ M768~M999
	Latched	256 Points	M512~M767
	Special	280 Points	M1000~M1279
Step Relays (Latched)	Initial Step Point	10 Points	S0~S9
	General Step Point	118 Points	S10~S127
Timers	Digital	64 Points	T0~T63 (100ms time base)
		63 Points	T64~T126 (10ms time base, when M1028 is ON)
		1 Points	T127 (1ms time base)
Counters	General	112 Points	C0~C111 (16-bit counters)
	Latched	16 Points	C112~C127 (16-bit counters)
	High-speed	13 Points 1-Phase 5KHz, 2-Phase 5KHz	C235~C254 (all latched type, 32-bit counters)
Data registers	General	408 Points	D0 ~ D407
	Latched	192 Points	D408~D599
	Special	312 Points	D1000~D1311
Digital/Analog	A→ D	4 Analog Input Channels	10-bit resolution (EX MPU only)
	D→ A	2 Analog Output Channels	8-bit resolution (EX MPU only)
High Speed Pulse Output		2 point (Y0, Y1), pulse output frequency: 10Hz to 10KHz	
Pointers/Interrupt	P / I	P : 64 Points / I : 4 Points	P0~P63 / I001, I101, I201, I301
Index Register	E / F	2 Points	E (=D1028), F (=D1029)
Constants	Decimal K	16-bit: -32768~+32767	32-bit: -2147483648~+2147483647
	Hexadecimal H	16-bit: 0000~FFFF	32-bit: 00000000~FFFFFFFF
Serial Communication		RS-232, RS-485 (2 Ports)	
Protection Features		Password, Execution Time, Illegitimate Command or Operand	
Monitor / Debug		Execution time, Device setting	

■ Electric Specifications 1

Item \ Model	DVP-14ES00□	DVP-24ES00□	DVP-32ES00□	DVP-60ES00□	DVP-20EX00□	DVP-14ES01□	DVP-24ES01□	DVP-32ES01□	DVP-20EX11□
Power Supply Voltage / Fuse	100~240VAC (-15%~10%), 50/60Hz ± 5% / 2 A / 250VAC					24VDC (-15%~10%) / 2 A / 250VAC			
Input Power Operating Characteristics	95-100VAC is needed to start the PLC. If the voltage drops to 70VAC or less, the PLC will stop.					A minimum of 18VDC is needed to start the PLC. If the voltage drops below 17.5VDC, the PLC will stop.			
Maximum Power Loss Time	10ms or less					5ms or less			
Power Consumption	20 VA	25VA	30VA	35VA	30 VA	5.5 W	6.5 W	8 W	10 W
DC24V Supply Current	400mA	400mA	400mA	200mA	400mA	—	—	—	—
Power Protection	DC24V output short circuit					DC24V input polarity			
Withstand Voltage	1500VAC(Primary-secondary), 1500VAC(Primary-PE), 500VAC(Secondary-PE)								
Insulation Resistance	>5 MΩ at 500VDC (Between all inputs/outputs and earth)								
Noise Immunity	ESD: 8KV Air Discharge EFT: Power Line: 2KV, Digital I/O: 1KV, Analog & Communication I/O: 250V Damped-Oscillatory Wave: Power Line: 1KV, Digital I/O: 1KV RS: 26MHz~1GHz, 10V/m								
Grounding	The diameter of grounding wire cannot be smaller than the wire diameter of terminals L and N (All DVP units should be grounded directly to the ground pole).								
Environment	Operation: 0℃~55℃ (Temperature), 50~95% (Humidity); Storage: -25℃~70℃ (Temperature), 5~95% (Humidity) Pollution degree 2								
Vibration /Shock Immunity	Standard: IEC1131-2, IEC68-2-6 (Test Fc) / IEC 1131-2 & IEC68-2-27 (Test Ea)								
Weight (g)	400	552	580	750	536	260	414	430	386

Input Point Electric Specification		
Input Point Type	Digit I/O	Analog I/O (EX)
Input Point Spec.	DC (Sink or Source) 24VDC 5mA	Input Voltage: -10V~+10V (Input Resistance: 40KΩ) Input Current: -20mA~+20mA(Input Resistance: 250Ω)
Active Level (Analog Input Resolution)	OFF→ON Above 10VDC ON→OFF Below 9VDC	Input Voltage Resolution: 10bit Input Current Resolution: 10bit
Reaction Time	About 10ms (Adjust D1020 and D1021 can be 0~15ms)	5ms (Adjusting D1118 can change reaction time)

Output Point Electric Specification			
Output Point Type	Relay-R	Transistor-T	Analog I/O (EX)
Current Spec.	2A/1 point (5A/COM)	0.3A/1point (1.2A/COM)	0~20mA
Voltage Spec.	Below 250VAC, 30VDC	30VDC	0V~+10V
Maximum Load	100 VA (Inductive) 120 W (Resistive)	9W	0.2W
Reaction Time	About 10ms	OFF→ON 20 μs ON→OFF 30 μs	10ms
Resolution	—	—	8 bit

■ Electric Specifications 2

Item \ Model	DVPPS01	DVP14SS11R2/T2	DVP08SM11N	DVP08SN11R/T	DVP08SP11R/T	DVP-16SP11R/T
Power Supply Voltage / Fuse	100~240VAC (50/60Hz)	24VDC (-15%~10%) (has power protect with opposite pole DC input)				
Input Power Operating Characteristics	5ms or less					
Maximum Power Loss Time	--	2A/ 250VAC				
Power Consumption	--	5 W			8 W	
Insulation Resistance	--	>5 MΩ at 500VDC (Between all inputs/outputs and earth)				
Noise Immunity	ESD: 8KV Air Discharge EFT: Power Line: 2KV, Digital I/O: 1KV, Analog & Communication I/O: 250V Damped-Oscillatory Wave: Power Line: 1KV, Digital I/O: 1KV RS: 26MHz~1GHz, 10V/m					
Grounding	The diameter of grounding wire cannot be smaller than the wire diameter of terminals L and N (All DVP units should be grounded directly to the ground pole).					
Environment	Operation: 0℃~55℃ (Temperature), 50~95% (Humidity); Storage: -25℃~70℃ (Temperature), 5~95% (Humidity) Pollution degree 2					
Vibration /Shock Immunity	Standard: IEC1131-2, IEC68-2-6 (Test Fc) / IEC 1131-2 & IEC68-2-27 (Test Ea)					
Weight (g)	210	214/208	128	154 /146	141 /136	162 /154

Input Point Electric Specification	
Input Point Type	DC (Sink or Source)
Input Current	24VDC 7mA
Active Level	OFF→ON Above 10VDC
	ON→OFF Below 9VDC
Reaction Time	About 10ms (Adjust D1020 and D1021 can be 0~15ms)

Output Point Electric Specification		
Output Point Type	Relay-R	Transistor-T
Current Spec.	2A/1 point (5A/COM)	55℃ 0.1A/1 point, 50℃ 0.15A/1 point 45℃ 0.2A/1 point, 40℃ 0.3A/1 point (2A/COM)
Voltage Spec.	Below 250VAC, 30VDC	30VDC
Maximum Load	100 VA (Inductive)	7.2 W
	120 W (Resistive)	
Reaction Time	About 10ms	OFF→ON 15 μs, ON→OFF 25 μs

3.1. Special Auxiliary Relays

◎ PLC Operation Status	
M1000	Normally ON contact (a contact)
M1001	Normally OFF contact (b contact)
M1002	ON only for 1 scan after RUN
M1003	OFF only for 1 scan after RUN
M1004	On when error occurs
M1008	Monitor timer flag (ON: PLC WDT time out)
M1009	24VDC down detection
M1010	PLSY Y0 mode selection. ON: output continuously
◎ Clocks	
M1011	10msec clock
M1012	100msec clock
M1013	1sec clock
M1014	1min clock
◎ Flags	
M1019	Cancel X0~X17 input delay
M1020	Zero flag
M1021	Borrow flag
M1022	Carry flag
M1023	PLSY Y1 mode selection. ON: output continuously
M1024	System used
M1025	Invalid communication request
M1028	10ms time base setting flag
M1029	PLSY Y0 Instruction execution completed flag
M1030	PLSY Y1 Instruction execution completed flag
M1070	The Pulse unit switching (ON: 100 μs) will be conducted at the PWM command.
M1031	Non-holding memory all clear
M1032	Holding memory all clear
M1033	Memory holding at STOP
M1034	All outputs disable
M1039	Constant scan mode

◎ Step Ladder Diagram	
M1040	Step transition inhibit
M1041	Step transition start
M1042	Start pulse
M1043	Origin reset completed
M1044	Origin condition
M1045	All outputs clear inhibit
M1046	STL state setting
M1047	STL monitor enable
◎ Interrupt Inhibit Setting	
M1050	I 001 masked
M1051	I 101 masked
M1052	I 201 masked
M1053	I 301 masked
◎ Error Flags	
M1060	CPU hardware error
M1061	CPU internal malfunction (Flag)
M1062	CPU internal malfunction (BIOS)
M1063	CPU internal malfunction (RAM)
M1064	Operator error
M1065	Syntax error
M1066	Program error
M1067	Program execution error
M1068	Execution error latch (ref. D1068)
◎ PLC Operation Execution	
M1072	PLC RUN command execution
M1073	Grammar inspection flag
M1078	Y0 pulse output/stop control flag
M1079	Y1 pulse output/stop control flag
M1083	From / To mode exchange
◎ RS-485 Communication	
M1120	Communication protocol holding
M1121	Transmission ready

M1122	Sending request
M1123	Receiving completed
M1124	Receiving wait
M1125	Communication reset
M1126	STX/ETX selection
M1127	MODRD, RDST commands. Data receiving completed
M1128	Transmitting / Receiving indication
M1129	Receiving time out
M1130	STX/ETX selection
M1131	M1131=On during the conversion (MODRD/RDST/MODRW data->HEX).
M1140	MODRD / MODWR data received error
M1141	MODRD / MODWR command error
M1142	VFD-A command data received error
M1143	ASCII / RTU mode selections
M1161	8/16-bit mode setting
◎ High Speed Counter (1-phase input)	
M1235	C235 counting mode (on: count down)
M1236	C236 counting mode (on: count down)
M1237	C237 counting mode (on: count down)
M1238	C238 counting mode (on: count down)
M1241	C241 counting mode (on: count down)
M1242	C242 counting mode (on: count down)
M1244	C244 counting mode (on: count down)
◎ High Speed Counter (1-phase 2 inputs)	
M1246	C246 monitor (on: count down)
M1247	C247 monitor (on: count down)
M1249	C249 monitor (on: count down)
◎ High Speed Counter (2-phase inputs)	
M1251	C251 monitor (on: count down)
M1252	C252 monitor (on: count down)
M1254	C254 monitor (on: count down)

3.2. Special Data Registers

◎ PLC System Information	
D1000	Watchdog timer (WDT) value
D1001	DVP model no. + memory cap. /type
D1002	Program memory capacitor
D1003	Sum of program memory
D1004	Error flag number
D1005	System message
D1008	Monitor the STEP position that occurs when timer time out
D1010	Current scan time (unit: 0.1ms)
D1011	Minimum scan time (unit: 0.1ms)
D1012	Maximum scan time (unit: 0.1ms)
D1020	X00~X07 input delay setting (0~15ms)
D1021	X10~X17 input delay setting (0~15ms)
D1022	AB phase counter mode selections
D1025	Communication error code
D1028	Index register E
D1029	Index register F
D1030	Output numbers of Y0 pulse (Low word)
D1031	Output numbers of Y0 pulse (High word)
D1032	Output numbers of Y1 pulse (Low word)
D1033	Output numbers of Y1 pulse (High word)
D1038	When PLC MPU is master, the setting of data response delay time. Time unit is 0.1ms.
D1039	Constant scan time (unit: ms)
◎ Step Ladder Diagram	
D1040	ON state number 1
D1041	ON state number 2
D1042	ON state number 3
D1043	ON state number 4
D1044	ON state number 5
D1045	ON state number 6
D1046	ON state number 7
D1047	ON state number 8

◎ Error Check	
D1061	System detailed error code
D1065	Syntax error code
D1066	Loop error code
D1067	Algorithm error code
D1068	Lock the algorithm error address
D1069	Step number of errors associated with flags M1065~M1067
◎ System Usage	
D1050 ↓ D1055	PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX. Refer to chapter 7 Application Commands for more information.
D1070 ↓ D1085	When the PLC built-in RS-485 communication command receives feedback signals from receiver, the signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data. Refer to chapter 7 for more details.
D1089 ↓ D1099	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in the registers to check the feedback data. Refer to chapter 7 for more details.
D1256 ↓ D1295	ES: MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256-D1295. User can check according to the content of these registers. (Using MOV, DMOV, BMOV to move the data in this area in version 4.9.)
D1296 ↓ D1311	ES: PLC system will convert ASCII in the content of the register that user indicates to HEX. (Using MOV, DMOV, BMOV to move the data in this area in version 4.9.)

◎ A/D, D/AConversion (Only EX Model)	
D1056	Present value of analog input channel 0 (CH0)
D1057	Present value of analog input channel 1 (CH1)
D1058	Present value of analog input channel 2 (CH2)
D1059	Present value of analog input channel 3 (CH3)
D1110	Average of analog input channel 0 (CH 0)
D1111	Average of analog input channel 0 (CH 1)
D1112	Average of analog input channel 0 (CH 2)
D1113	Average of analog input channel 0 (CH 3)
D1116	Analog output channel 0 (CH 0)
D1117	Analog output channel 1 (CH 1)
D1118	For EX model only. It is the filter wave time setting between the A/D conversions, and with the default setting as 0 and the unit as 1ms, all will be regarded as 5ms if D1118≦5
◎ PLC System Setting	
D1119	System used (PLC operation mode)
D1121	PLC communication address
◎ RS-485 Serial Communication Port	
D1120	RS-485 communication protocol
D1122	Residual words of transmitting data
D1123	Residual words of receiving data
D1124	Start character definition
D1125	First ending character definition (ETX1)
D1126	Second ending character definition (EXT2)
D1129	RS-485 time-out setting (ms)
D1130	MODBUS return error code record
◎ Auxiliary System Check Information	
D1136	System used (Error diagnosis)
D1137	Address of operator error occurs
D1140	Special extension module number
D1141	System used (Self-diagnosis code)
D1142	Input points (X) of extension unit
D1143	Output points (Y) of extension unit

Additional Information

M: Read Only Relay, can work as a contact yet cannot work as an output coil. But M1131and M1132 are used for system, they can't work as a contact or a output coil.
D: Read Only Register.

Here are the descriptions of the special devices, also refer to chapters 6 and 7 for more details.

Device	Descriptions	Device	Descriptions	Device	Descriptions
M1025	If the PLC receive an illegal communication service request when it is connected with an HPP, PC or HMI (Human-Machine-Interface), the M1025 will be set and save the error code in D1025.	D1001	Users have access to where the software version is saved in Device D1001, e.g. D1001 = H 40**27** is an indication of Version 2.7. HPP is utilized to read the data. When "Knnnnn" is displayed, simply press the <H> key to switch to the HEX display mode.	D1121	Saves the PLC communication addresses. This is a Latched Register.
M1028	OFF: the time base of timer T64~T126 is 100ms. ON: the time base is 10ms.	D1003	Sums up the memory content of the PLC. Users could use this data register to identify the internal program of the PLC.	D1119	Identify the PLC operating mode: 1: A/D operating mode 2: D/A operating mode 3: A/D, D/A operating mode 4: normal mode (DI/DO)
M1143	Used with the MODRD/MODWR commands: OFF: the ASCII mode ON: the RTU mode	D1025	Error Code: 01: illegal command 02: illegal device command 03: requested data exceeds the range 07: checksum error		

3.3. High Speed Counters

Type / Input	1-phase input							1-phase 2 inputs			2-phase inputs		
	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X00	U/D				U/D		U/D	U	U	U	A	A	A
X01		U/D			R		R	D	D	D	B	B	B
X02			U/D			U/D			R	R		R	R
X03				U/D		R	S			S			S

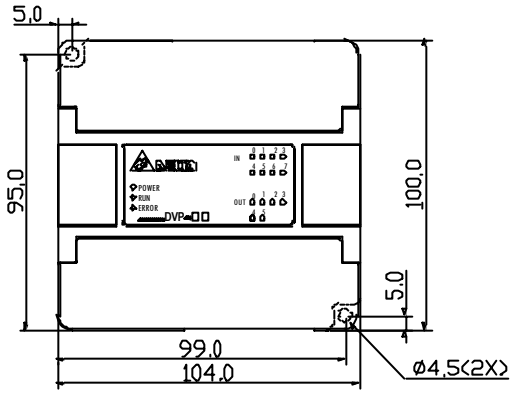
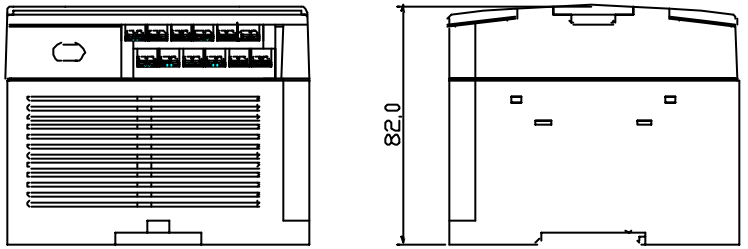
U: Increasing
D: Decreasing
A: A-phase input
B: B-phase input
S: Start input
R: Clear input

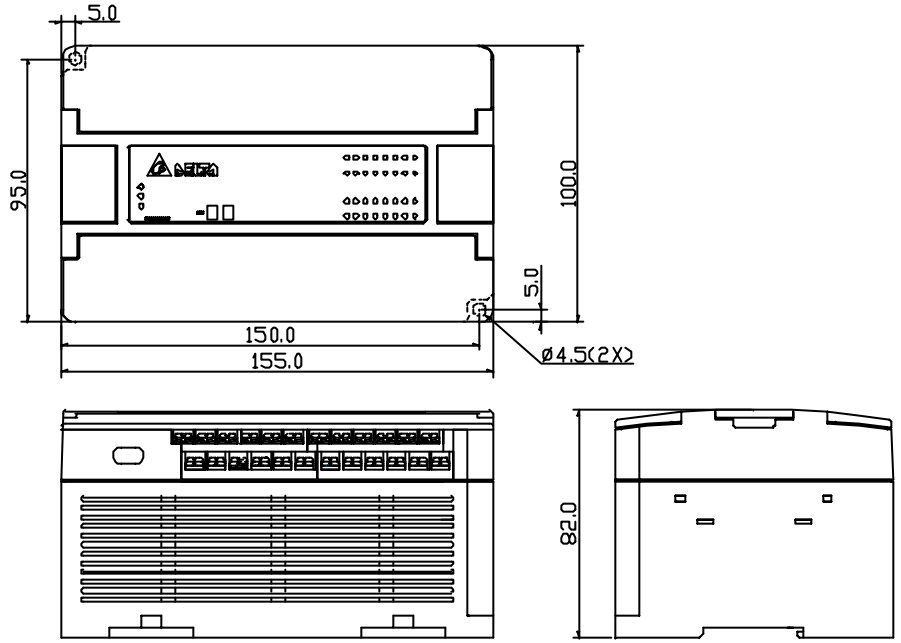
☞ When high speed counters C235~C254 are in use, the inputs of the counter will be disabled for other use.

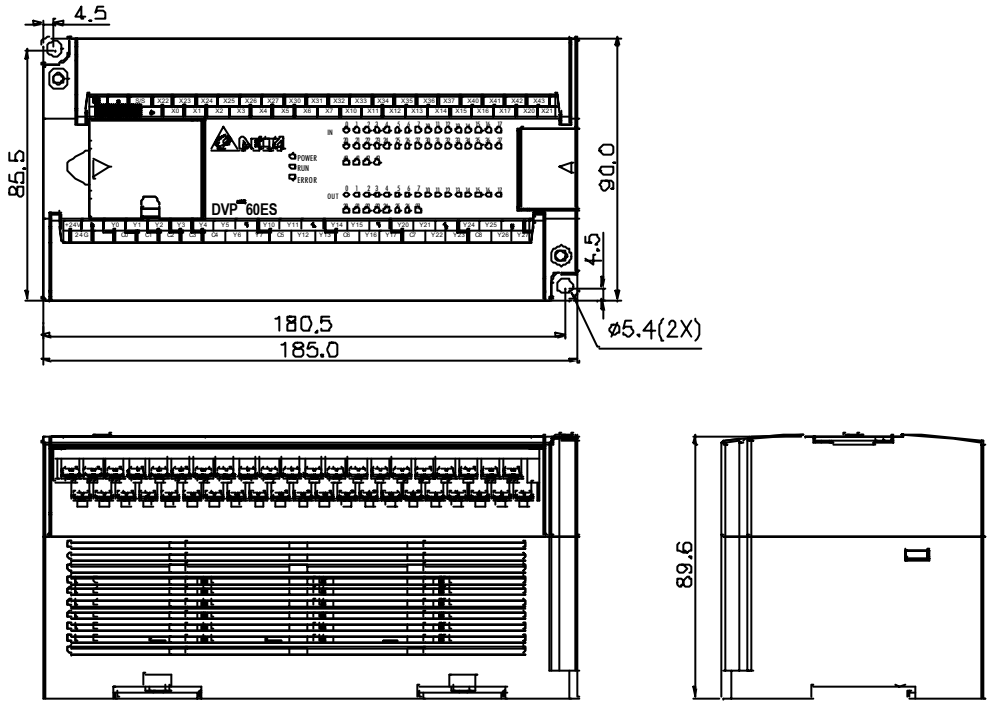
☞ Refer to chapter 7 Application Instructions for more details on API53 DHSCS and API54 DHSCR.

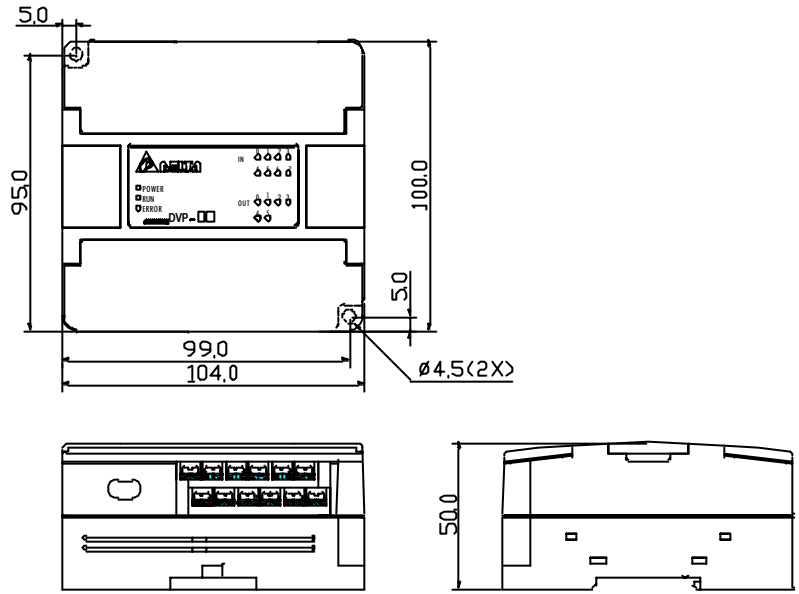
4.1. Dimension and Terminals

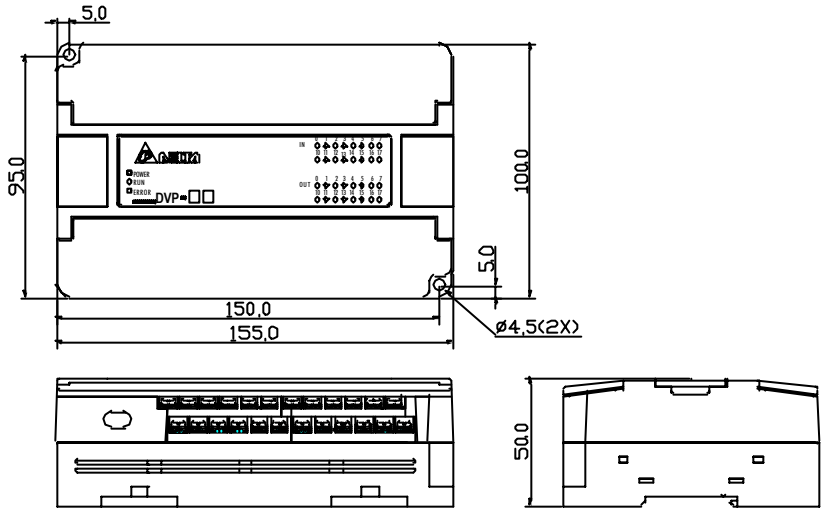
■ Dimensions

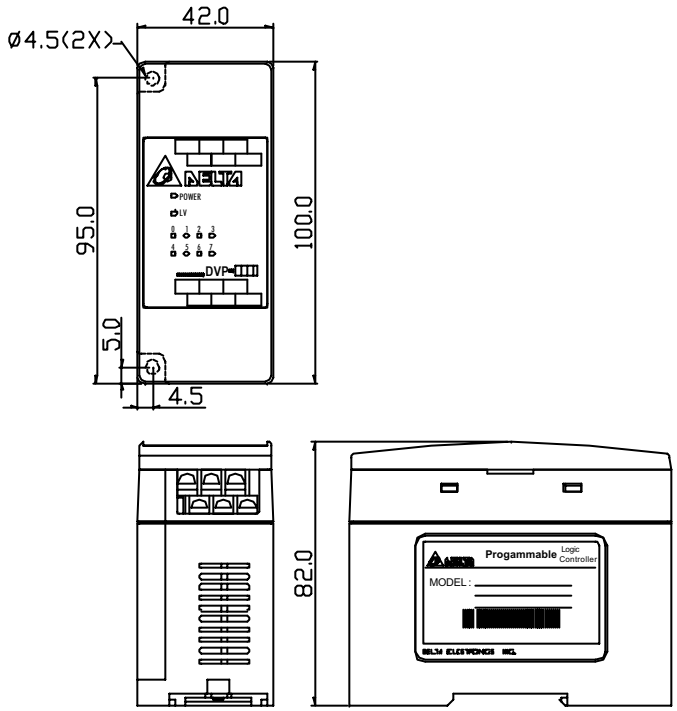
Model		Dimensions
MPU (Including power supply)	DVP14ES00 R2/T2	
Digital I/O Extension Unit (No power supply)	DVP16XM11N	

Model		Dimensions
MPU (Including power supply)	DVP24ES00 R2/T2 DVP32ES00 R2/T2 DVP20EX00 R2/T2	
MPU (No power supply)	DVP20EX11 R2/T2	
Digital I/O Extension Unit (Including power supply)	DVP24XP00 R/T DVP24XN00 R/T DVP32XP00 R/T	
Digital I/O Extension Unit (No power supply)	DVP16XN11 R/T DVP24XN11 R/T DVP24XP11 R/T DVP32XP11 R/T	

Model		Dimensions
MPU (Including power supply)	DVP60ES00 R2/T2	

Model		Dimensions
MPU (No power supply)	DVP14ES01 R2/T2	
Digital I/O Extension Unit (No power supply)	DVP16XM01N	

Model		Dimensions
MPU (No power supply)	DVP24ES01 R2/T2 DVP32ES01 R2/T2	
Digital I/O Extension Unit (No power supply)	DVP16XN01 R/T DVP24XP01 R/T DVP24XN01 R/T DVP32XP01 R/T	

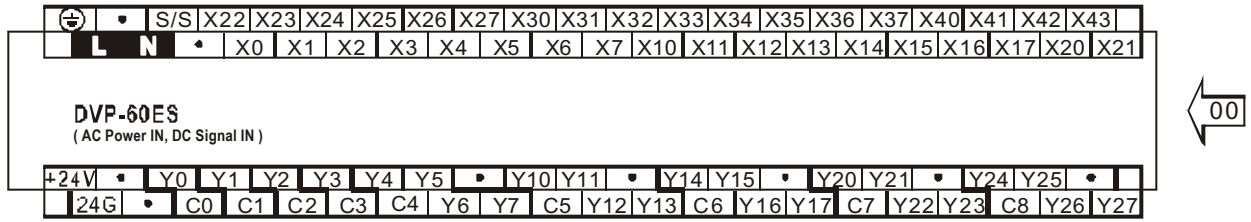
Model		Dimensions
Digital I/O Extension Unit (No power supply)	DVP08XM11N DVP08XN11 R/T DVP08XP11 R/T	

4.2. Terminal Wiring

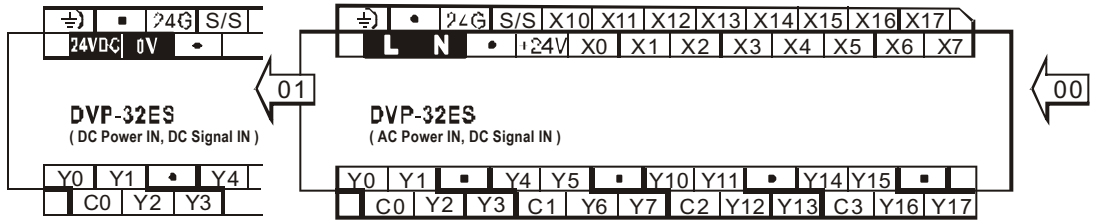
■ Terminal Layouts of the Standard Function MPU

What follows is a complete display of the terminal wiring for all the model types within the DVP Series; refer to locations 13 and 14 on Figure 1-1 of 1.2 Product Profile and Outline for detail.

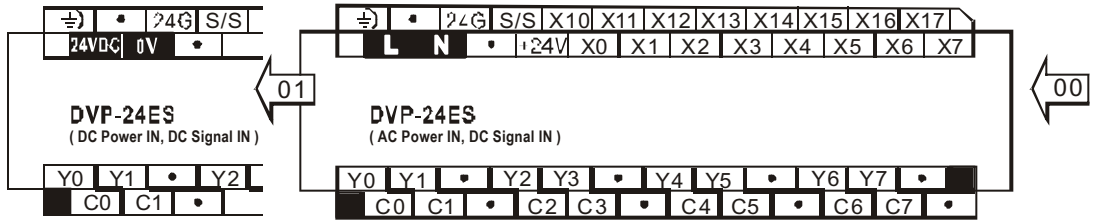
DVP60ES

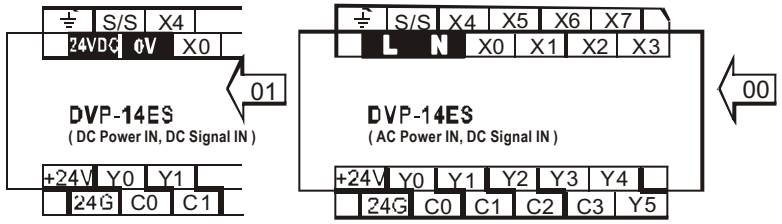


DVP32ES

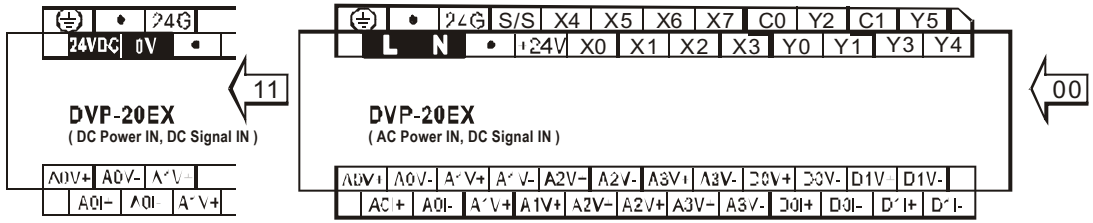


DVP24ES

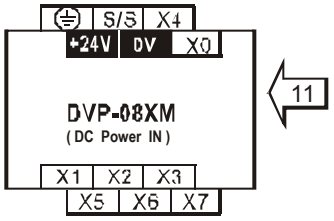
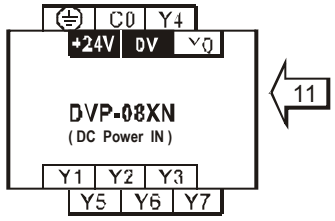
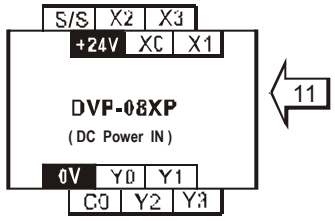


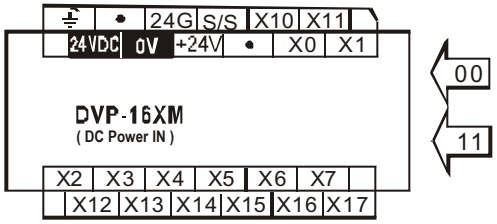
DVP14ES	

■ Terminals Layouts of Special Function MPU

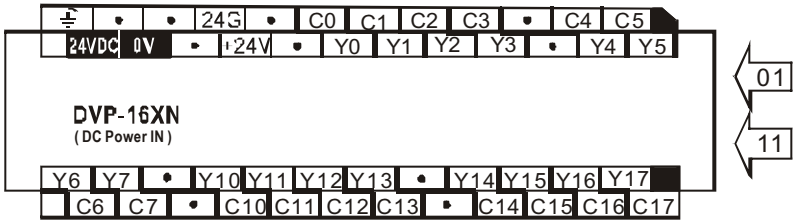
DVP20EX	

■ Terminals Layouts of Digital I/O Extension Unit

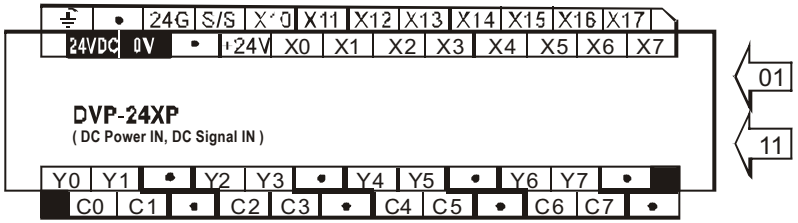
DVP08XM		DVP08XN		DVP08XP	

DVP16XM	

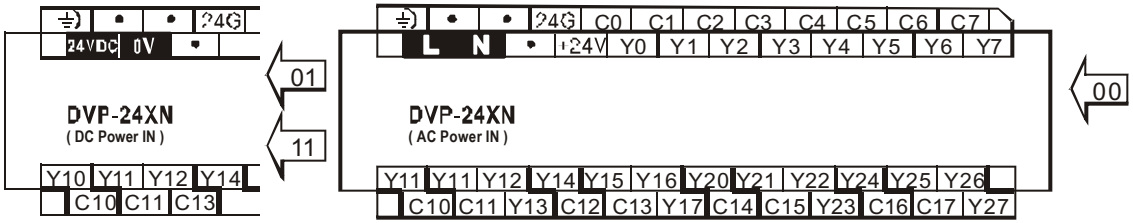
DVP16XN



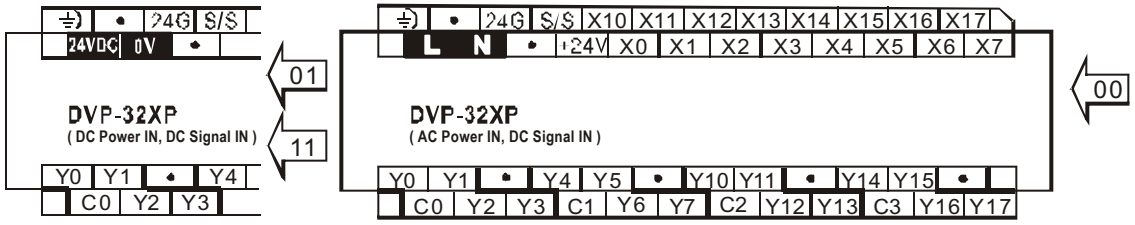
DVP24XP



DVP24XN



DVP32XP



- For the information of terminals layouts of DVP SS Series, refer to the DVP SS Series user manual.

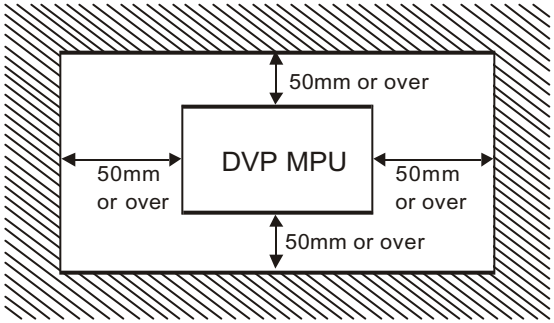
4.3. PLC Mounting Arrangements and Wiring Notes

The installation of the DVP products has been designed to be safe and easy. Whether the products associated with this manual are used as a system or individually, they must be installed in a suitable enclosure. The enclosure should be selected and installed in accordance to the local and national standards.

■ PLC mounting arrangements

PLC should be mounted on a vertical position. To prevent a rise in temperature, units should always be mounted on the back wall of an enclosure. Never mount PLC to the floor or ceiling of the enclosure.
Caution:

1. Do not install units in areas with excessive or conductive dust, corrosive or flammable gas, moisture or rain, excessive heat, regular impact shocks or excessive vibration.
2. Do not allow debris to fall inside the unit during installation, e.g. cut wires, shavings etc. After installation, remove the protective paper band to prevent overheating.
3. Always ensure that units are kept as far as possible from high-voltage cables and equipment.



◎ DIN Rail Installation

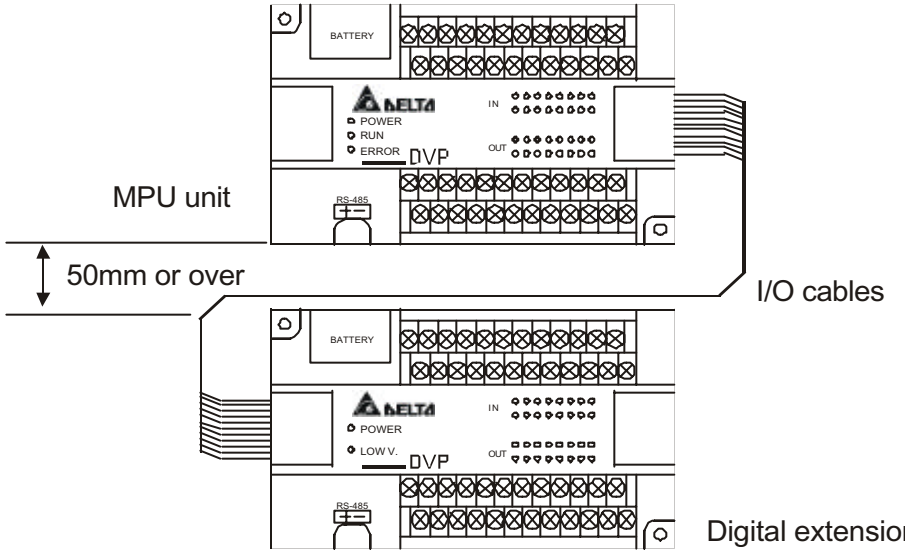
The DVP-PLC can be secured to a cabinet by using DIN rail. The DIN rail should be 35mm high, with a depth of 7.5mm. When mounting the PLC on a DIN rail, please use end brackets to stop any side-to-side motion of the PLC. This will reduce the chance of any wires being pulled loose.

On the bottom of the PLC is a small retaining clip. To secure the PLC to a DIN rail, place it onto the rail and gently push up on the clip. To remove the PLC, pull down on the retaining clip and gently pull the PLC away from the DIN rail.

◎ Direct mounting

Using the specified dimensions and installing the DVP PLC directly on a vertical flat by M4 screws. Make sure you follow the installation guidelines to allow proper spacing from other components.

◎ Parallel connection



1. I/O cables of digital I/O extension unit are easier to be interfered, therefore please keep the I/O cables away from the output cables and power cables at least a 50mm or more distance.

2. The digit I/O extension unit can be connected in parallel, therefore please make sure the I/O cables be firmly connected to the left extension ports of the digit I/O extension unit when connecting one digit I/O extension unit to the other. As for the right extension ports of the digit I/O extension unit, they are used for the next extension.

3. The attached standard cable with the digit I/O extension unit is 80mm. If user desires to work as the left figure shown, please order the specified cable (DVPACAB403, 30cm Cable).

■ Wiring Notes

The following guidelines provide general information on how to wire the I/O connections to DVP PLCs.

☑ Environment

1. **DO NOT** store the PLC in a dusty, smoky, or corrosive atmosphere.
2. **DO NOT** store the PLC in an environment with high temperature or high humidity.
3. **DO NOT** install PLC on a shelf or on an unstable surface.

☑ Construction

1. Some machine fabrication environments may accidentally cause conductive debris to fall through the DVP cooling vents and into the unit. ALL DVP units come with a protective sheet wrapped round the unit, covering the cooling vents. However, it must be removed before electrical operation.
2. There should be a 50mm or more distance between the PLC and other control components. Also, keep the PLC away from high voltage lines & power equipment.

☑ Avoid creating sharp bends in the wires.

☑ Avoid running DC wiring in close proximity to AC wiring.

☑ To minimize voltage drops on long wire runs, consider using multiple wires for the return line.

☑ Avoid running input wiring close to output wiring where possible.

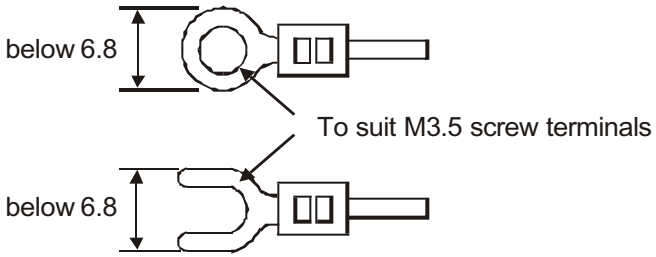
☑ Avoid running wires near high power lines.

☑ Use wire trays for routing where possible.

☑ Use the shortest possible wire length.

☑ Always use a continuous length of wire. Do not splice wires to attain a needed length.

☑ Recommended wire terminations.



1. Cables terminating at a screw terminal of a DVP product should be fitted with insulated crimp terminals, see examples shown at left. Terminal screws should be tightened to between 5 and 8 kg-cm（4.3 and 6.9 in-lbs）. Screw terminals must be secure enough to prevent a loose connection from causing a malfunction.
2. **DO NOT** wire to the No function terminals.
3. Input and output signal wires should not run through the same multi-wire cable, conduit, or near high voltage cables.
4. All low voltage wires should cross high voltage cables at 90° when possible.
5. Use Copper Conductors only
6. 75℃ only

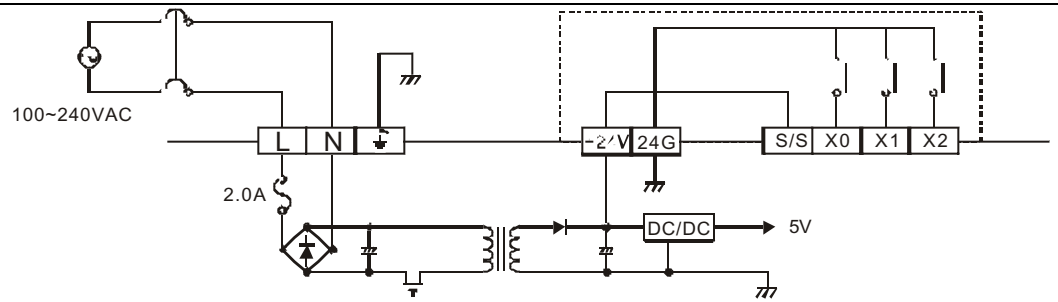
■ Recommended Grounding

For grounding, use at least 2mm^2（AWG14）cable. Ground resistance must be less than 100ohm（Class 3）. The PLC's ground should not be shared with that of the power circuits. While grounding is recommended, if it is not possible, the PLC will still operate correctly without it. Ground terminal : All ground terminals should be linked with 2mm^2（AWG14）cable. The linked terminals should all be connected to a single earth point.

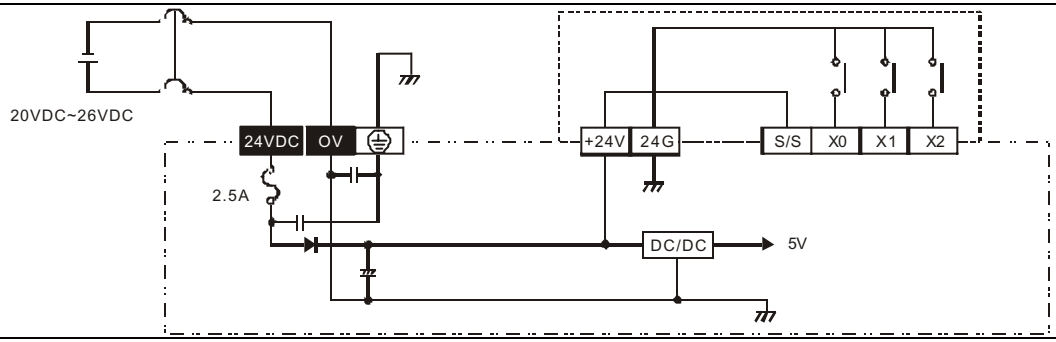
4.4. Wiring Guidelines

■ Power Input Wiring

The following diagram shows various possible external power connections for DVP PLC. When wiring AC power, the 'Live' cable should be connected to the ' L' terminal and the ' Neutral' cable should be connected to the ' N' terminal. When wiring DC power, the ' positive' cable should be connected to the ' +' terminal and the negative should be connected to the '-' terminal. At no time should the power supply terminals be connected to any other terminal on the PLC.

AC Input Type	 ■ The +24V supply output is rated at 0.4 Amperes. DO NOT connect external power supply to this terminal. ■ FUSE Protection: there are internal fuses on all DVP PLCs. However, the fuse does not guarantee the prevention of DVP PLC damage, but it will provide added protection.

When DC voltage is supplied to the PLC, make sure the power is at terminals 24VDC and 0V (power range is 20VDC~26VDC). When voltage is lower than 17.5VDC, PLC will stop operating, all outputs will turn OFF and the ERROR LED will flash continuously.

DC Input Type	

■ Safety Guidelines

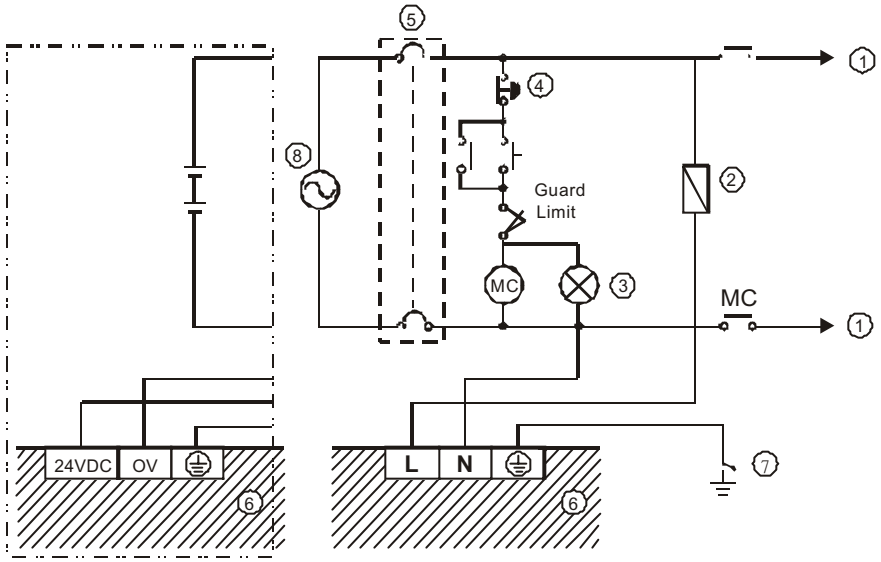
Providing a safe operating environment for personnel and equipment is your responsibility and should be your primary goal during system planning and installation. Automation systems can fail and may result in situations that can cause serious injury to personnel or damage to equipment. Do not rely on the automation system alone to provide a safe operating environment. You should use external electromechanical devices, such as relays or limit switches, which are independent of the PLC application to provide protection for any part of the system that may cause personal injury or damage.

DVP-series PLC input power supply includes two inputs: AC input and DC input. Please take a note of listed items when operating the PLC.

1. When voltage fluctuations are larger than the specified value, connect a constant-voltage transformer.
2. Connect the AC input (100Vac to 240Vac) to terminals L and N. Any AC voltage connected to the +24V terminal or input point will permanently damage the PLC.
3. Service power supply: If the system being installed uses the service supply from both the PLC and powered extension block, then both these units should have their 0V terminals linked. DO NOT however, link the 24V terminals; External DC supplies should not compromise the SELV aspects of the DVP product.

4. When the Momentary Power Loss Time is less than 10ms, the PLC will continue its operation without any interruption. When the Momentary Power Loss Time is longer than 10ms or the input voltage has dropped below minimum values, the PLC will stop its output. When the power returns the PLC will automatically resume operation.

Recommended Wiring for Input Power and Safety Devices.



① Power supply for AC loads
② Circuit protection device (3A Limit)
③ Power ON pilot indicator
④ Emergency stop
The machinery must provide a quick manual method of disconnecting all system power. The disconnect device or switch must be clearly labeled “Emergence Stop”.
⑤ Circuit isolation device (System Power Disconnect)
Using electromechanical devices, such as master control relays and/ or limit switches, to prevent accidental equipment startup at an unexpected time. These devices should be installed in such a manner to prevent any machine operations from occurring.
⑥ DVP MPU (main processing unit)
⑦ Grounding
⑧ Power supply
AC: 100~240VAC, 50/60Hz
DC: 24VDC

■ Input Point Wiring

Prior to performing any wiring, always turn the power off. In some special circumstance, if the user needs to perform wiring to input points while power is on, always stop the PLC. Otherwise, output points may be activated and cause accidently damage to the systems.

PLC Isolation Boundaries: PLC circuitry is divided into three main regions separated by isolation boundaries. Electrical isolation provides safety, so that a fault in one area does not damage another. A transformer in the power supply provides magnetic isolation between primary and secondary sides. Opto-couplers provide optical isolation in Input and Output circuits. This isolates logic circuitry from the field side, where factory machinery connects. Note that the discrete inputs are isolated from the discrete outputs, because each is isolated from the logic side.

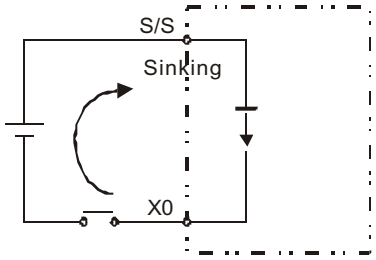
All versions of the DVP PLC have Input / Output circuits that can connect to a wide variety of field devices. DC Input PLCs have two modes of operation: SINK and SOURCE.

Sink = Current flows into the common terminal S/S

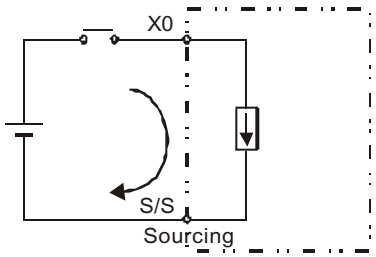
Source = Current flows out of common terminal S/S

For example, we simply connect the common terminal S/S to the supply source(+). By adding the switch, between the supply(-) and the input, we have completed the circuit. Below are two circuit diagrams showing both the sinking and sourcing inputs.

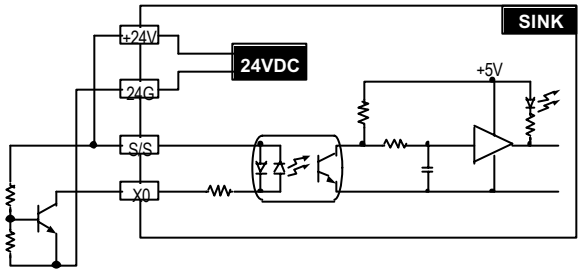
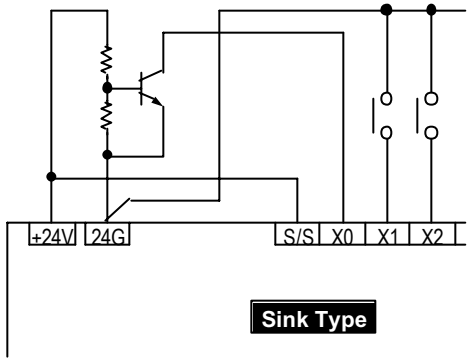
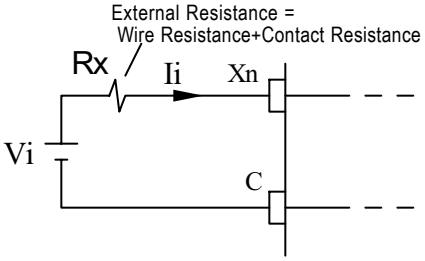
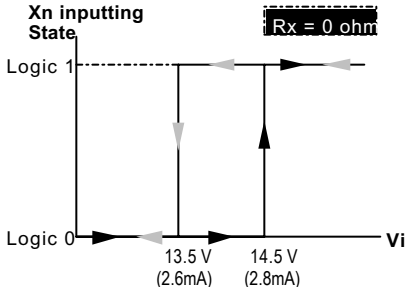
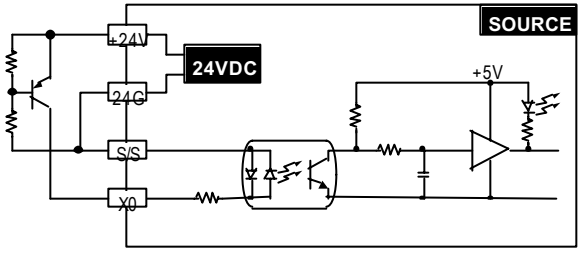
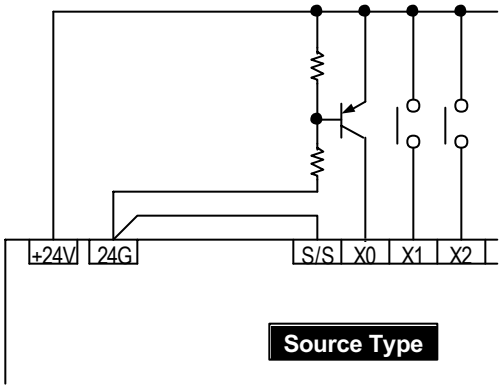
Sinking input



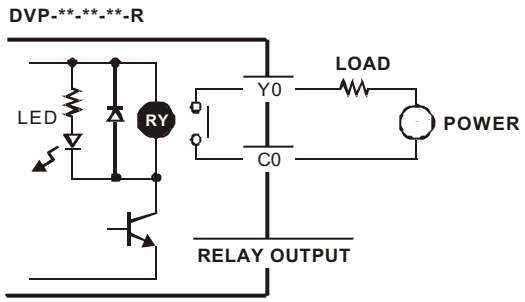
Sourcing input

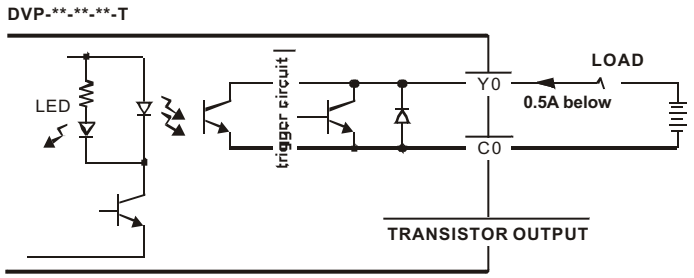


Practically Wiring

	Input Point Loop Equivalent Circuit	Wiring Loop	Voltage Activation Level
DC Type （DC Signal IN） SINK Mode			  If Rx = 100 ohm, the Vi must be greater than 14.8V (14.5V+2.8mA*100), and Xn input state will be logic 1.
DC Type （DC Signal IN） SOURCE Mode	Input Point Loop Equipvalent Circuit 	Wiring Loop 	

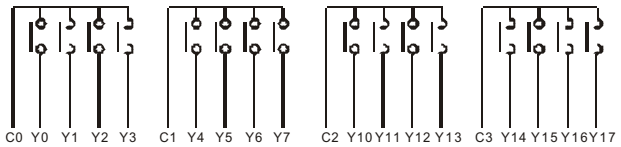
■ Output Point Wiring





1. There are three kinds of DVP-Series PLC outputs: Relay, SSR and Transistor. All relays used in DVP series PLC have passed the standard of IEC 947-5-1 under AC-15 (the rated current and voltage) specification for a cycle test of 6050 times.

2. Be careful with the connection of the common terminals when wiring output terminals. For example, when wiring DVP32ES00R, note that there are sixteen normally-open SPST relays available. They are organized into 4 groups with individual commons. The figure below shows the relays and the internal wiring of the PLC. Note that each group is isolated from the other 3 relay groups:



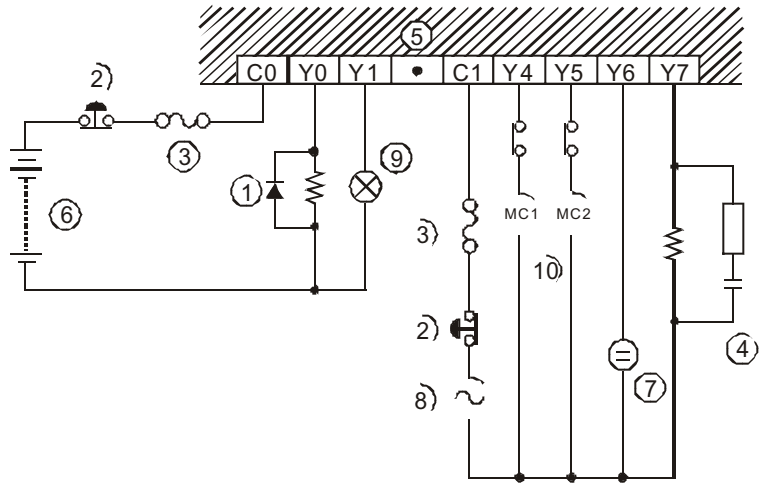
Output terminals Y0 to Y3 share one common terminal C0, and Y4 to Y7 share C1, Y10 to Y13 share C2, Y14 to Y17 share C3.

Other Models please refer to Chap 4.2.

⚠ Output Point Overload Capacity

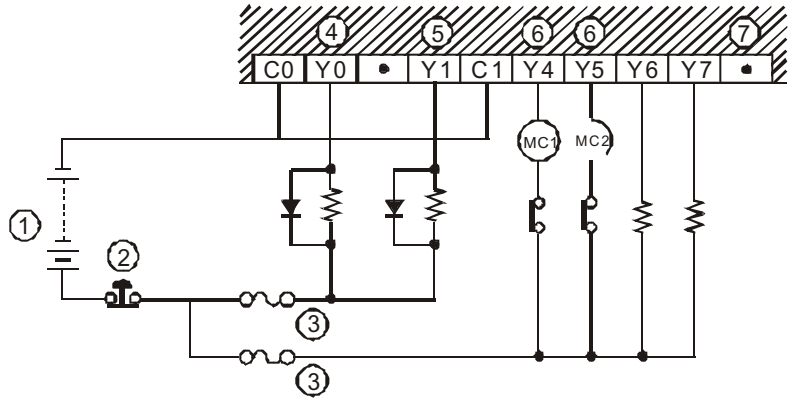
Each output point is capable of 200% of rated current for 5 minutes; the overload capacity of the common point is 150% of rated current for 2 minutes. If the system is over this limit, the PLC output contacts may be damaged and the internal circuit board may be damaged.

◎ Relay Output Wiring Methods



①	Surge absorbing diode: increases relay contact life
②	Emergency stop: use an external switch
③	Fuse: 5 to 10A for every 4 output points to protect the PLC's output circuit.
④	Surge absorber: reduces noise on AC inductive loads
⑤	Unused terminal: do not connect
⑥	DC supply
⑦	Neon lamp
⑧	AC supply
⑨	Incandescent lamp
⑩	Mutually exclusive outputs: Use external hardware interlocks, as well as those in the PLC program, for maximum safety.

◎ Transistor Output Wiring Methods



①	DC supply
②	Emergency stop
③	Fuse
④	If Y0 is used as a pulse train output with PLSY, use a pull up resistor to ensure the output current is greater than 0.01A for correct operation.
⑤	If Y1 is used with PWM, use a pull up resistor to ensure the output current is greater than 0.01A for correct operation.
⑥	Mutually exclusive outputs: use external hardware interlocks, as well as those in the PLC program, for maximum safety.
⑦	Unused terminal: do not connect

■ Power Indication

1. The "POWER ON" LED on the MPU and the Extension Unit will be lit if the power is on. Or if the LED is not lit, it is an indication that the PLC's 24VDC terminal is overloaded, and it is thus necessary to remove the wiring on terminals +24V and 24G, and to provide a 24VDC power supply for each terminal respectively. Moreover, if the ERROR LED blinks continuously, it suggests that the power supply of PLC (+24 V) is low.
2. The "LOW V." LED on the Extension Unit is an indication that the power voltage is low and thus, all outputs of the Extension Unit should be turned off.

■ Preparation, Operation and Test

1. Prior to applying power, please verify the power lines and input/output wiring is correct.
2. If the ERROR LED is not blinking after using the peripheral equipments to write the program into the MPU. It means that the program is legal and the PLC will wait for a RUN command.
3. Enter the RUN command. The RUN LED will light up.
4. Use a HPP or PC to monitor the settings & the data sets of the Timer, the Counter, and the Data Register, and consequently, to perform the ON/OFF functions for the output points. If the ERROR LED is lit (but not blinking), it means that the program setting is over the preset overtime limit, and with this occurrence, please double check the program and perform the ON/OFF functions again (at the time when PLC is returning to the STOP mode).

■ PLC Input/Output Reaction Time

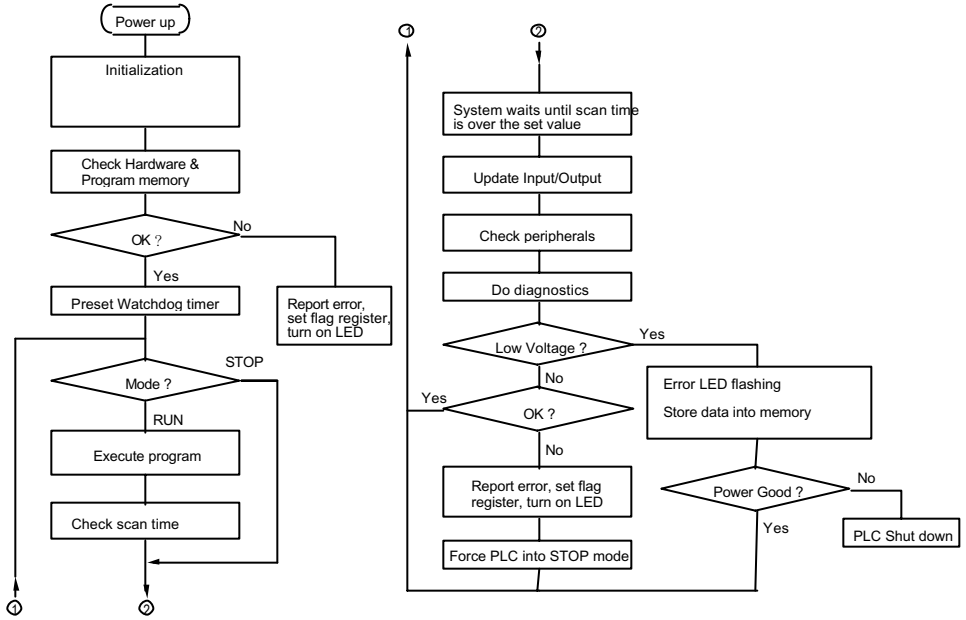
The total reaction time from the input signal to the output operation is calculated as follow:

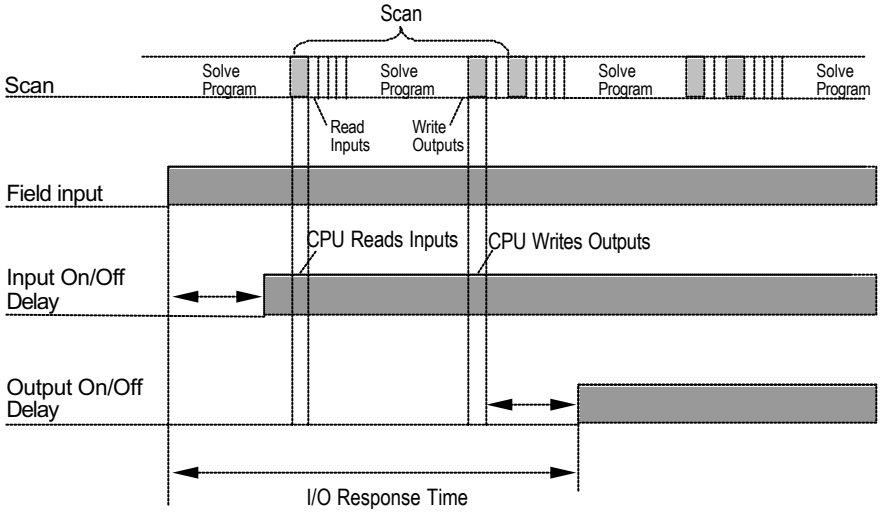
Reaction Time = input delay time +
program scan time +
output delay time

Input delay time	10ms (factory default), 0~15ms adjustable. Please refer to the usage of special devices D1020~1021.
Program scan time	Please refer to the usage of special device D1010.
Output delay time	Relay module: 10ms Transistor module: 20~30 μ s

Please refer to following diagrams for more detail.

- PLC Operation flowchart and timing





In this case, you can calculate the response time by simply adding the following items:

Input Delay + Scan Time + Output Delay = Response Time

■ Basic Instructions Table

Instructions	Functions	Operand	Steps
LD	Load a contact	S, X, Y, M, T, C	1
LDI	Load b contact	S, X, Y, M, T, C	1
AND	Series connection-a contact	S, X, Y, M, T, C	1
ANI	Series connection-b contact	S, X, Y, M, T, C	1
OR	Parallel connection-a contact	S, X, Y, M, T, C	1
ORI	Parallel connection-b contact	S, X, Y, M, T, C	1
OUT	Output Coil	S, Y, M	1
SET	Latch (ON)	S, Y, M	1
ANB	Series connection (Multiple Circuits)	None	1
ORB	Parallel connection (Multiple Circuits)	None	1
MPS	Stores the operation result	None	1
MRD	Reads the operation result (pointer won' t move)	None	1
MPP	Reads, then clears the operation result	None	1
NOP	No operation action	None	1
MC	Master control START command	N0~N7	3
MCR	Master control RESET command	N0~N7	3
RST	Clears the contact or the register	S, Y, M, T, C, D	3
P	Pointer	P0~P63	1
I	Interrupt pointer	I001 / I101 / I201 / I301	1
END	Program END	None	1

☞ The following instructions have API codes associated with them. When using the HPP, users may input API codes, or use the specified keys, TMR, CNT, and DCNT to generate their program.

Table 6.1

API	Instructions	Functions	Operands	Steps
96	TMR	16-bit Timer	T-K or T-D	4
97	CNT	16-bit Counter	C-K or C-D (16-bit)	4
97	DCNT	32-bit Counter	C-K or C-D (32-bit)	6

☞ When using an HPP, the items below may only be entered by their API codes.

API	Instructions	Functions	Operands	Steps
89	PLS	Rising-edge output	Y, M	3
90	LDP	Rising-edge detection operation	S, X, Y, M, T, C	3
91	LDF	Falling-edge detection operation	S, X, Y, M, T, C	3
92	ANDP	Series connection command for the rising-edge detection operation	S, X, Y, M, T, C	3
93	ANDF	Series connection command for the falling-edge detection operation	S, X, Y, M, T, C	3
94	ORP	Parallel connection command for the rising-edge detection operation	S, X, Y, M, T, C	3
95	ORF	Parallel connection command for the falling-edge detection operation	S, X, Y, M, T, C	3
98	INV	Inverting operation	None	1
99	PLF	Falling-edge output	Y, M	3

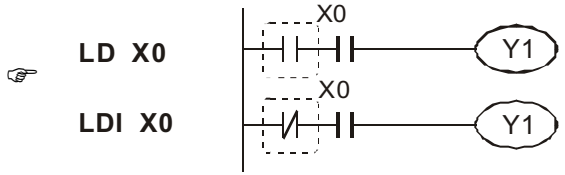
■ Step Ladder Instructions

Instructions	Functions	Operands	Steps
STL	Step transition ladder start instruction	S	1
RET	Step transition ladder return instruction	None	1

■ Contact Instructions

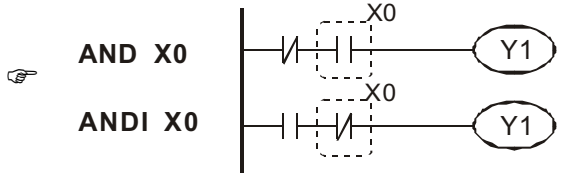
Instruction	Operands					
LD / LDI	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

LD is the contact A operation instruction and LDI is the contact B operation instruction.



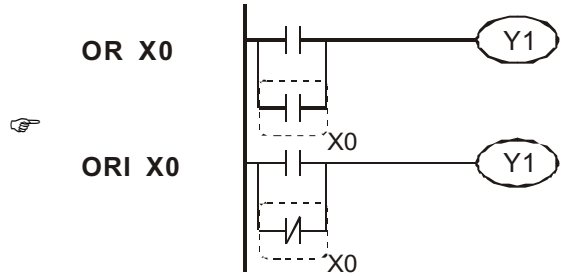
Instruction	Operands					
AND / ANI	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

AND is the series connection instruction of one A contact and ANI is the series connection instruction of one B contact.



Instruction	Operands					
OR / ORI	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

OR is the parallel connection instruction of one A contact and ORI is the parallel connection instruction of one B contact.

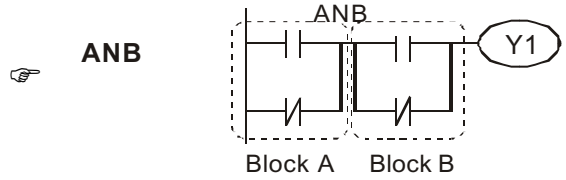


■ Connection Instructions

Instruction	Operands
ANB / ORB	None

This instruction performs the AND operation of block A and block B, and uses it as an operation result.

The symbol of ANB is not a contact symbol but a connection symbol. ANB can be written consecutively up to 8 times. If more ANBs are written consecutively, error indication is given by self-check function and corresponding error code is stored in special register D1004.



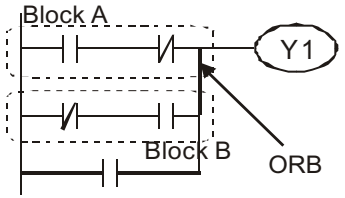
ORB instruction performs the OR operation of block A and Block B, and uses it as an operation result.

ORB performs parallel connection of circuit block with two or more contacts. For parallel connection of circuit blocks which have only one contact, OR and ORI are used and ORB is not required.

The symbol of ORB is not a contact symbol but a connect symbol.

ORB can be written consecutively up to 8 times. If more ORBs are written consecutively, error indication is given by self-check function and corresponding error code is stored in special register D1004.

ORB



Instruction	Operands
MPS / MRD / MPP	None

MPS:

Stores the operation result (ON/OFF) immediately preceding the MPS instruction.

The MPS instuction can be used up to 8 times. However, if an MPP instruction is used between any two MPS instructions, then the total number of MPS instructions is reduced by 1.

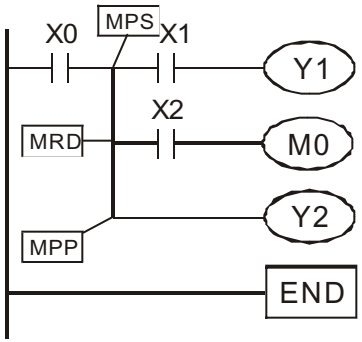
MRD:

Reads the operation result stored by the MPS instruction, and uses the operation result, starting at the next step.

MPP:

Reads the operation result stored by the MPS instruction, and uses the operation result, starting at the next step.

Ladder Diagram



Commands

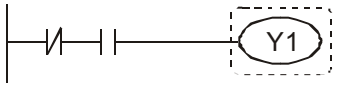
```
LD    X0
MPS
AND   X1
OUT   Y1
MRD
AND   X2
OUT   M0
MPP
OUT   Y2
END
```

■ Output Instructions

Instruction	Operands					
OUT	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓		✓	✓		

This instruction outputs the operation result for the elements preceding the OUT instruction.

OUT Y1



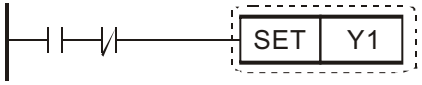
Operation Result	OUT Instruction		
	Coil	Contact	
		No contact	NC contact
OFF	OFF	Non-continuity	Continuity
ON	ON	Continuity	Continuity

Instruction	Operands					
SET	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓		✓	✓		

When the SET input turns on, the specified device is turned on.

The specified device remains on even if the SET input turns off. The device can be turned off by the RST instruction.

☞ **SET Y1**



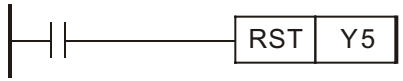
Instruction	Operands					
RST	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

When the RST input turns on, the specified device changes as described below:

Device	Status
S, X, Y, M	Coil and contact are turned off.
T, C	Preset value is set to 0, and coil and contact are turned off.
D	Content is set to 0.

When the RST input is off, the status of device does not change.

☞

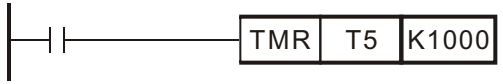


Instruction	Operands	
TMR	T-K	T0~T127, K0~K32767
	T-D	T0~T127, D0~D1143

When the operation result of instructions preceding the TMR instruction are on, the coil of timer turns on and counts up to the set value. When the timer times out (counted value >= set value), the contact is as indicated below.

NO contact	Continuity
NC contact	Non-continuity

☞



When the operation result of instructions preceding the TMR instruction change from ON to OFF, the following occurs.

Timer Coil	Preset Value of Timer	Before Time Out		After Time Out	
		NO contact	NC contact	NO contact	NC contact
OFF	0	Non-continuity	Continuity	Continuity	Non-continuity

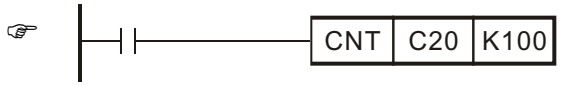
After the timer has timed out, the status of the contact will not change until the RST instruction is executed.

A negative number (-32768 to –1) cannot be set as a set value.

Instruction	Operands	
CNT	C-K	C0~C127, K0~K32767
	C-D	C0~C127, D0~D1143

When the operation result of instructions preceding the CNT instruction has changed from OFF to ON, 1 is added to the count value. When the counter has counted out (count value = set value), the state of the contact is as indicated below.

NO contact	Continuity
NC contact	Non-continuity



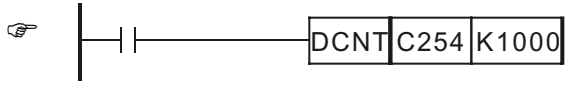
When the operation result of the instructions preceding the CNT instruction remain on, counting is not performed. (It is not necessary to convert the count input into a pulse.)

After the counter has counted out, the count value and the status of the contact will not change until the RST instruction is executed.

A negative number (-32768~-1) cannot be used as a set value. When the set value is 0, the same processing as for 1 is performed.

Instruction	Operands	
DCNT	C-K	C235~C254, K-2147483648~K2147483647
	C-D	C235~C254, D0~D1143

Counters C232 to C255 are used for high speed counters. When the operation result preceding the DCNT has changed from OFF-ON, 1 is added to the count value. When the counter has counted out (count value = set value), the state of the output contact is changed (ON or OFF).



The counted value is not cleared when the operation result is OFF. Use the RST C2XX instruction to clear the counted value and to turn OFF the contact.

■ Master Control Instructions

Instruction	Operands
MC / MCR	N0 ~ N7

MC:

MC is master control start instruction. When the ON/OFF command for the MC is on, operation results from MC to MCR remain unchanged.

Scanning between the MC and MCR instructions is executed even when the ON/OFF command for the MC instruction is OFF. Scan time does not therefore become shorter.

When ON/OFF command for the MC is off, the operation result of MC to MCR is as indicated below.

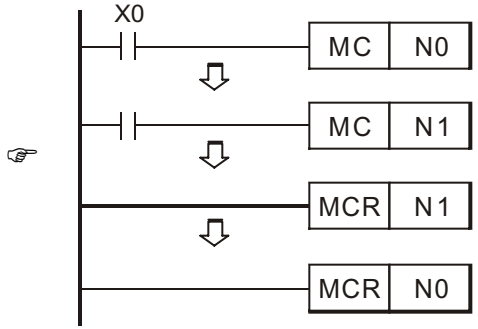
Timers	Count value becomes 0. Coil and contact turn OFF
Counter	Coil turns OFF. Count value and contact hold present status.
Devices in the OUT instruction	All turn OFF.
Devices in the SET, RST instructions	Hold present status.

MCR:

MCR is a master control reset instruction and indicates the end of master control range.

Do not use a contact instruction before the MCR instruction.

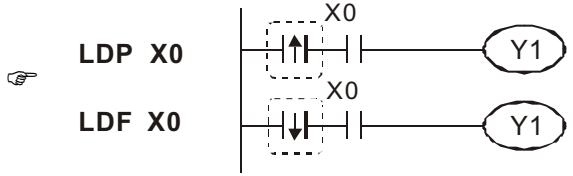
The MC instructions can be used by nesting. Range of each MC instruction is identified by a nesting number. Nesting numbers are used in the range of N0 to N7.



■ Contact Rising/Falling edge Instructions

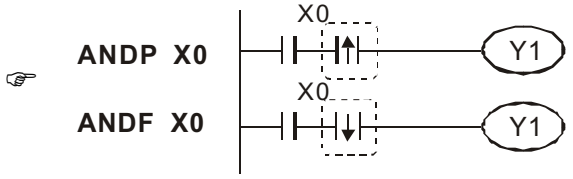
Instruction	Operands					
LDP / LDF	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

Rising/Falling-edge detection operation instruction



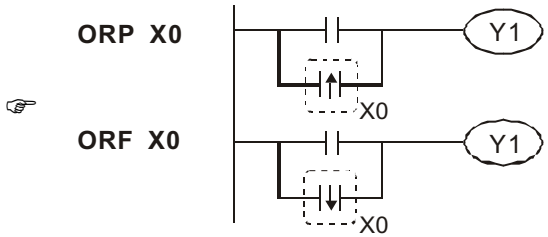
Instruction	Operands					
AND / ANDP	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

Series connection command for the rising/falling-edge detection operation.



Instruction	Operands					
ORP / ORF	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
	✓	✓	✓	✓	✓	✓

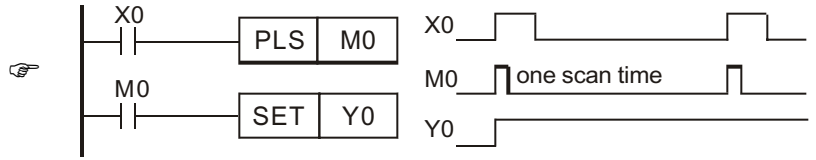
Parallel connection command for the rising/falling-edge detection operation.



■ Rising/Falling edge Output Command

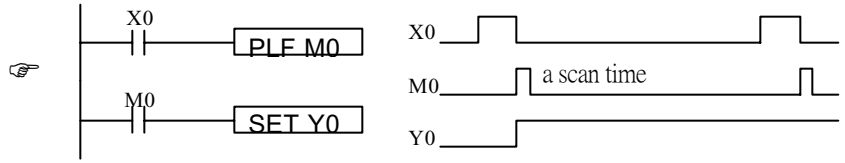
Instruction	Operands					
PLS	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
			✓	✓		

PLS command: the rising-edge output command. When X0=OFF→ON (the rising-edge is touched off), the PLS command will be executed, and M0 will send out one pulse. Length of this pulse is one scan time.



Instruction	Operands					
PLF	S0~S127	X0~X177	Y0~Y177	M0~M1279	T0~T127	C0~C127 C235~C254
			✓	✓		

PLF command: the falling-edge output command. When X0= ON→OFF (the falling-edge is touched off), the PLF command will be executed, and M0 will send out one pulse. Length of this pulse is one scan time.

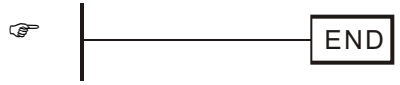


■ Termination Instructions

Instruction	Operands
END	None

This instruction indicates the end of program. At this step, the scan returns to step 0.

The END instruction cannot be used midway through the sequence program or subsequence program. If END processing is necessary halfway through the program, use the FEND instruction.



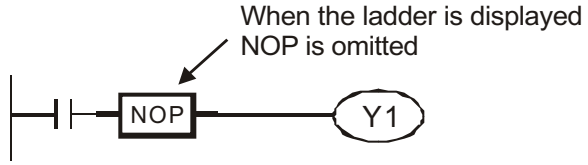
■ Other Instructions

Instruction	Operands
NOP	None

This is a no-operation instruction and has no effect on the previous operation.

NOP is used in the following cases:

1. To provide space for debugging of sequence programs.
2. To delete an instruction without changing the number of steps. (Overwrite with NOP)
3. To delete an instruction temporarily.



Instruction	Operands
P / I	P0 ~ P63 / I001, I101, I201, I301

Pointers (P)

Pointers are used with the jump instructions (CJ, CALL) in two different ways as follows.

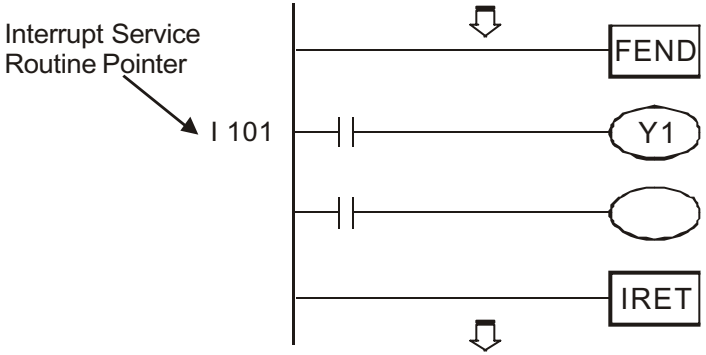
1. Designation of the JUMP destination (CJ) and at the head of destination (Label).
2. Designation of the subroutine destination (CALL) and at the head of the subroutine program (Label).

A label number cannot be used at more than one place. If used, more than once an error will occur.

☞ Pointer P10

Interrupt Pointers (I) :

Interrupt pointers are used as the label at the head of each interrupt program. Each interrupt program begins with an interrupt pointer and ends with the IRET instruction.



☞

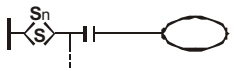
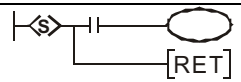
Instruction	Operands
INV	None

Inverting the operation result and use the new data as an operation result.

☞ **INV**

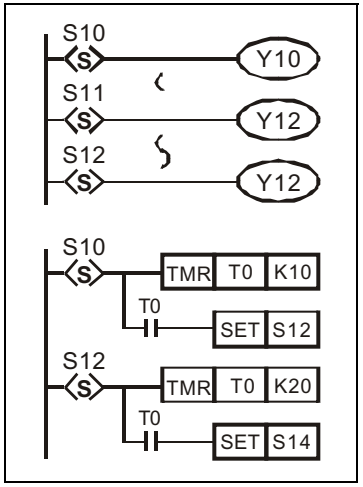
■ Step Ladder Commands

- Symbols and functions:

Command	Functions	Symbol and devices	
STL [Sn]	Step ladder diagram starts		● Command Length: 1 Step ● Device [Sn]: S0~S127 ● Begins from S0~S9 ● Step S can not be repeated
RET	End of step ladder diagram	 RET (return command must be added at the end of STL)	● Command length: 1 Step ● Step point **S** ● RET command is used at the end of the step ladder diagram that begins with S0 to S9.

- Command Instruction

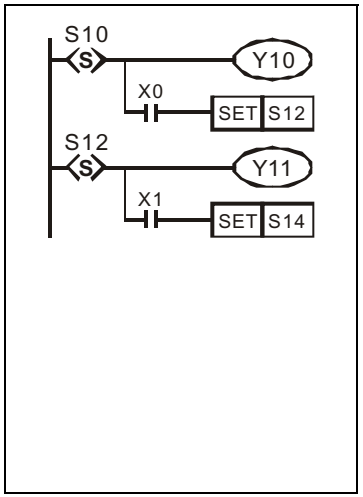
Example:



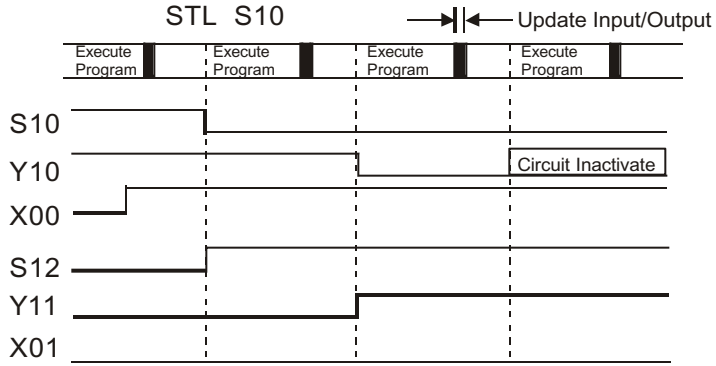
☞ When step point Sn=ON, the subroutine is active, the action will delay for one scan time. Whereas Sn is OFF, the subroutine is inactive.

☞ In the example, the same device (Y12) can be used in different step points. That is, when S11 or S12 step point is activated (ON), Y12 will output. Y2 will be closed during the process that S11 transfer to S12. And then output Y2 after S12 is ON. In this situation, no matter S11 or S12=ON, Y12 will always be ON.

☞ The timer can be repeatedly used in discontinuous step points. (This is a special feature of the step ladder diagram. However, users should try to avoid repeated output relays. You should also avoid using the same coil number that used in step ladder diagram after returning to general ladder diagram.



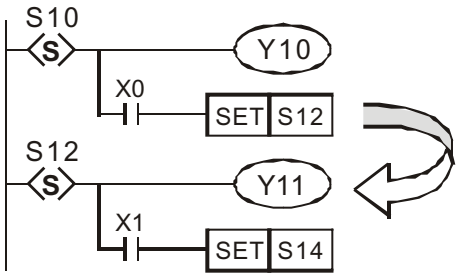
☞ When step point S10 and S12 transition simultaneously (S10➔OFF, S12➔ON, there is a delay for one scan time), and the output Y10 and Y11 will not be ON simultaneously.



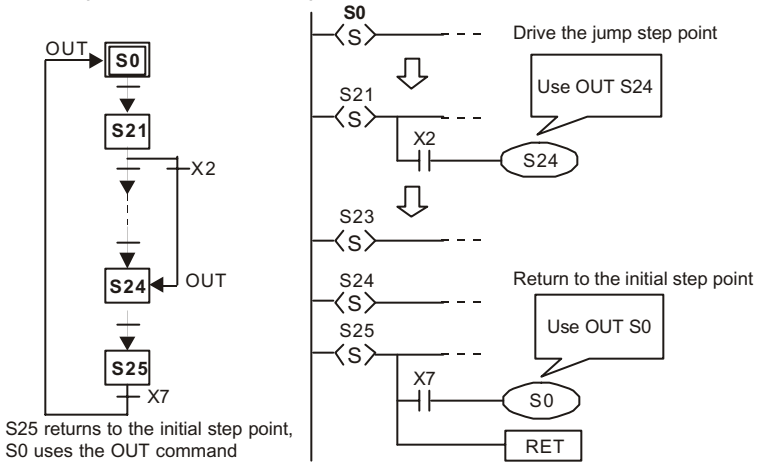
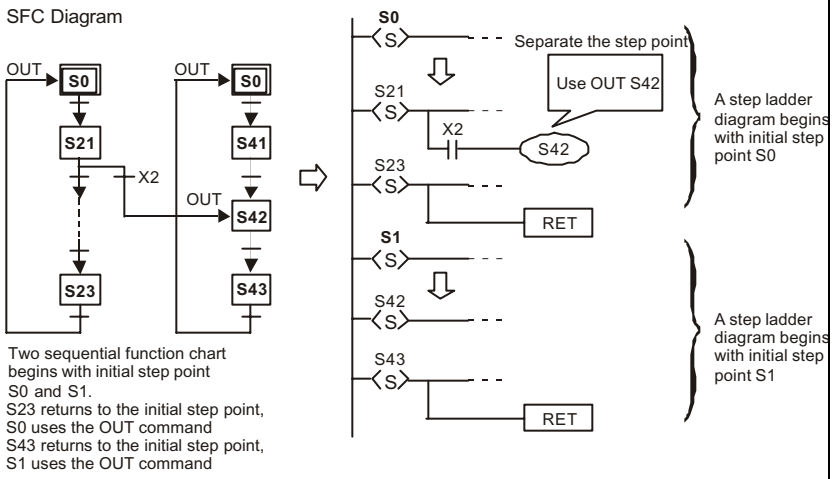
- Step Point Transition

Commands SET Sn and Out Sn are for activating another step point. There are different occasions when these commands are used. Please see the following examples.

SET Sn	The step ladder control that begins with S0 to S9 will move to the next step point, the action of the previous step point will be deleted.

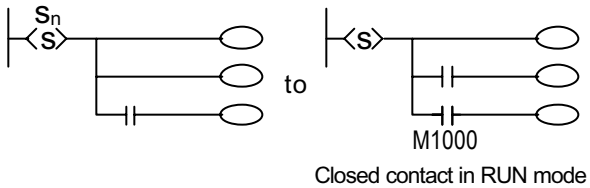


When SET S12 executes, step point will move from S11 to S12. S11 and the output of (Y10) will be deleted.

OUT Sn	Return back to initial step point and step point jump up or jump down to the point that is not sequenced in order within the program or transitions to a different step ladder program. Once the status is shifted, outputs from all the previous motion status points will be deleted.	
	① In a step ladder program, return back to initial step point. ② In a step ladder program, step point jump up or jump down to the point that is not sequenced in order. 	③ Transition to a different step ladder program (begin with different initial step point). 

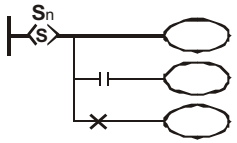
- Note

1. When using LD or LDI commands, user will not be able to write in the no-contact program. The ladder diagram will have to be refined as followed:



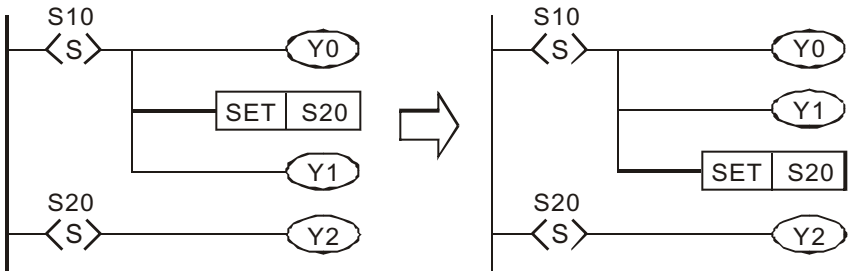
Closed contact in RUN mode

☞ If there already exists an input device, inputs that follow this input device cannot be connected to the end of the STL command directly, for that the current condition requires the operation outcome from both the STL status and the input device. As shown follows.

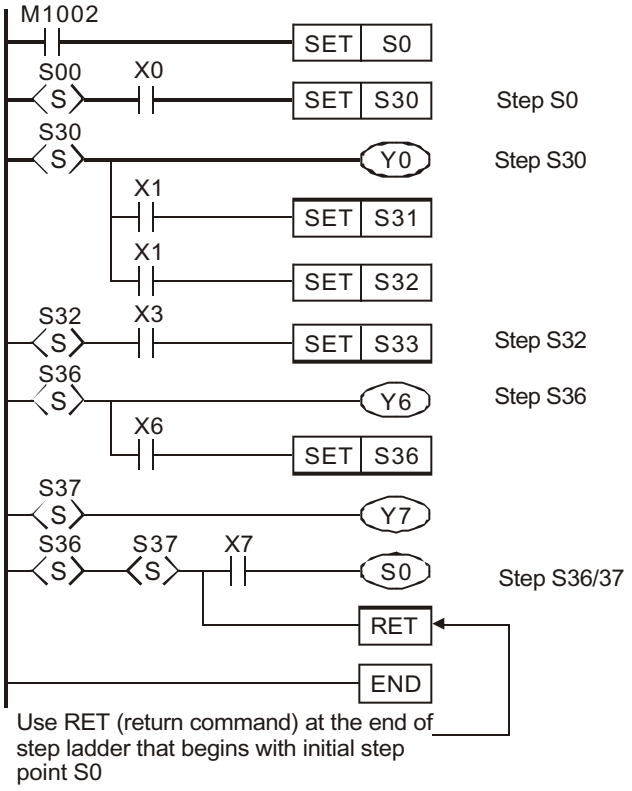


2. Every subroutine of a step point is the same as a general ladder diagram, but there are some limitations to the commands.
   1. Do not use MC/MCR commands in step points.
   2. STL commands cannot be used in Sub-programs and Interrupt programs.
   3. Try to avoid using CJ commands in the STL program, it will complicate the entire program.

3. It is always better to finish all commands before moving to the next step point. The SET Sn command is better to be placed at the end of step point, as shown below.



☞ Step ladder diagram:

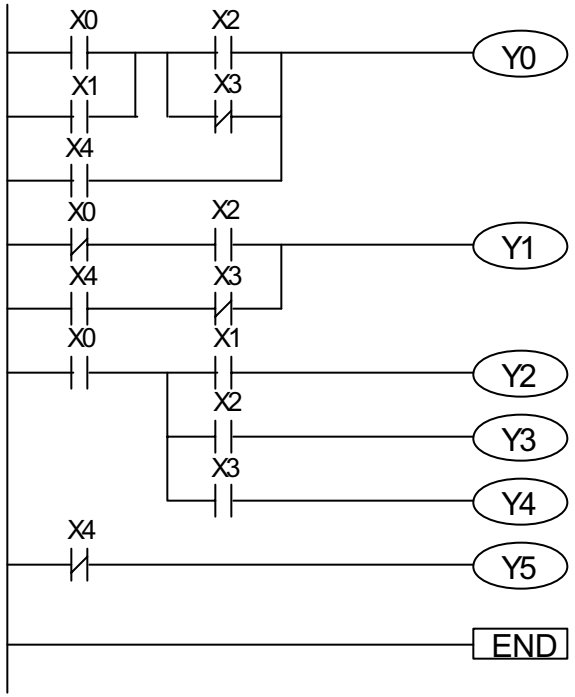


◆ Example: Programming 1 (Excluding Step Ladder Instructions STL/RET)

◆ Using Instruction Coding

Step	Instruction	
0	LD	X0
1	OR	X1
2	LD	X2
3	ORI	X3
4	ANB	
5	OR	X4
6	OUT	Y0
7	LDI	X0
8	AND	X2
9	LD	X4
10	ANI	X3
11	ORB	
12	OUT	Y1
13	LD	X0
14	MPS	
15	AND	X1
16	OUT	Y2
17	MRD	
18	AND	X2
19	OUT	Y3
20	MPP	
21	AND	X3
22	OUT	Y4
23	LDI	X4
24	OUT	Y5
25	END	

◆ Using Ladder Diagram Coding



◆ Example: Programming 2 (including Step Ladder Instructions STL/RET)

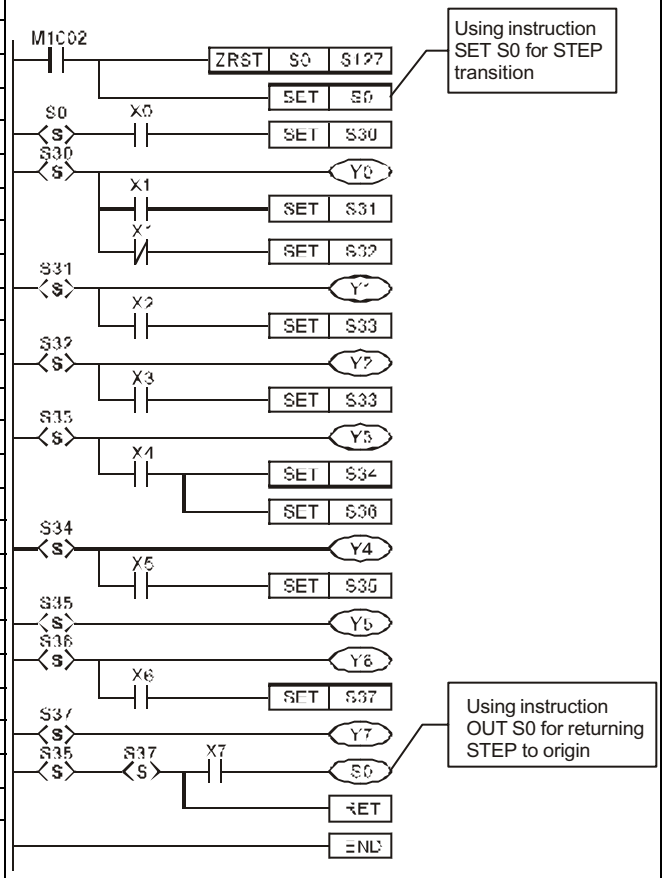
Instruction List

	Step	Instruction		Step	Instruction
	0	LD M1002		29	STL S34
	1	ZRST S0 S127		30	OUT Y4
	6	SET S0		31	LD X5
	7	STL S0		32	SET S35
	8	LD X0		33	STL S35
	9	SET S30		34	OUT Y5
	10	STL S30		35	STL S36
	11	OUT Y0	2*	36	OUT Y6
	12	LD X1		37	LD X6
	13	SET S31		38	SET S37
	14	LDI X1		39	STL S37
	15	SET S32		40	OUT Y7
	16	STL S31		41	STL S35
1*	17	OUT Y1		42	STL S37
	18	LD X2		43	LD X7
	19	SET S33		44	OUT S0
	20	STL S32		45	RET
	21	OUT Y2		46	END
	22	LD X3			
	23	SET S33			
	24	STL S33			
	25	OUT Y3			
	26	LD X4			
	27	SET S34			
	28	SET S36			

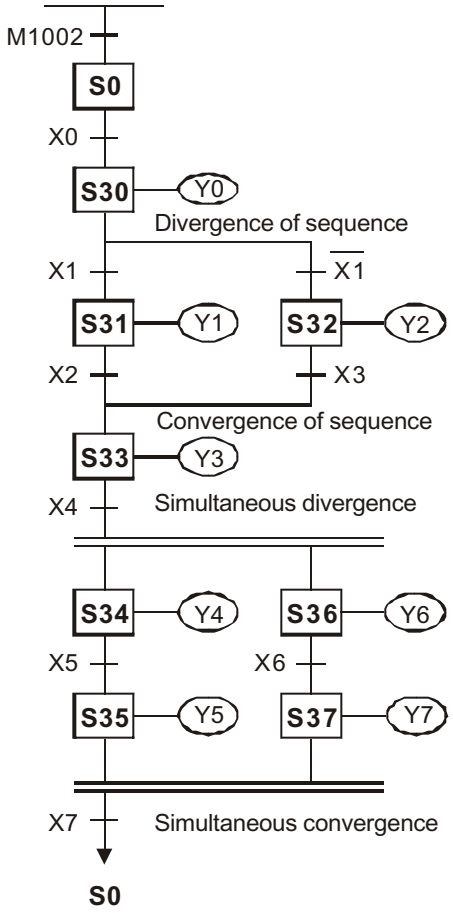
1*: Divergence/convergence of sequence

2*: Simultaneous divergence/convergence sequences

Ladder Diagram



Step Transition Flowchart



■ Application Instructions Structure

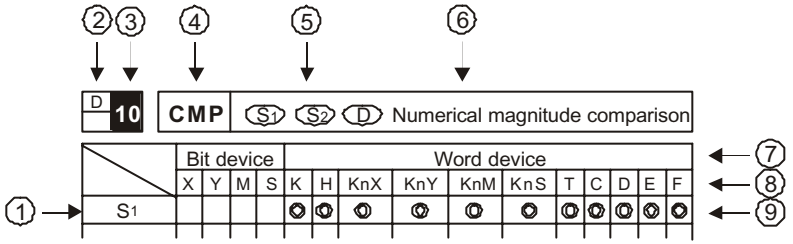
1. Many instructions may be divided into an instruction part and a device as follows:

   Instruction part : Indicates the function.

   Device : Indicates the data for use with that instruction.

2. The application instructions structure may be largely classified as follows with the instruction part and device (s) combined:
   - Instruction part
     Retains the device status and mainly controls the program.
     Example: FEND, IRET,…
   - Instruction part + Source device + Destination device
     Operation is performed using the destination data and operation result is stored to the destination.
     Example: MOV, …
   - Instruction part + Source 1 device + Source 2 device + Destination 1 device + Destination 2 device
     Operation is performed using the source 1 data and source 2 data, and the operation result is stored to the destination.
     Example: ZCP, SFTL, RS, …
   - Others…Combination of the formats above.

■ Instructions Format

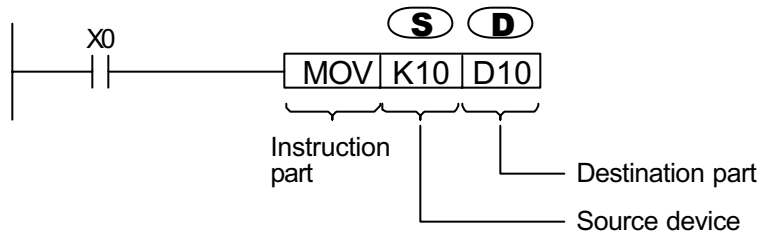


Explanations:

① Operand
② Indicates 32-bit instruction (D is added to the head of instruction, example as above: DCMP)
③ Indicates API number
④ Instruction
⑤ Operand format of the instruction
⑥ Describes the instruction function
⑦ Device types
⑧ Device name
⑨ Symbol ◎ are given to devices which can be used for this operand

■ Application Instructions Input

The application instructions of DVP-Series PLC are controlled by command codes API 00 to API 246. Each command code has its own meaning, for example, API 12 stands for MOV (move data). When using ladder diagram Editor to input programs, you will need to type in the instruction "MOV". If using the HPP to input the program, we will have to enter the API command codes. Each application instruction has its unique operand.



This instruction is to move the value of (S) operand to the appointed (D) operand.

(S):

1. Source data used for operation.
2. Source data may be A:
   - Constant
     Specify the numeric value used for the operation. This value is set while the program is being written and cannot be changed when the program is running.
   - Bit device, word device
     Specify the device, which stores the data used for the operation. The data must be stored to the specified device before the operation is initiated. By changing the data to be stored to the specified device during program run, the data used with the instruction can be changed.

Source operand: if there are more than 1 source operand, then we use (S1), (S2).

(D):

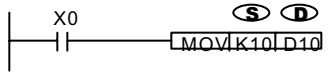
1. Stores data after operation is performed.

Destination operand: if there is more than one operand, then we use (D1), (D2).
If the operand may only be represented as a constant K, H or register D then we will use (m1), (m2) or (n1), (n2).

■ The Length of Operand

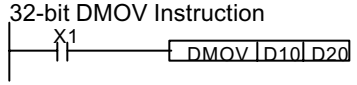
The length of Operand can be divided into two groups: 16-bit and 32-bit to process different length data. A "**D**" before an instruction separates 32-bit from 16-bit instructions.

16-bit MOV Instruction



K10 has been sent to D10.

32-bit DMOV Instruction



Data of (D11,D10) have been sent to (D21,D20)

■ Data format

X, Y, M, S are only be single point ON/OFF, these are defined as a BIT. However, 16-bit (or 32-bit) T, C, D are data registers and are defined as WORDs. We also can add Kn in front of X, Y, M and S to be defined as WORDs, whereas n=1 means 4-bit. So16-bit can be described from K1 to K4, and 32-bit can be described from K1 to K8. For Example, K2M0 means there are 8-bit from M0 to M7.

Move the contents of M0 to M7 to D10 segments 0 to 7, and segments 8 to 15 are set to 0.

■ Bit processing

Bit processing is performed when a bit device (X, Y, M, S) has been specified. Either 1 bit or digit specification processing may be selected.

- 1-bit processing

  When the sequence instruction is used, no more than one bit may be specified.

  For example: LD X10, OUT Y0

- Digit specification processing

  When application instructions are used, the number of digits may need to be specified for the bit device. Up to 32 points can be specified in 4 point increments.

1. 16-bit instruction: K1 to 4 (4 to 16 points)

   When there is digit specification on the source (S) side, the range of numeric values handled as source data are shown below.

- Specified Number of Digits (16-bit Instruction)

K1 (4 points)	0 to 15
K2 (8 points)	0 to 255
K3 (12 points)	0 to 4095
K4 (16 points)	-32768 to 32767

When there is digit specification on the Source side, the number of points set by the digit specification is used on the destination side.

2. 32-bit instruction: K1 to 8 (4 to 32 points).

When there is digit specification on the source (S) side, the range of numeric values handled as source data are as shown below.

- Specified Number of Digits (32-bit Instruction)

K1 (4 points)	0 to 15
K2 (8 points)	0 to 255
K3 (12 points)	0 to 4095
K4 (16 points)	0 to 65535
K5 (20 points)	0 to 1048575
K6 (24 points)	0 to 167772165
K7 (28 points)	0 to 268435455
K8 (32 points)	-2147483648 to 2147483647

■ Handling of Numeric Values

In the DVP PLC series, there are instructions, which handle numeric values in 16 bits and 32 bits format. The highest bits of 16 bits and 32 bits are used for the judgment of positive and negative numbers. Numeric values handed by 16 bits and 32 bits are as follows:

16 bits : -32768 to 32767
32 bits : -2147483648 to 2147483647

- **Double word (32-bit data) processing**

  32-bit data is stored using digit specification of K1 to 8 when it is stored in bit or in word devices.

- Storing data in bit devices

  Refer to **Digit Specification Processing**.

- Storing data in word devices

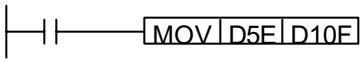
  1) Two consecutive word devices are used to store 32-bit data.

  2) To store the data of bit devices with which digit specification of K1 to K8 was done, refer to **Digit Specification Processing**.

  3) Cautions : If the storing word device is assigned to the final device number of each device, an error will occur.

■ Indirect Assigned Method

E and F represent constants used as operands. They are the same as other operands and may be moved, compared, and be used in the word devices (KnX, KnY, KnM, KnS, T, C, D) to serve as the indirect assigned function, however, they are not to be used in the bit devices (X, Y, M, S) and in the constants (K, H) to serve as the direct assigned function.



When E = 8, F = 14
D5E = D(5+8) = D13
D10F = D(10+14) = D24

Move the content of D13 to D24 when execute this command.

■ Handling of Decimal

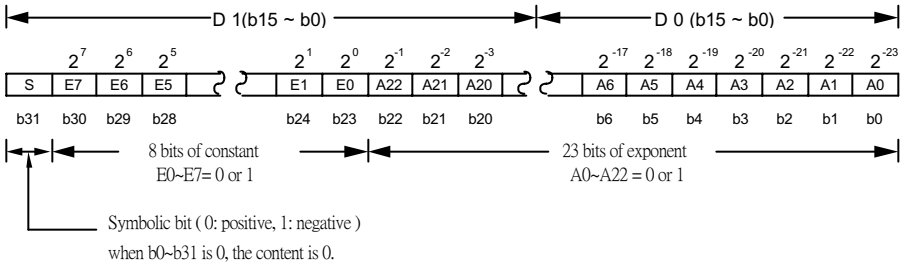
The internal operation of DVP PLC usually gets the value of BIN integer. When operating integer division, the decimal will be erased. For example: 40 ÷3=13, remainder is 1 and the decimal will be erased. But if using decimal operation, you can get decimal.

The application commands relate to decimal point are shown in the following.

API 49 (FLT)	API 110 (D ECMP)	API 111 (D EZCP)	API 118 (D EBCD)
API 119 (D EBIN)	API 120 (D EADD)	API 121 (D ESUB)	API 122 (D EMUL)
API 123 (D EDIV)	API 124 (D EXP)	API 125 (D LN)	API 126 (D LOG)
API 127 (D ESQR)	API 128 (D POW)	API 129 (INT)	API 130 (D SIN)
API 131 (D COS)	API 132 (D TAN)		

Floating point of decimal number system

- The floating point that DVP-PLC uses is binary number system, you should convert floating point of binary number system to decimal number system.
- Floating point of decimal number system is stored in the register with 2 continuous numbers. The register with small number stores constant and the register with greater number stores exponent.
  For example, using register (D1, D0) to store a floating point of decimal number system.
  Floating point of decimal number system = [constant D0] X 10 [exponent D1]
  constant D0 = 1,000 ~ 9,999
  exponent D1 = -41 ~ +35
  the left-most bit of (D1, D0) is symbol bit.
  Besides, constant 100 doesn' t exist in D0 due to 100 will be shown with $1{,}000\times10^{-1}$.
  The usage range of decimal floating point is:
  Minimum absolute value is $1{,}175\times10^{-41}$ and the maximum absolute value is $3{,}402\times10^{35}$.
- Floating point of decimal number system can be used in the following command.



The floating point of binary number system =

$$\pm\left(2^0 + A22\times2^{-1} + A21\times2^{-2} + \cdots + A0\times2^{-23}\right)$$
$$\times 2^{\left(E7\times2^7+E6\times2^6+\cdots+E0\times2^0\right)/\,2^{127}}$$

(For example) A22=1, A21=0, A20=1, A19~A0=0,
E7=1, E6~E1=0, E0=1

The floating point of binary number system =

$$\pm\left(2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + \cdots + 0\times2^{-23}\right)$$
$$\times 2^{\left(1\times2^7+0\times2^6+\cdots+1\times2^0\right)/\,2^{127}}$$
$$= \pm1.625\times2^{129}/2^{127} = \pm1.625\times2^2$$

b31 is the bit for positive/negative sign

■ Application Command Chart

API	Mnemonic Codes		Functions		Operand	Steps	
	16-bit	32-bit				16-bit	32-bit
00	CJ	-	Executes the program of specified pointer number when the jump command is on		S	3	-
01	CALL	-	Executes the subroutine program specified by the pointer (P**)		S	3	-
02	SRET	-	Executes the sequence program located at the next step to the CALL instruction		None	1	-
03	IRET	-	Indicates the termination of processing of interrupt program		None	1	-
04	EI	-	Enables the interrupt		None	1	-
05	DI	-	Disables the interrupt program until the EI instruction is executed so that interrupt signals are ignored		None	1	-
06	FEND	-	Terminates the main routine program		None	1	-
07	WDT	-	Resets the watchdog timer in a sequence program		None	1	-
08	FOR	-	Nested loop begins	When the processing of FOR to NEXT instructions is executed "n" times unconditionally, performs the processing of the next step to the NEXT instruction.	S	3	-
09	NEXT	-	Nested loop ends		None	1	-
10	CMP	DCMP	Comparison operation instruction, making numerical magnitude comparison between two pieces of data.		S1, S2, D	7	13
11	ZCP	DZCP	Zone comparison		S1, S2, S, D	9	17
12	MOV	DMOV	Transfers the data of the device specified at source to the device specified at destination		S, D	5	9
14	CML	DCML	Counter transfer		S, D	5	9
15	BMOV	-	Block move		S, D, n	7	-
16	FMOV	DFMOV	Multiple points movement		S, D, n	7	13
17	XCH	DXCH	Data exchange		D1, D2	5	9
18	BCD	DBCD	Converts BIN data of device specified into BCD		S, D	5	9
19	BIN	DBIN	Converts BCD data of device specified into BIN		S, D	5	9
20	ADD	DADD	Performs the addition of BIN data		S1, S2, D	7	13
21	SUB	DSUB	Performs the subtraction of BIN data		S1, S2, D	7	13
22	MUL	DMUL	Performs the multiplication of BIN data		S1, S2, D	7	13
23	DIV	DDIV	Performs the division BIN data		S1, S2, D	7	13
24	INC	DINC	Performs the addition of 1 to the device specified		D	3	5
25	DEC	DDEC	Performs the subtraction of 1 from the device specified		D	3	5
26	WAND	DAND	Performs the logical product of data of device specified		S1, S2, D	7	13
27	WOR	DOR	Performs the logical add of data of device specified		S1, S2, D	7	13

API	Mnemonic Codes		Functions	Operand	Steps	
	16-bit	32-bit			16-bit	32-bit
28	WXOR	DXOR	Performs the exclusive or of the data of device specified	S1, S2, D	7	13
29	NEG	DNEG	Complementary of 2	D	3	5
30	ROR	DROR	Rotate to the right	D, n	5	9
31	ROL	DROL	Rotate to the left	D, n	5	9
32	RCR	DRCR	Rotate to the right with the carrying flag attached	D, n	5	9
33	RCL	DRCL	Rotate to the left with the carrying flag attached	D, n	5	9
34	SFTR	-	Shifts the data of device specified to the right	S, D, n1, n2	9	-
35	SFTL	-	Shift the data of device specified to the left	S, D, n1, n2	9	-
40	ZRST	-	Resets a range of devices specified.	D1, D2	5	-
41	DECO	-	8 ⇨ 256 bit decode	S, D, n	7	-
42	ENCO	-	256 ⇨ 8 bit encode	S, D, n	7	-
43	SUM	DSUM	Sum of the ON bit	S, D	5	9
44	BON	DBON	Determine the ON bit	S, D, n	7	13
45	MEAN	DMEAN	Mean value	S, D, n	7	13
48*	SQR	DSQR	The square root of BIN	S, D	5	9
49*	FLT	DFLT	BIN integer→ decimal of binary number system	S, D	5	9
50	REF	-	Input/Output refresh immediately	D, n	5	-
53	HSCS	DHSCS	High speed counter comparison SET	S1, S2, D	7	13
54	HSCR	DHSCR	High speed counter comparison RESET	S1, S2, D	7	13
57	PLSY	DPLSY	Pulse Output	S1, S2, D	7	13
58	PWM	-	Pulse width modulation output	S1, S2, D	7	-
59	PLSR	DPLSR	Pulse wave output with accel/decel speeds	S1, S2, S3, D	9	17
60	IST		Manual/auto control	S, D1, D2	-	-
66	ALT	-	ON/OFF Alternate command	D	3	-
73	SEGD	-	Decode the 7-step display panel	S, D	5	-
74	SEGL	-	7-step display scan output	S, D, n	7	-
78	FROM	DFROM	Read special module CR data	m1, m2, D, n	9	17
79	TO	DTO	Special module CR data write in	m1, m2, S, n	9	17
80	RS	-	Data communication is performed according to the data in the specified communication area	S, m, D, n	9	-
82	ASCI	-	Converts the specified hexadecimal value into the ASCII code	S, D, n	7	-
83	HEX	-	Converts the specified ASCII code into the hexadecimal value	S, D, n	7	-

API	Mnemonic Codes		Functions	Operand	Steps	
	16-bit	32-bit			16-bit	32-bit
87	ABS	DABS	Absolute value	D	3	5
88	PID	-	PID calculation	S1, S2, S3, D	9	-
100	MODRD	-	Modbus data read command	S1, S2, n	7	-
101	MODWR	-	Modbus data write command	S1, S2, n	7	-
102	FWD	-	Delta inverter VFD-A series forward command	S1, S2, n	7	-
103	REV	-	Delta inverter VFD-A series reverse command	S1, S2, n	7	-
104	STOP	-	Delta inverter VFD-A series stop command	S1, S2, n	7	-
105	RDST	-	Delta inverter VFD-A series status data read command	S, n	5	-
106	RSTEF	-	Delta inverter VFD-A series external fault clearing command	S, n	5	-
110*	ECMP	DECMP	Comparison of decimal of binary system	S1, S2, D	7	13
111*	EZCP	DEZCP	Comparison of the area of decimal of binary system	S1, S2, S, D	9	17
118*	EBCD	DEBCD	Decimal of binary number→ decimal of decimal system	S, D	5	9
119*	EBIN	DEBIN	Decimal of decimal system→ decimal of binary system	S, D	5	9
120*	EADD	DEADD	Addition of decimal of binary system	S1, S2, D	7	13
121*	ESUB	DESUB	Subtraction of decimal of binary system	S1, S2, D	7	13
122*	EMUL	DEMUL	Multiplication of decimal of binary system	S1, S2, D	7	13
123*	EDIV	DEDIV	Division of decimal of binary system	S1, S2, D	7	13
124*	EXP	DEXPP	Exponent obtain command	S, D	5	9
125*	LN	DLN	Natural logarithm obtain command	S, D	5	9
126*	LOG	DLOG	Logarithm obtain command	S1, S2, D	7	13
127	ESQR	DESQR	Square root of decimal of binary system	S, D	5	9
128*	POW	DPOW	Floating value command	S1, S2, D	7	13
129*	INT	DINT	Decimal of binary system → BIN integer	S, D	5	9
130*	SIN	DSIN	SIN operation of decimal of binary system	S, D	5	9
131*	COS	DCOS	COS operation of decimal of binary system	S, D	5	9
132*	TAN	DTAN	TAN operation of decimal of binary system	S, D	5	9
147	SWAP	DSWAP	Swap upper and lower 8-bit	S	3	5
150	MODRW	-	MODBUS data read/write	S1, S2, S3, S4, n	11	-
224	LD=	DLD=	(S1) = (S2)	S1, S2	5	9
225	LD>	DLD>	(S1) > (S2)	S1, S2	5	9

API	Mnemonic Codes		Functions	Operand	Steps	
	16-bit	32-bit			16-bit	32-bit
226	LD<	DLD<	(S1) < (S2)	S1, S2	5	9
228	LD<>	DLD<>	(S1) ≠ (S2)	S1, S2	5	9
229	LD<=	DLD<=	(S1) ≦ (S2)	S1, S2	5	9
230	LD>=	DLD>=	(S1) ≧ (S2)	S1, S2	5	9
232	AND=	DAND=	(S1) = (S2)	S1, S2	5	9
233	AND>	DAND>	(S1) > (S2)	S1, S2	5	9
234	AND<	DAND<	(S1) < (S2)	S1, S2	5	9
236	AND<>	DAND<>	(S1) ≠ (S2)	S1, S2	5	9
237	AND<=	DAND<=	(S1) ≦ (S2)	S1, S2	5	9
238	AND>=	DAND>=	(S1) ≧ (S2)	S1, S2	5	9
240	OR=	DOR=	(S1) = (S2)	S1, S2	5	9
241	OR>	DOR>	(S1) > (S2)	S1, S2	5	9
242	OR<	DOR<	(S1) < (S2)	S1, S2	5	9
244	OR<>	DOR<>	(S1) ≠ (S2)	S1, S2	5	9
245	OR<=	DOR<=	(S1) ≦ (S2)	S1, S2	5	9
246	OR>=	DOR>=	(S1) ≧ (S2)	S1, S2	5	9

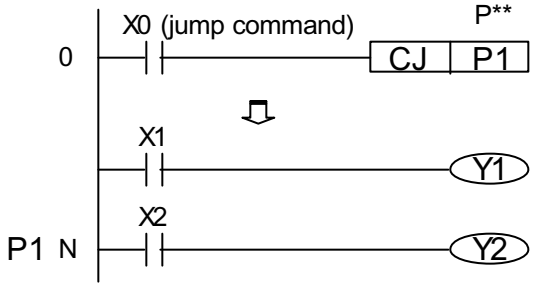
API	Code	Function	Operand	Steps
89	PLS	Rising-edge output	Footnote 1	3
90	LDP	Rising-edge detection operation		3
91	LDF	Falling-edge detection operation		3
92	ANDP	Series connection command for the rising-edge detection operation		3
93	ANDF	Series connection command for the falling-edge detection operation		3
94	ORP	Parallel connection command for the rising-edge detection operation		3
95	ORF	Parallel connection command for the falling-edge detection operation		3
96	TMR	16-bit Timer		4
97	CNT	16-bit Counter		4
97	DCNT	32-bit Counter		6
98	INV	Inverting operation		1
99	PLF	Falling-edge output		3

Footnote 1: API 89~99 belong to the basic commands, and refer to Ch6 Basic Commands for the operand and usage explanations.

Footnote 2: API number with the " *" mark indicates the API commands only support the version V5.4 or above of ES / EX / SS series models.

00 CJ 【P**】 P 00 to P 63 Conditional jump

- Executes the program of specified pointer when the jump command is on.
- Executes the program of the next step when the jump command is off.
- **Program Example**



◎ When X0=ON, program address jumps from 0 to N (label P1) continuing the program, skipping the addresses in between.

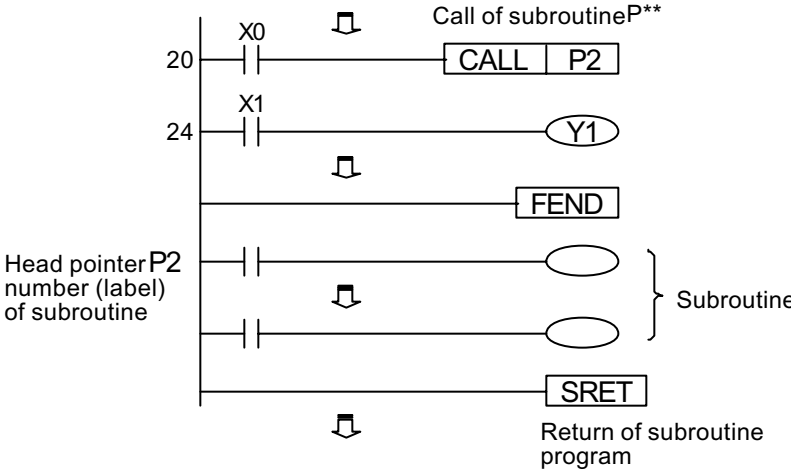
◎ If there is a TMR (timer) in the middle of the address, the timer will stop counting. When X0=OFF, the program will continue from address 0, the CJ command will not be activated, and the timer will continue counting.

01 CALL 【P**】 P 0 to P 63 Call subroutine

- Up to five levels of nesting of the CALL instruction are allowed.
- Executes the subroutine program specified by the pointer (P**)

02 SRET Termination of subroutine program

- Executes the sequence program located at the next step to the CALL instruction when the SRET instruction is executed.
- Indicates the end of subroutine program.
- **Program Example**



◎ When X0 = ON then starts CALL command, jump to P2 and run subroutine. When run SRET command, it will jump back to address 24.

◎ Program continues in the subroutine after the FEND command.

◎ When using CALL command, the numbers of P0 to P63 cannot be duplicated with CJ command.

03 IRET Termination of processing of interrupt program

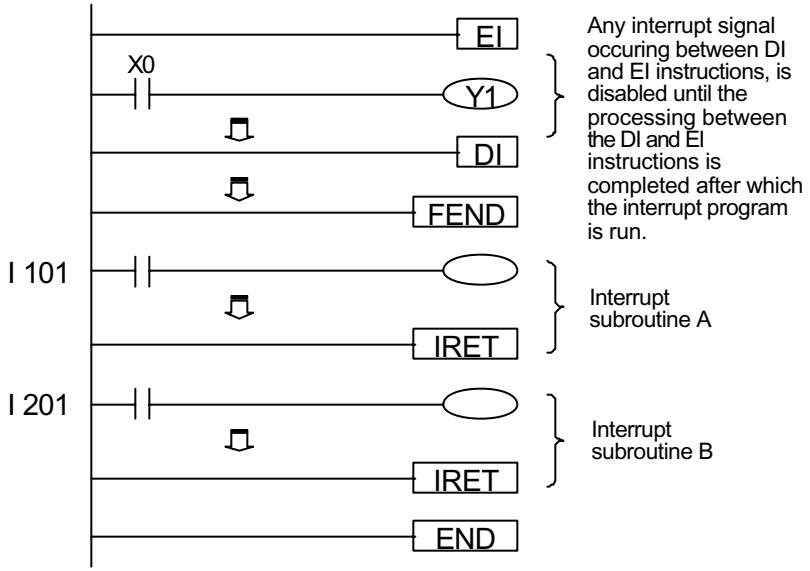
- Indicates the termination of interrupt program.

04 EI Enables interrupt

05 DI Disables interrupt

- During the PLC operation, the program scans the commands between EI and DI, if X001 and X002 are ON, the Interrupt Service Subroutine I 001 and I 201 will be activated when IRET is reached, the main program will resume.

- When interrupting a special auxiliary relay M1050 to M1053, the same interrupting request will not be activated.
- Interrupting cursor (I001 to I201) must be used after the FEND command.
- **Program Example**

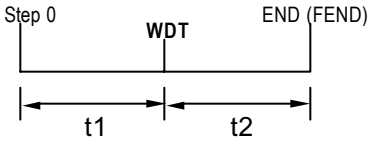


06 FEND Terminates the main routine program

- It has the same function as END command during PLC operation.
- CALL must follow right after FEND command. Interrupt commands also have to follow after FEND command.
- If using several FEND commands, please place the subroutine and interrupt service programs after the last FEND command.

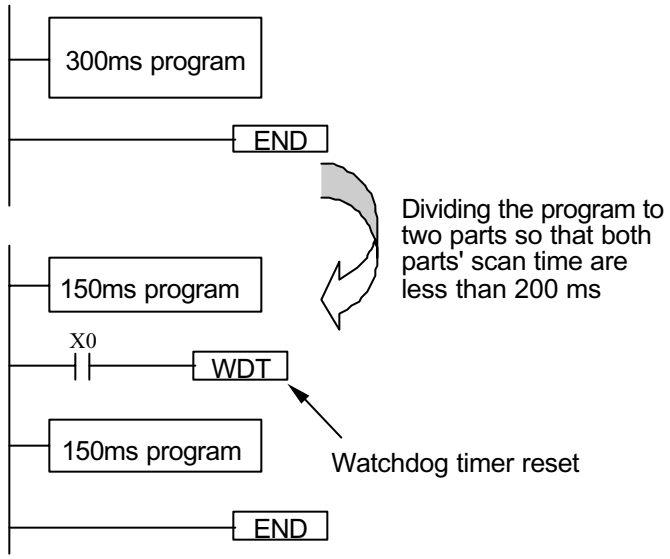
07 WDT Resets the watchdog timer

- Resets the watchdog timer.
- Used when the period of time from step 0 to END (FEND) in the sequence program exceeds the set value of the watchdog timer.
- Set the value of the watchdog timer so "t1" is from step 0 to WDT instruction and "t2" is from the WDT to END (FEND) instruction. (See the diagram below)



- The WDT instruction can be used two or more times during one scan. However, care should be exercised because, during an error, the outputs cannot be turned off immediately.
- If the PLC scan time is more than 200ms, the ERROR LED will flash. The user will have to turn the PLC OFF and then back ON to clear the fault.
- If the program scan time is over 200ms, users can divide the program into 2 parts. Insert the Watchdog Timer in between, so both programs' scanning time will be less than 200ms.
- The WDT time can be changed by the setting value of D1000 (default is 200ms).
- When the ERROR LED of PLC is steady, M1008 and D1008 can be watched.

■ **Program Example**



◎ Program used when the setting of watchdog timer is 200ms and the period of time from 0 to **END** (**FEND**) instruction is 300ms.

08 **FOR** (S) Nested loop begins

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎

09 **NEXT** Nested loop ends

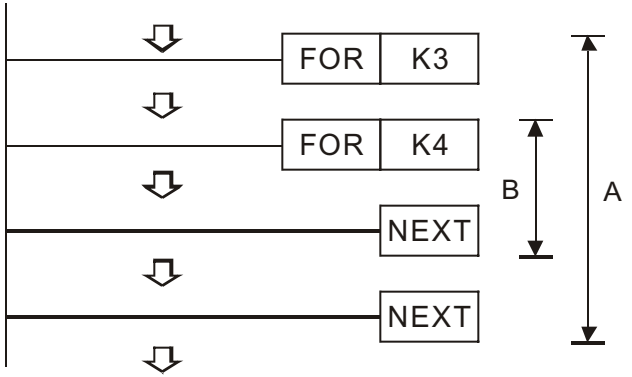
■ **FOR** and **NEXT** instructions are used when "n" nested loops are needed.

■ "n" may be specified as 1 to 32767.

■ When it is not desired to execute the **FOR** to **NEXT** instructions, use the **CJ** command.

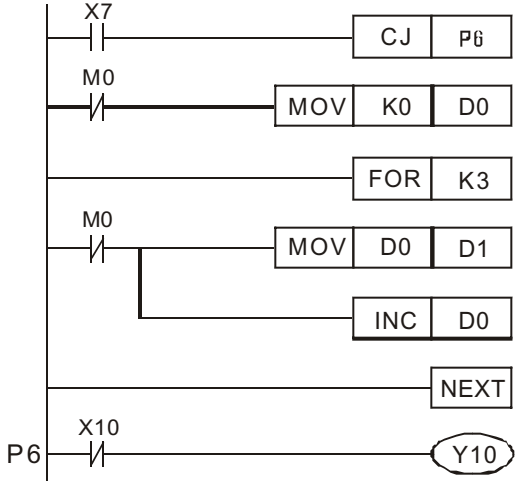
■ Up to four levels of nested **FOR** loops are allowed.

■ For example, loop A operates 3 times but within this loop there is nested loop, B. For every completed cycle of loop A, the loop B will be completed executed 4 times. Therefore, the numbers of loop B operation will be 3 x 4=12 times (A x B).



■ If there are too many loops, the processing time will be executed.

■ **Program Example**

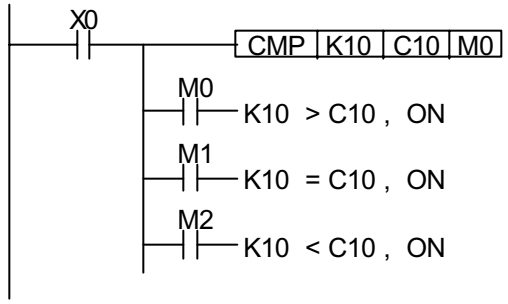


◎ Program which executes the **FOR** to **NEXT** instructions when X7 is off. It does not execute the **FOR** to **NEXT** instructions when X7 is on.

D 10 CMP (S1) (S2) (D) Comparison Output

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)		◎	◎	◎											

- This function compares the two values that are considered **BIN** values. In the case of comparing hexadecimal. If a numeric value (8 to F) having 1 at the highest bit (B15 in a 16-bit instruction or B31 in a 32-bit instruction) is specified, the comparison will regard the value as the negative of the **BIN** value.
- The comparison instructions verifies 3 items (=, >, <) between the data (S1) and (S2). See the example below.

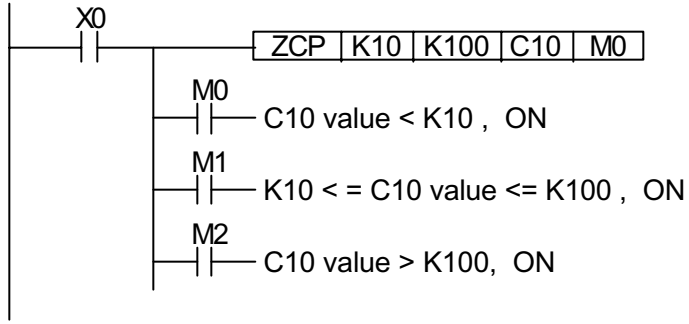


◎ If (D) is set to M0, then M0, M1, M2 will work as the program example as above.

D 11 ZCP (S1) (S2) (S) (D) Zone comparison

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)		◎	◎	◎											

- Compare the value of Operand (S) with its limits (S1) and (S2). If (D) is set to M0, then M0, M1, and M2 will work as follows.



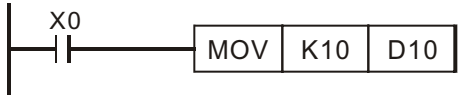
D 12 MOV (S) (D) Data movement

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

- Transfers the data of device (S) to the device (D).
- If the calculation result is a 32-bit output, (i.e. the application **MUL**) and the data of a 32-bit high-speed counter, users will have to use **DMOV**

command.

■ **Program Example**



◎ When X0 is Off, the content of D10 remain unchanged. If X0 turns On, the data of K10 is moved to D10 data register.

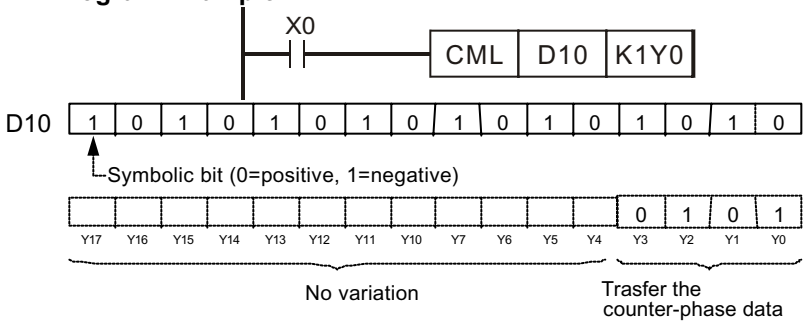
D 14 CML (S) (D) Counter transfer

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Counter phase the contents of (S) (0→1, 1→0) and have the contents transferred to (D). If the content is of Constant K, this Constant K will be converted to the BIN value automatically.

■ This command can be used during the counter-phase output.

■ **Program Example**



■ When X0 is ON, contents of D10, b0~b3, will be counter transferred to K1Y0.

15 BMOV (S) (D) (n) Block move

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)											◎	◎	◎		
(D)											◎	◎	◎		
(n)					◎	◎									

■ Move the contents of the n register, with this n register obtained from counting the registers within the (S)-assigned numbers, to the n register within the (D)-assigned number.

■ If the (n)-assigned points exceed the usage range of this device, only those that are within the effective range will be moved.

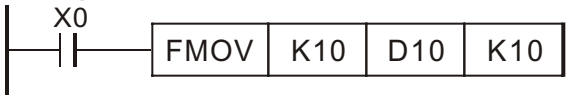
■ **Program Example**

◎ When X0 is ON, move the contents of the three registers D0~D2, to their corresponding registers D10~D12.

D 16 FMOV (S) (D) (n) Multiple points movement

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎					◎	◎	◎		
(D)											◎	◎	◎		
(n)					◎	◎									

- Move the contents of (S) to the n register, with this n register obtained from counting the registers within the (D) -assigned numbers.
- If the n-assigned points exceed the usage range of this device, only those that are within the effective range will be moved.
- **Program Example**

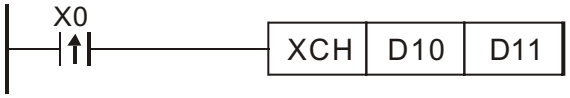


◎ When X0 is ON, move Constant K10 to the ten registers (D10~D19) starting from D10.

D **17** **XCH** (D1) (D2) Data exchange

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(D1)								◎	◎	◎	◎	◎	◎	◎	◎
(D2)								◎	◎	◎	◎	◎	◎	◎	◎

- Exchange the contents of (D1) and (D2) with each other.
- Generally, it is advised to use the pulse wave contact to drive this command, otherwise, this command will keep going on and on.
- **Program Example**

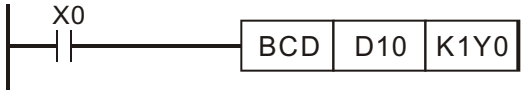


When X0 goes from OFF→ON, contents of D10 and D11 will exchange with each other.

D **18** **BCD** (S) (D) Converts BIN data into BCD

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)							◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

- Converts **BIN** data (0~9999) of the device (S) into **BCD** and transfers the result to the device (D).
- If the data of source (S) is outside the range of 0 to 9999, there is an operation error and the error flag (M1067 and M1068) turns on.
- **Program Example**

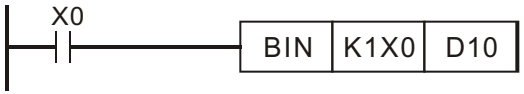


◎ When X0 = ON, the D10 **BIN** value will be converted to **BCD**, and the result saved in K1Y0.

D **19** **BIN** (S) (D) Converts BCD data into BIN

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)							◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

- Converts **BCD** data (0 to 9999) of device (S) into **BIN** and transfer the result to the device (D).
- If each digit of source (S) is outside the range of 0 to 9, there is an operation error and the error flag (M1067 and M1068) turns on.
- **Program Example**



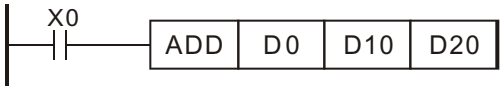
◎ Program which converts the **BCD** data of X0 to X03 into **BIN** and stores the result into D10 when X10 turns on.

D **20** **ADD** (S1) (S2) (D) Performs the addition of BIN data

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the addition on **BIN** data (S1) and the **BIN** data (S2), and stores the addition result into the device (D).

■ **Program Example**



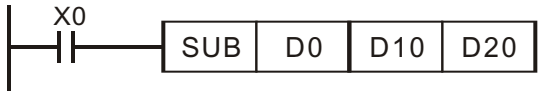
◎ Program which adds the contents of D0 to the content of D10 and outputs the result to D20 when X0 turns on.

D **21** **SUB** (S1) (S2) (D) Performs the subtraction of BIN data

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the subtraction of **BIN** data (S1) and the **BIN** data (S2), and stores the subtraction result into the device (D).

■ **Program Example**



◎ Program which outputs the difference between the content of D0 and the content of D10 to D20 when X0 turns on.

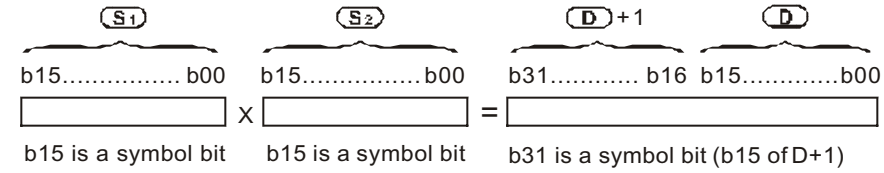
D **22** **MUL** (S1) (S2) (D) Performs the multiplication of BIN data

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the Multiplication of **BIN** data (S1) and the **BIN** data (S2), and stores the multiplication result into the device (D).

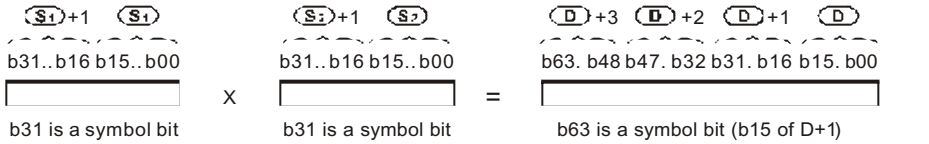
■ The judgment of whether the data of (S1) and (S2) are positive or negative is made at the highest bit (b15) and for (D) at (b31).

■ 16-bit operation



b15=0,S1is a positive value b15=0,S2is a positive value b31=0,Sdis a positive value
b15=1,S1is a negative value b15=1,S2is a negative value b31=1,Sdis a negative value

■ 32-bit operation



b31=0,S(S+1) are positive value b31=0,S(S2+1) are positive value b63=0, D1(D1+1) (D1+2) (D1+3) are positive value
b31=1,S(S+1) are negative value b31=1,S(S2+1) are negative value b63=1, D1(D1+1) (D1+2) (D1+3) are negative value

■ **Program Example** (16-bit)

◎ The above program stores the multiplication result of D0 and D10 in BIN to D20 and D21 when X0 turns on.

◎ The upper 16-bit will be saved in D21 and the lower 16-bit will be saved in the D20.

◎ The polarity of the result is indicated by the OFF/ON of the most significant bit. OFF indicates the value of positive 0 and ON indicates the value of negative 1.

■ **Program Example** (32-bit)

◎ The above program stores the multiplication result of the 32-bit **BIN** data of D0 and D1 and the 32-bit BIN data of D10 and D11 to the 64-bit **BIN** data of D20 to D23.

◎ The polarity of the result is indicated by the OFF/ON of the most significant bit. OFF indicates the value of positive 0 and ON indicates the value of negative 1.

D **23** **DIV** (S1) (S2) (D) Performs the division BIN data

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

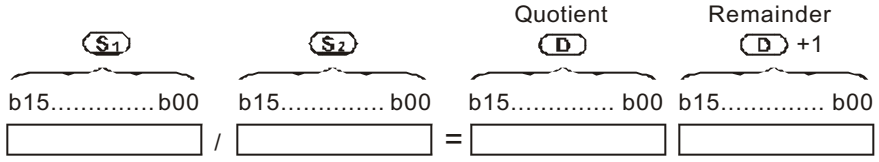
■ Performs the division of **BIN** data (S1) and the **BIN** data (S2), and stores the result into the device (D).

■ In regards to the operation result, the quotient and remainder are stored using 32 bits for a word device, and only the quotient is stored using 16 bits for a bit device.

■ The judgment of whether the data of (S1) and (S2) are positive or negative is made at the highest bit (b15) and for (D), at (b31).

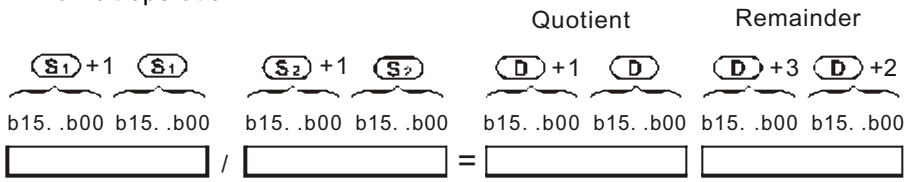
■ When using the DDIV command, the special register (additional special device D) cannot serve as indirect assigned function.

■ 16 bit operation



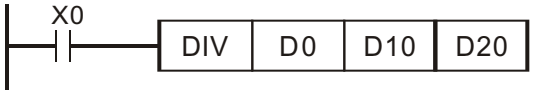
Quotient : Stored to the lower 16 bits.
Remainder : Stored to the upper 16 bits. (Stored only in the case of a word device.

■ 32 bit operation



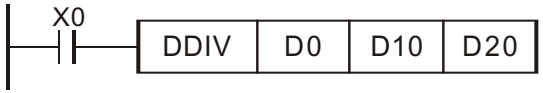
Quotient : Stored to the lower 32 bits.
Remainder : Stored to the upper 32 bits. (Stored only in the case of a word device.

■ **Program Example** (16 bit)



◎ Program which outputs the quotient and remainder, obtained by dividing the data D0 by D10, to D20 and D21 when X0 turns on.

■ **Program Example** (32-bit)



■ When X0 = ON, performs the division of **BIN** data (D1, D0) and **BIN** data (D11, D10), the quotient stored (D21, D20) and the remainder stored in (D23, D22).

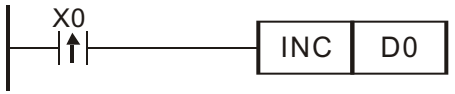
D **24** **INC** (D) Performs the addition of 1

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the addition of 1 to the device (D).

■ Generally, it is advised to use the pulse wave contact to drive this command, otherwise, this command will keep going on and on.

■ **Program Example**



◎ When X0 = ON, the content of D0 will perform the addition of 1.

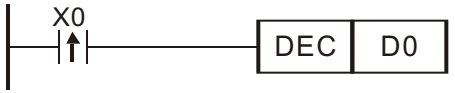
D **25** **DEC** (D) Performs the subtraction of 1

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the subtraction of 1 from the device (D).

■ Generally, it is advised to use the pulse wave contact to drive this command, otherwise, this command will keep going on and on.

■ **Program Example**



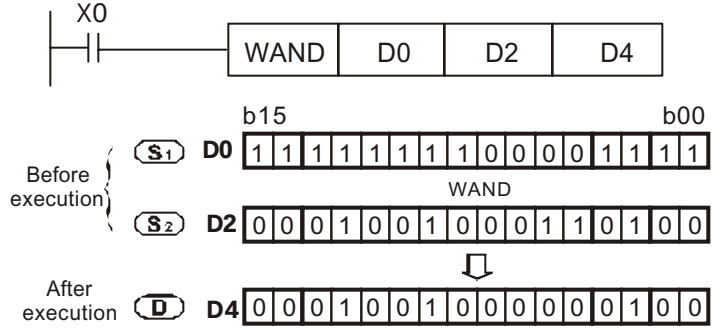
◎ When X0 = ON, Performs the subtraction of 1 from the device (D).

D **26** **WAND** (S1) (S2) (D) Performs the logical product

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the logical product of the 16-bit data of device (S1) and the 16-bit data of device (S2) per bit, and stores the result into the device (D).

■ **Program Example** (16-bit)

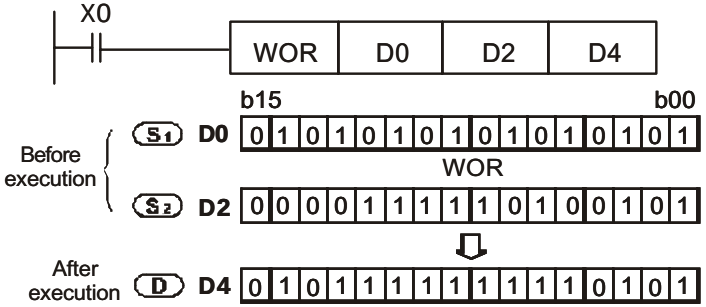


D 27 WOR S1 S2 D Logic or (OR) operation

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
S2					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
D								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the logical add of the 16-bit data of device S1 and the 16-bit data of device S2 per bit, and stores the result into the device D.

■ **Program Example** (16-bit)

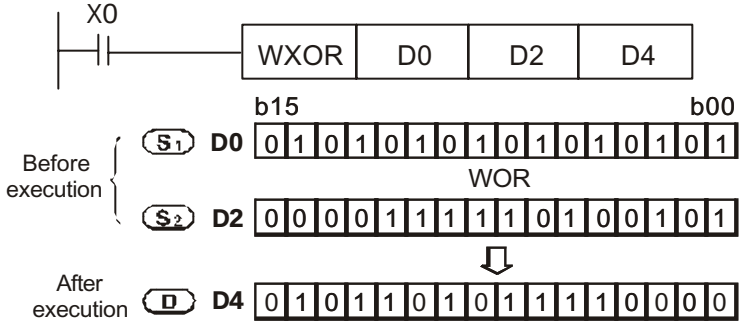


D 28 WXOR S1 S2 D Performs the exclusive logical add

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
S2					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
D								◎	◎	◎	◎	◎	◎	◎	◎

■ Performs the exclusive OR of the 16-bit data of device S1 and the 16-bit data of device S2 per bit, and stores the result into the device D.

■ **Program Example** (16-bit)



D 29 NEG D Complementary of 2

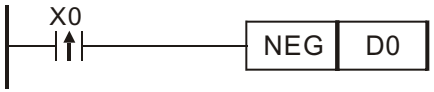
Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
D								◎	◎	◎	◎	◎	◎	◎	◎

■ Once the command is executed, the assigned device, D, will be served as the complementary of 2.

■ It is generally advised to use the pulse wave contact to drive this

command, otherwise, it will keep going on and on.

■ **Program Example**



◎ When X goes from OFF→ON, every bit of the D0 contents will be countered (0→1, 1→0) and be added with 1, and will then be saved in the original register, D0.

◎ This command could convert the negative BIN value to the positive number, and that is, to get its absolute value.

D 30 ROR (D) (n) Rotate to the right

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	K n X	K n Y	K n M	K n S	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

■ Essential condition: n≦16 (16-bit command), n≦32 (32-bit command).

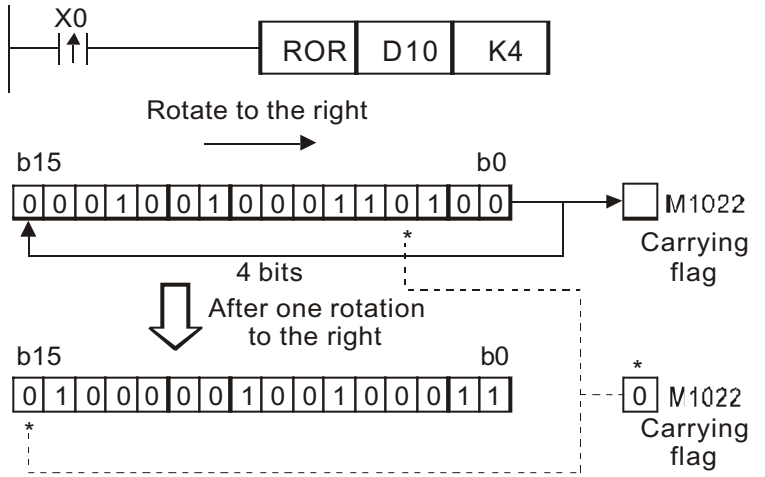
■ The (D) ROR command is used to assign the (32) 16-bit data to conduct the rotate-to-the-right command. The LSB that is rotating to the right simultaneously will be moved to the carrying flag (CY) M1022.

■ It is generally advised to use the pulse wave contact to drive this command, otherwise, it will keep going on and on.

■ When M, Y, S are assigned to serve as the bit operand, only K4 (16-bit) and K8 (32-bit) are effective, e.g. K4M0, K8Y0.

■ Command Motion Explanation

◎ When X0 goes from OFF→ON, the 16 bit data of D10 will rotate 4 bits to the right, as shown in the diagram, and b3 that located at D10 originally will then be moved to the carrying flag (CY) M1022.



D 31 ROL (D) (n) Rotate to the left

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	K n X	K n Y	K n M	K n S	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

■ Essential condition: n≦16 (16-bit command), n≦32 (32-bit command).

■ The (D) ROL command is used to assign the (32) 16-bit data to conduct the rotate-to-the-left command. The MSB that is rotating to the left simultaneously will be moved to the carrying flag (CY) M1022.

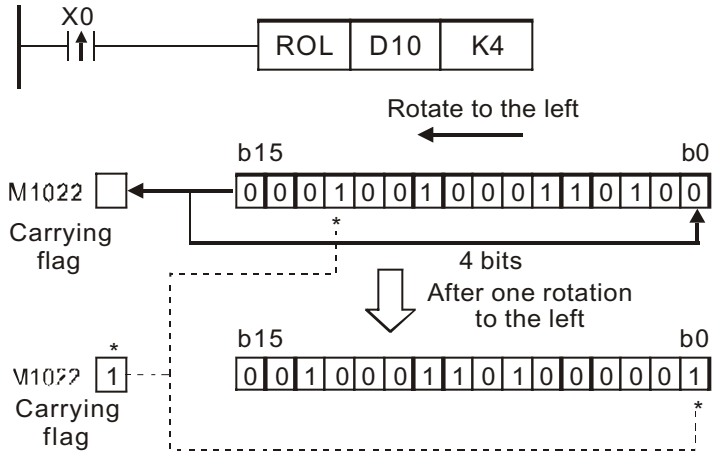
■ It is generally advised to use the pulse wave contact to drive this command, otherwise, it will keep going on and on.

■ When M, Y, S are assigned to serve as the bit operand, only K4 (16-bit) and K8 (32-bit) are effective, e.g. K4M0, K8Y0.

■ Command Motion Explanation

◎ When X0 goes from OFF→ON, the 16 bit data of D10 will rotate 4 bits to the left, as shown in the diagram, and b12 that located at D10

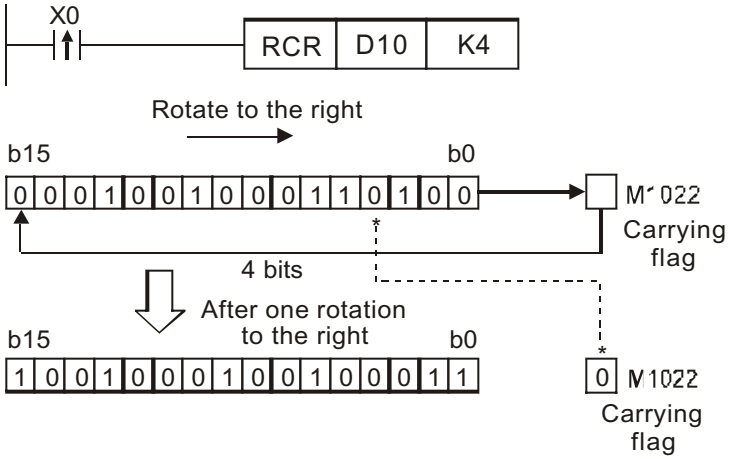
originally will then be moved to the carrying flag (CY) M1022.



D **32** **RCR** (D) (n) Rotate to the right with the carrying flag attached

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	K n X	K n Y	K n M	K n S	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

- Essential condition: n≦16 (16-bit command), n≦32 (32-bit command).
- The (D) RCR command is used to assign the (32) 16-bit data with the attached carrying flag (M1022) to conduct the rotate-to-the-right command.
- It is generally advised to use the pulse wave contact to drive this command, otherwise, it will keep going on and on.
- When M, Y, S are assigned to serve as the bit operand, only K4 (16-bit) and K8 (32-bit) are effective, e.g. K4M0, K8Y0.
- Command Motion Explanation
- When X0 goes from OFF→ON, the 16 bit data of D10, along with the attached carrying flag (M1022), will rotate 4 bits to the right, as shown in the diagram, and b3 that located at D10 originally will then be moved to the carrying flag M1022, and that the original contents of the carrying flag M1022 will be moved to the bit of b12.



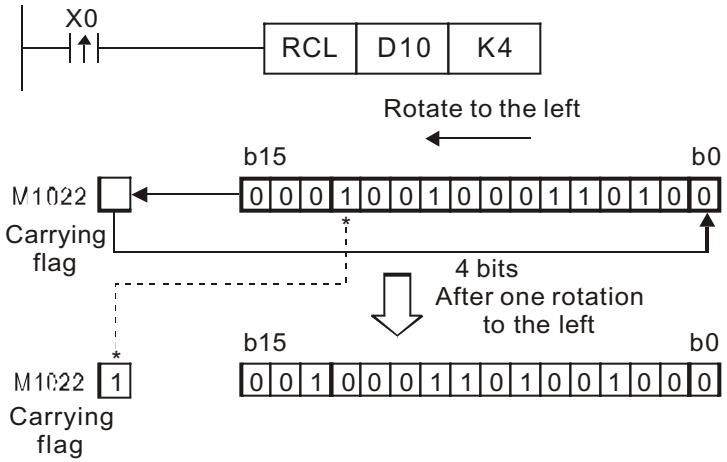
D **33** **RCL** (D) (n) Rotate to the left with the carrying flag attached

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	K n X	K n Y	K n M	K n S	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

- Essential condition: n≦16 (16-bit command), n≦32 (32-bit command).
- The (D) RCL command is used to assign the (32) 16-bit data with the attached carrying flag (M1022) to conduct the rotate-to-the-left command.
- It is generally advised to use the pulse wave contact to drive this command, otherwise, it will keep going on and on.

- When M, Y, S are assigned to serve as the bit operand, only K4 (16-bit) and K8 (32-bit) are effective, e.g. K4M0, K8Y0.
- Command Motion Explanation

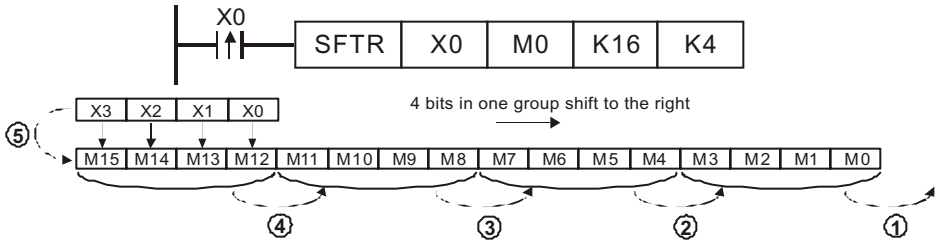
◎ When X0 goes from OFF→ON, the 16 bit data of D10, along with the attached carrying flag (M1022), will rotate 4 bits to the left, as shown in the diagram, and b12 that located at D10 originally will then be moved to the carrying flag M1022, and that the original contents of the carrying flag M1022 will be moved to the bit of b3.



34 SFTR (S) (D) (n1) (n2) Shifts the data of device specified to the right

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)	◎	◎	◎	◎											
(D)		◎	◎	◎											
(n1)					◎	◎									
(n2)					◎	◎									

- Requirement: n2≦n1≦512. Shifts (n2) data bits of device (S) to the right by (n1) bits. (n2) bits, which begin with (D), are shifted to the right.
- It is generally advised to use the pulse wave contact to drive this command, otherwise, it will keep going on and on.
- **Program Example**



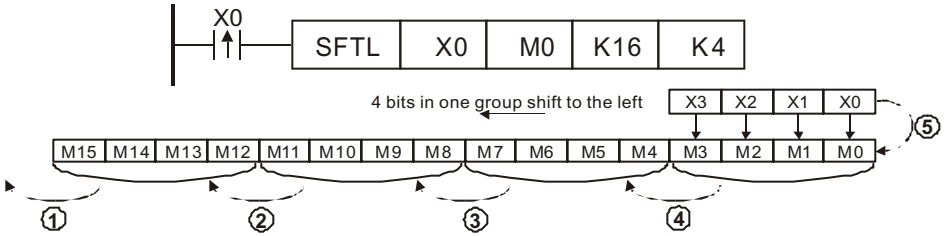
◎ Please Notice: If use the pulse wave contact to drive this command, X0 will shift (n2) bits to the right when X0 is in the rising-edge. If use the normal On contact to drive this command, the bit shifting operation will occur every time during every scan.

35 SFTL (S) (D) (n1) (n2) Shifts the data of device specified to the left

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)	◎	◎	◎	◎											
(D)		◎	◎	◎											
(n1)					◎	◎									
(n2)					◎	◎									

- Requirement: n2≦n1≦512
- Shifts (n2) data bits of device (S) to the left by (n1) bits. (n2) bits, which begin with (D), are shifted to the left.

■ **Program Example**



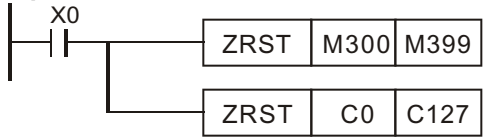
◎ Please Notice: If use the pulse wave contact to drive this command, X0 will shift (n2) bits to the right when X0 is in the rising-edge. If use the normal On contact to drive this command, the bit shifting operation will occur every time during every scan.

40 ZRST (D1) (D2) Resets a range of device specified.

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(D1)		◎	◎	◎							◎	◎	◎		
(D2)		◎	◎	◎							◎	◎	◎		

■ Requirement: (D1) ≦ (D2), and must be in the same category.

■ **Program Example**



◎ When X0 = O, M300 to M399 will be OFF. The value of C200 to C210 counters will be set to 0.

41 DECO (S) (D) (n) 8 ⇨ 256 bit decode

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)	◎	◎	◎	◎	◎	◎					◎	◎	◎	◎	◎
(m)		◎	◎	◎							◎	◎	◎	◎	◎
(D)					◎	◎									

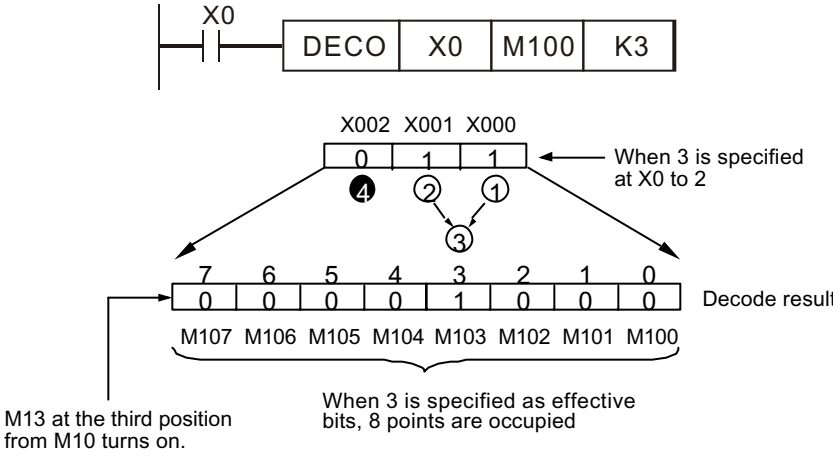
■ Decodes the lower "n" bits of device (S) and stores the result of decode data to 2^n bits which begin with the device (n).

■ For "n", 1 to 8 can be specified.

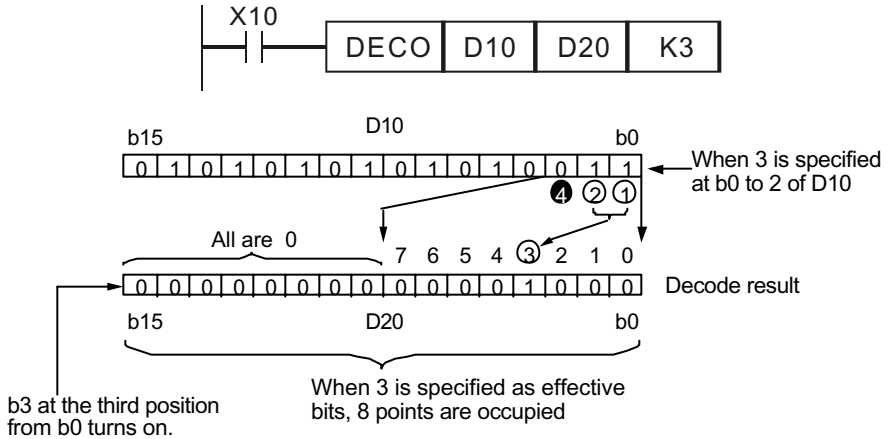
■ When n=0 or n>8 the calculation error will occur.

■ A bit device is treated as one bit and a word device as 16 bits.

■ (D) is used in the case of a bit device, 0<n≦8



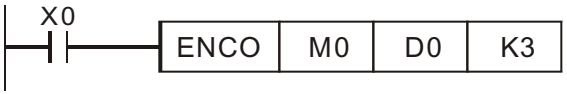
■ (D) is used in the case of a word device, 0<n≦4

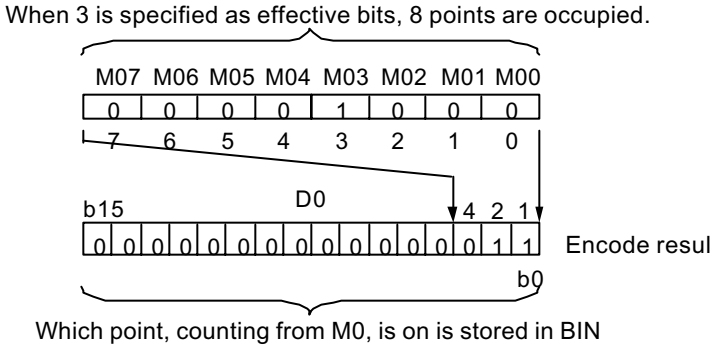


42 ENCO (S) (D) (n) 256 ⇨ 8 bit encode

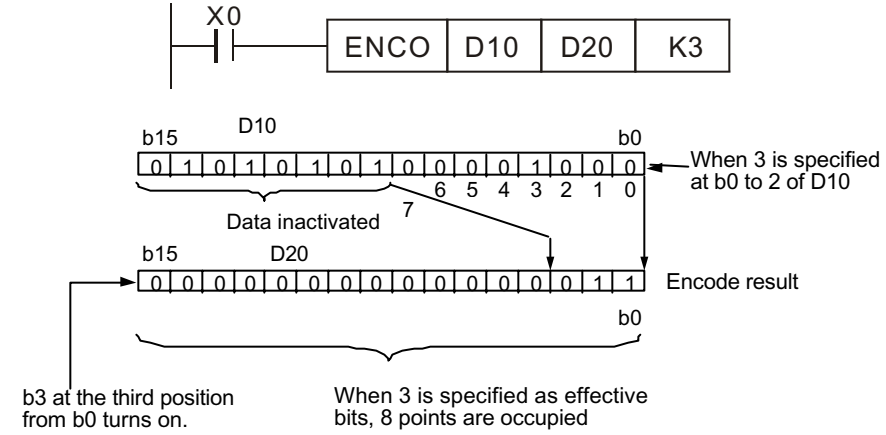
Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)	◎	◎	◎	◎							◎	◎	◎	◎	◎
(D)											◎	◎	◎	◎	◎
(n)					◎	◎									

- Encodes the data of 2^n bits, which begin with (S), and stores the result to (D).
- For (n), 1 to 8 can be specified.
- When n=0 or n>8 the calculation error will occur.
- When multiple bits are 1, processing is performed for the last bit position.
- (S) is used in the case of a bit device, 0<n≦8





- (S) is used in the case of word device, 0<n≦4

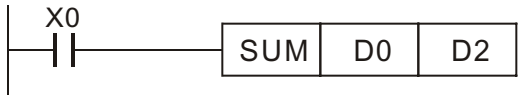


D 43 SUM (S) (D) Sum of the ON bit

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ Count all the bits with "1" as its content within (S), and have this counted number saved in (D).

■ **Program Example**



◎ When X=ON, all the bits that with "1" as its content within D0 will be counted, and have this counted number saved in D2.

◎ If the contents of these 16 bits are "0", the "Zero" flag signal M8020=ON.

◎ When using the 32 bit commands, DSUM or DSUMP, in the above-mentioned program, the total number of the 32 bits within D1 and D0 that possess "1" as its content will be counted and saved in D2, and D3=0.

D 44 BON (S) (D) (n) Determine the ON bit

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)		◎	◎	◎											
(n)					◎	◎									

■ Essential condition: when it is of the 16 bit commands, n=0~15, and when it is of the 32 bit commands, n=0~13.

■ With the content of (S), if the content of the (n) bit is "1", the (D)-assigned bit device will be set as "ON".

■ **Program Example**

◎ When X0=ON and that the 15th bit of D0 is "1", M0=ON.

◎ Once X0 is switched to OFF, M0 will stay at its previous ON/OFF status.

D 45 MEAN (S) (D) (n) Mean value

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)							◎	◎	◎	◎	◎	◎	◎		
(D)								◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

■ Add the contents of the (S)-assigned (n) registers, and have the sum ddivided by (n) to obtain a mean value.To save this mean value in the designated (D).

■ **Program Example**

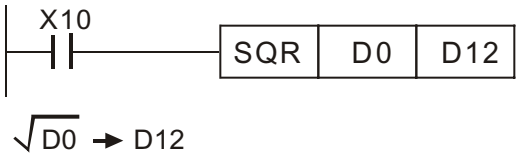
◎ When X0=ON, add up the contents of the 4 registers starting from D10 (assigned by this command), divide the sum by 4 to obtain the mean value.To save the value in the assigned D20.

If there is remainder in this calculation, discard the remainder. If the assigned device number exceeds the normal usable range, only those that within the range could be processed.

D 48 SQR (S) (D) The square root of BIN

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎							◎		
(D)													◎		

- After getting the sqare root of the content of device that (S) designates, save it into the device that (D) designates.
- The value that (S) can designate is positive number. If (S) designates negative number, PLC will regard it as command operation error, M8067=On and this command won' t be executed.
- (D) just gets the integer of the result and the decimal will be discarded. If there is decimal discarded, carrying flag M1021=On.
- If the operation result of (D) is 0, zero flag M1020=On.
- **Program Example**

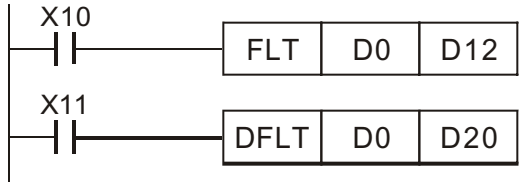


◎ When X10=On, the content of D0 will be saved in D12 after the operation of square root.

D 49 **FLT** (S) (D) BIN integer→ decimal of binary number system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

- (S): change source device. (D): the device that used to save change result.
- When M1081=Off, convert BIN integer to decimal of binary system.
- When M1081=On, convert decimal of binary system to BIN integer (discard the value of decimal)
- Constant K and H will be converted to decimal of binary system in decimal operation so they don' t need to use this command to convert.
- **Program Example**

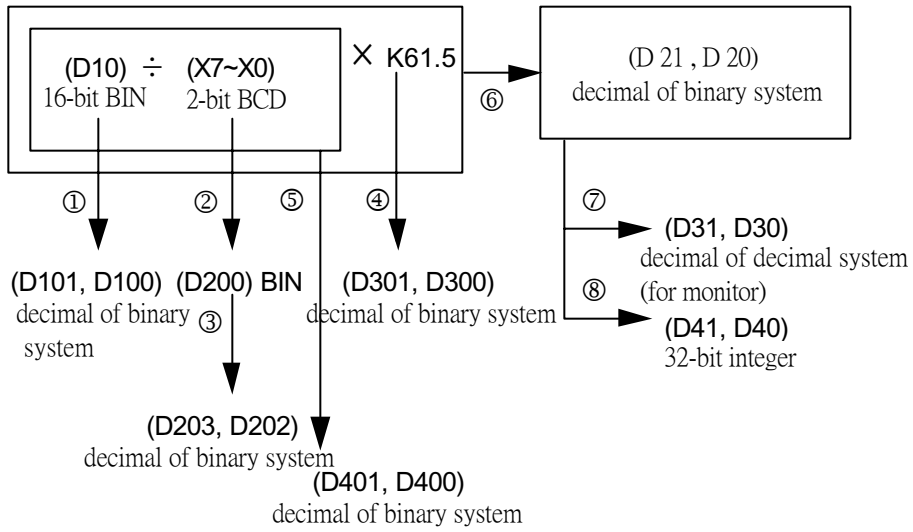


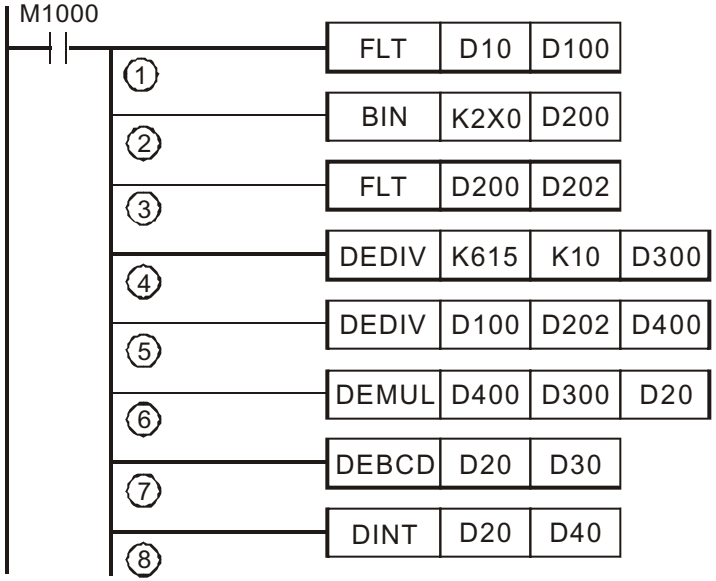
◎ When X10=On, D0(BIN integer) will be converted to D13, D12(decimal of binary system)

◎ When X11=On, D1 and D0(BIN integer) will be converted to D21, D20(decimal of binary system)

- **Program Example**

Using application command to complete the following operation.





- Covert D10(BIN integer) to D101, D100(decimal of binary system).
- Covert the value of X7~X0(BIN integer) to D200(BIN integer).
- Covert D200(BIN integer) to D203, D202 (decimal of binary system).
- Save the result of K615 ÷ K10 to D301, D300 (decimal of binary system)
- Save the result of decimal of binary system of (D101, D100) ÷ (D203, D202) to D401, D400 (decimal of binary system)
- Save the result of decimal of binary system of (D401, D400) × (D301, D300) to D21, D20 (decimal of binary system)
- Covert decimal of binary system (D21, D20) to decimal of decimal system (D31, D30)
- Covert decimal of binary system (D21, D20) to BIN integer D41, D40
- Please refert to page 7-4 Handling of Decimal for detail.

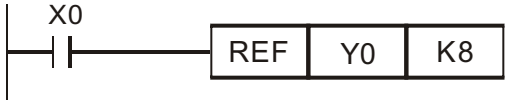
50 REF (D) (n) Input/Output refresh immediately

(D)	X0, X10, Y0, Y10
(n)	K8, K16, H8, H10

- The state of all PLC inputs and outputs will be refreshed after scanning to END. The state of inputs is read from external inputs to save in inputs memory. The output terminals send outputs memory to output device after END command. Therefore, this command can be used during algorithm process when need to input/output the newest data.
- The state of all inputs and outputs may change immediately after they are scanned. If the user does not want to wait for the next scan time, the instruction REF may be used.
- The input points and output points that this command handles is the I/O point of MPU: X0~X17, Y0~Y17
- **Program Example**

◎ When X0 = ON, the 8 input points will be scanned immediately.

- **Program Example**



◎ When X0 = On, the output signal Y0~Y7 (8 points) are sent to output terminal.

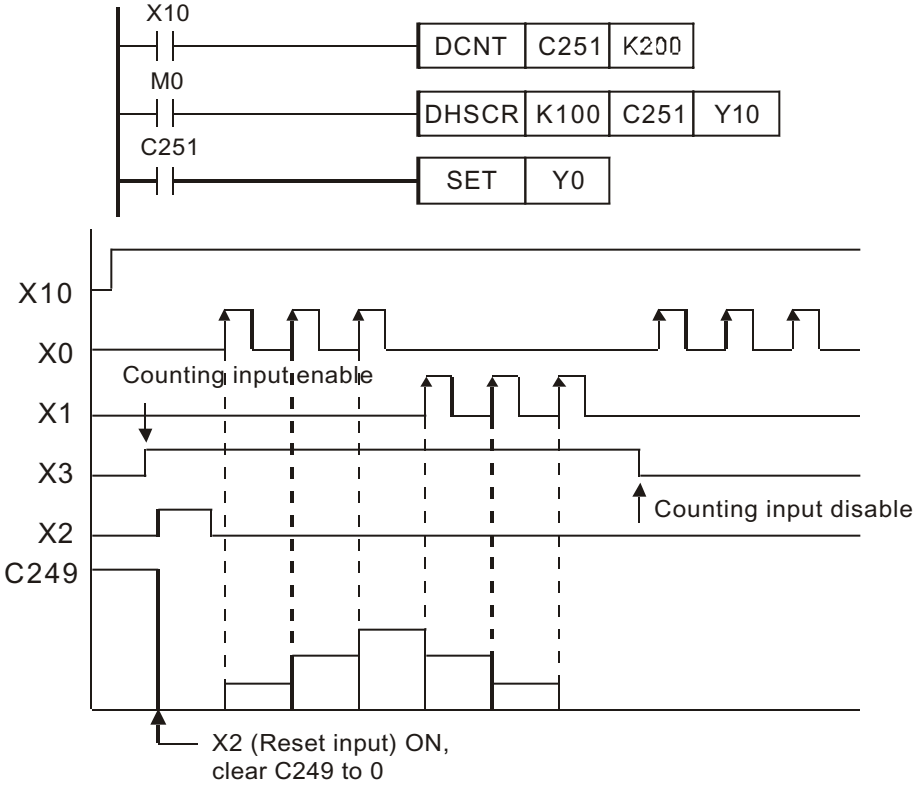
D **53** **HSCS** (S1) (S2) (D) 32-bit High speed counter comparison set

Operand \ Device	Bit device				Word device								
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)	C235~C254												
(D)		◎	◎	◎									

D **54** **HSCR** (S1) (S2) (D) 32-bit High speed counter comparison reset

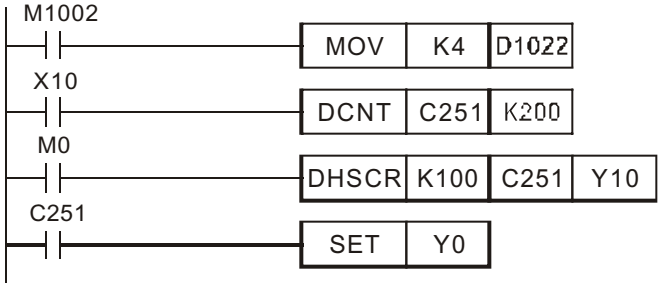
Operand \ Device	Bit device				Word device								
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)	C235~C254												
(D)		◎	◎	◎									

- There is no 16-bit command for API 53 and API 54. (only 32-bit command, DHSCS and DHSCR are available)
- Users must use X0~X3 for High-Speed Counter inputs.
- The goal of counting is to do a special action when the count (S2) reaches a preset value (S1). A preset is a number you derive and store so the counter will constantly compare and use for other functions.
- The counter compares the current count with up to 4 preset values, which you define by using instruction DHSCS and DHSCR. If (D) is device Y, then only devices Y00~Y17 are effective.
- All high speed counters have its specified high speed counter terminals. Every input rapid pulse by high speed counting use an interrupt process to input signal counting value.
- **Program Example**

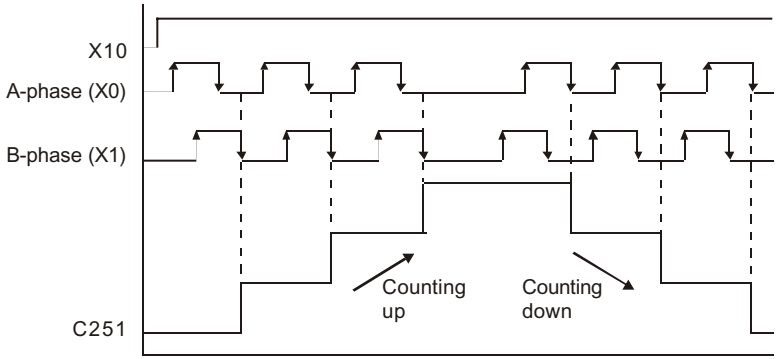


- When M0 = On and the present value of the high speed timer C249 changes from (99 to 100) or (101 to 100), then Y10 will be ON.
- When the present value of high-speed timer C249 changes from (999 to 1000) or (1001 to 1000). C249 will be activated, and Y17 will be ON, but there will be a delay due to the program scan time.
- **Program Example**

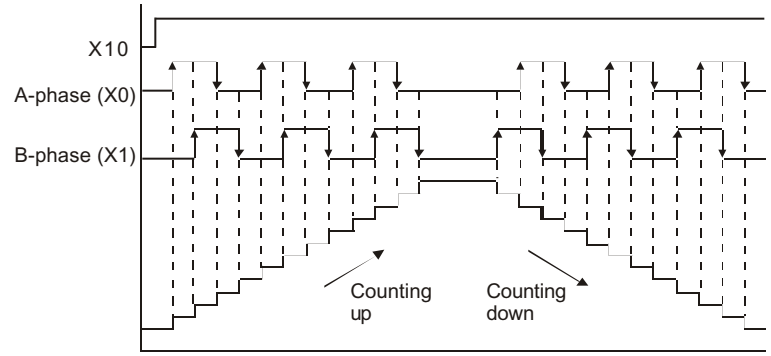
◎ AB phase high speed counter can be changed to inactivated by using D1022 double frequency setting mode when PLC goes from STOP to RUN.



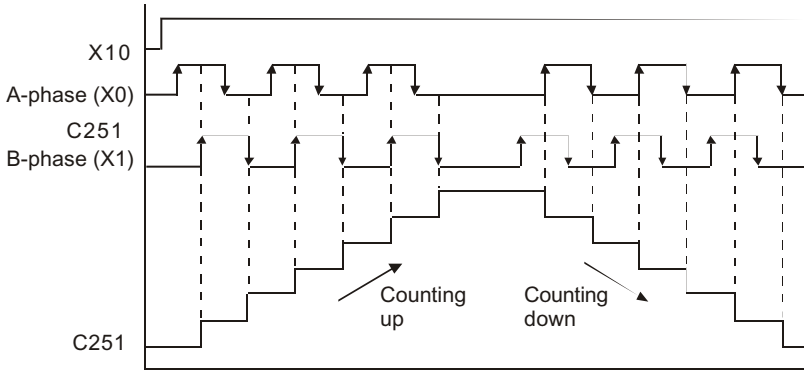
◎ When D1022=K1, the timing diagram of one double frequency will be as follows:



◎ When D1022=K4, the timing diagram of one double frequency will be as follows:



◎ When D1022=other value, the timing diagram of tw0 double frequency will be as follows:



■ When M0 = ON, High speed counter C251 counts as follow:

Count value			Contact status	Y10
101	→	100	No change	No change
100	→	99	OFF	OFF
99	→	100	ON	ON
100	→	101	ON	ON

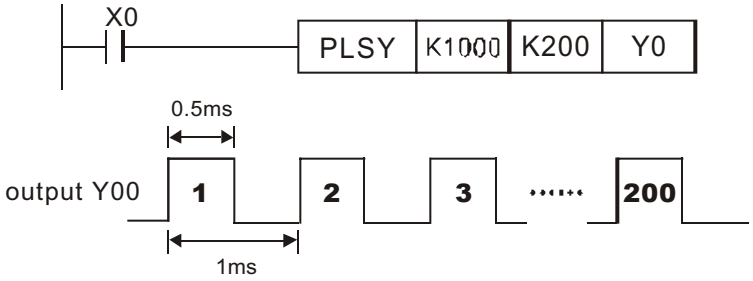
■ When high-speed counter C251' s value changes from (199 to 200) or from (201 to 200), the contact of C251 will be ON and Y10 will be ON. However, there will be scan time delay.

D **57** **PLSY** (S1) (S2) (D) Pulse output

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)	Y0, Y1														

■ Generate specified frequency and number of pulse commands

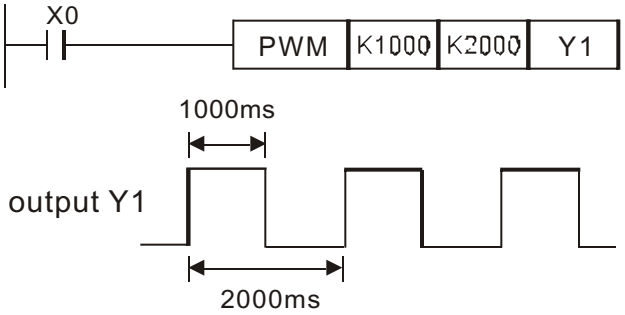
- (S1) specified the frequency 10 ~ 10KHz.
- 10KHz could be reached with single shaft, whereas 5KHz could be reached with dual shaft.
- (S2) specified the pluses. 16-bit: 1 to 32767, 32-bit: 1 to 2147483647
- (D) specified the output Y as output pulse, only Y0, Y1 is effective (please use the transistor output as the output module).
- When M1010 is ON, the Y0 pulse wave would output continuously and would not be restricted to the pulse wave' s quantity. When M10230 is ON, the Y1 pulse wave will output continuously and would not be restricted to the pulse wave quantity.
- After the output of the Y0 pulse wave is completed, M1029 will be set as ON. After the output of the Y1 pulse wave is completed, M1030 will be set as ON. And when the PLSY command is OFF, M1029 or M1030 will be OFF.
- **Program Example**



58 PWM (S1) (S2) (D) Pulse width modulation output

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)	Y01														

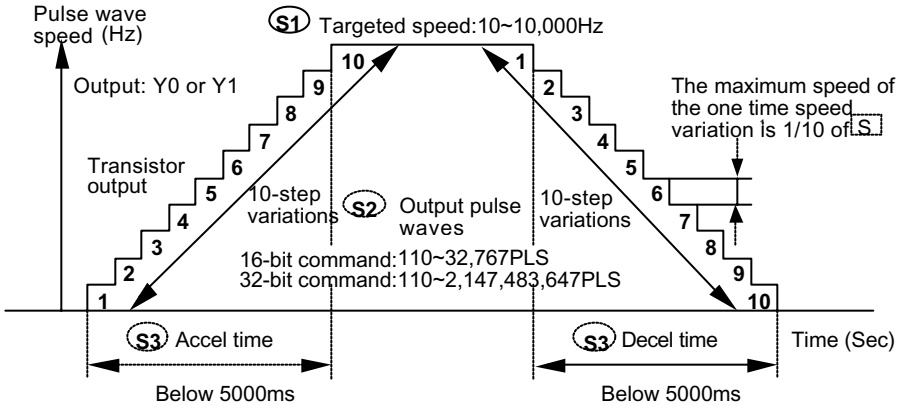
- (S1) specified the pulse width as t: 0 to 32767ms.
- (S2) specified cycle as T: 1 to 32767ms (S1) ≦ (S2).
- (D) specified the output Y as output pulse, only Y01 is effective.
- Once M1070 is of the PWM command, the pulse unit will switch the flag. If ON it is 100 μs, and if OFF, 1ms.
- **Program Example**



59 PLSR (S1) (S2) (S3) (D) Pulse wave output with accel/decel speeds

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S3)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)	Y0, Y1														

- The wave output command with the accel/decel speeds.
- The acceleration is conducted when the pulse wave goes from the static status to reaching its targeted speed, and getting faster when the targeted speed is to be reached. The pulse wave will stop its output once the targeted distance is reached.



■ **Program Example**

◎ Settings of all the operands are as follows.

S1 Maximum speed (H_Z)

- Settings: 10~10,000 (H_Z)
  The maximum speed is deemed to be the multiples of 10, if not, the first unit will be discarded automatically.
- 1/10 of the maximum speed is the one time variation of the accel/decel speed. Note that the condition meets the acceleration requirement of the step motor and would not result in the step motor crash.

S2 Content of the pulse wave output quantity (PLS)

- Settings: 16-bit command: 110~32,767(PLS)
  32-bit command: 110~2,147,483,647(PLS)
  If the setting is below 110, the pulse wave cannot output normally.
- When using the 32-bit command, DPLSR, the output pulse wave quantity is the content of D1 and D0.

S3 Acceleration/Deceleration time (ms)

- Setting: below 5000ms, but have to meet the following three conditions. The accel time and the decel time have to be the same and cannot be set without one another.

① The accel/decel time has to be over 10 times the maximum scan time (contents of D1012). If the setting is below 10 times, the slope of the accel/decel speed will be inaccurate.

② Minimum setting of the accel/decel time could be obtained from the following equation.

$$S_3 \geqslant \frac{90000}{S_1} \times 1.22$$

If the setting is smaller than the result of the above-mentioned equation, the acceleration/deceleration time will be greater, and if the setting is smaller than the minimum setting, the minimum setting will be treated as its regular setting.

③ Maximum setting of the accel/decel time could be obtained from the following equation.

$$S_3 \leqslant \frac{S_2}{S_1} \times 818$$

④ Number of the accel/decel speed variation steps is fixed to be 10.

If the above-mentioned requirements cannot be met, please lower down the maximum speed.

Among all the output Y numbers of the **D**-assigned output pulse wave, only Y0 and Y1 are effective (please use the transistor output module).

◎ The speed range for the pulse wave of this command is 2~10,000H_Z. And if the settings for the high speed and the accel/decel time exceed this range, use the allowable setting within this range for operation.

◎ When X10 is OFF, output will be interrupted, and when turned ON again, counting of the pulse wave will be counted from 0.

◎ It is not acceptable to change the setting of every operand during the execution of the command. The previous setting would only be valid when the command is executed again.

◎ Once the (S2)-set pulse waves are transmitted, the Y0 output will be completed and M1029 =ON, and the Y1 output will be cmpleted and M1030 =ON.

◎ Number of times of the command usage
For commands PLSY (DPLSY), PWM and PLSR (DPLSR), they could only be used once for each output.

60 IST (S) (D1) (D2) Manual/Auto Control

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)	◎	◎	◎												
(D1)				◎											
(D2)				◎											

■ Command IST is for initializing the special step ladder flow control. Configuring with the special relay to perform as the auto control mode.

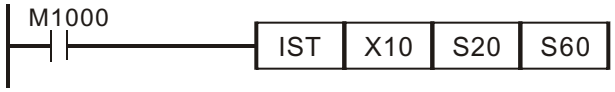
■ (S) starting input device for specified operation mode.

■ (D1) initial step point in automatic control mode.

■ (D2) The largest step point in automatic control mode.

■ (D1), (D2): S20~S127, However, (D2) > (D1).

■ **Program Example**



(S) X20: Manual operation mode
X21: Origin reset mode
X22: Step operation mode
X23: One program cycle operation mode
X24: Continuously operation mode
X25: Origin reset mode starting switch
X26: Step/One cycle/Continuously cycle mode starting switch
X27: Step/One cycle/Continuously cycle mode stopping switch

◎ When command IST is in operation, the listed special relay will automatically switch.

M1040: Step Transition Inhibit
M1041: Step point Transition Start
M1042: Start pulse
M1047: STL Monitoring enable

S0: Manual mode initial step point
S1: Original point back initial step point
S2: Auto mode initial step point

When using the IST command, S10 to S19 are for retruning to the original point. These step points can not be used in other step ladder programs.

66 ALT (D) ON/OFF Alternate command

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(D)		◎	◎	◎											

■ **Program Example**

■ When X0 is activated for the first time, M0=ON, Y1=ON. When X0 is activated for the second time, M0=OFF, Y0=ON, Y1=OFF.

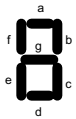
73 SEGD (S) (D) Decode the 7-step display panel

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ **Program Example**

◎ Contents (0~F: 16 bits) of the lower 4 bits (b0~b3) of D0 will be decoded as readable in the 7-step display panel for output. The decoding results will be saved in K2Y0.

◎ Decodimg Chart of the 7-Step Display Panel

16bits	Bit Combi-nation	Composition of the 7-Step Display Panel	Status of Every Step							Data Displayed
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	OFF	2
3	0011		ON	ON	ON	ON	ON	OFF	OFF	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		OFF	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	OFF	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	OFF	ON	ON	ON	9
A	1010		OFF	OFF	OFF	ON	ON	OFF	ON	A
B	1011		OFF	OFF	ON	ON	OFF	OFF	ON	b
C	1100		OFF	ON	OFF	OFF	OFF	ON	ON	C
D	1101		ON	OFF	OFF	ON	OFF	ON	ON	d
E	1110		OFF	OFF	OFF	ON	ON	ON	ON	E
F	1111		OFF	OFF	OFF	OFF	OFF	OFF	OFF	

74 SEGL (S) (D) (n) 7-step display scan output

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(D)		◎													
(n)					◎	◎									

Note: The usage range of operand n is 0~7. Please refer to function specification chart for device usage range. SEGL command can be used twice in the program. Please refer to footnote for detail.

- (S): display source device of 7-step display. (D): start device of 7-step display scan output (n): polarity setting of output signal and scan signal.
- 8 or 12 continuous external output points that start from this command (D) output 1 or 2 groups of 4 digits of 7-step display by scanning and display the content of (S) on 7-step display. (n) will decide the numbers of groups of 4 digits of 7-step display and also indicate the polaritys of PLC output terminal and 7-step display input terminal.
- The points number of 7-step display output command that a group of 4 digits use is 8 points and 2 groups of 4 digits use are 12 points.
- Scan output terminal will circulate in sequence when this command executes. The condition contact will be changed from OFF to ON and scan output execute again.
- **Program Example**:

◎ When X10=ON, command will start to execute. 7-step display scan loop is composed of Y10~Y17. The value of D10 will be converted to BCD code and send to the first group of 7-step display to display. The value of D11 will be converted to BCD code and send to the second group of 7-step display to display. If any value of D10 or D11 is greater than 9999, operation error will happen.

◎ When X10=ON, Y14~Y17 will scan in circles automatically. Each circle scan needs 12 scan time. M1029=ON is a scan period after a circle scan.

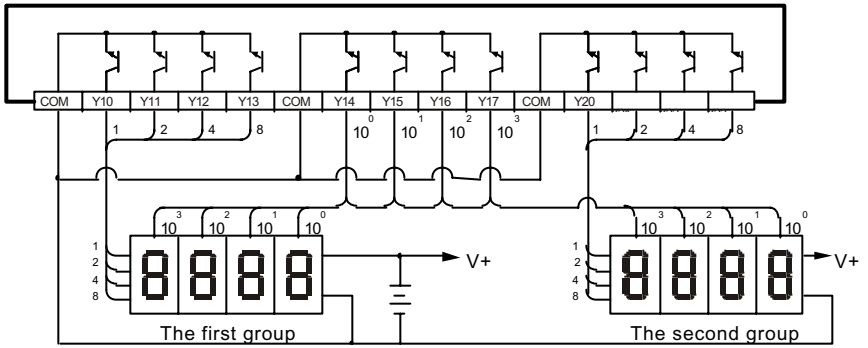
◎ 4 digits of a group, n=0~3.

- After the terminal of 1, 2, 4, 8 of decoded 7-step display connects itself in parallet, they should connect to Y10~Y13 of PLC. Latch terminal of each number connects to Y14~Y17 of PLC individually.
- When X10=ON, the content of D10 will be transmitted to 7-step display to display in sequently according to Y14~Y17 circulates in sequence.

◎ 4 digits of 2 groups, n=4~7.

- After the terminal of 1, 2, 4, 8 of decoded 7-step display connects itself in parallet, they should connect to Y20~Y23 of PLC. Latch terminal of each number and the first group share Y14~Y17 of PLC.
- The content of D10 will be transmitted to the first group of 7-step display and the content of D11 will be transmitted to the second group of 7-step display to display.

◎ 7-step display scan output wiring.



■ Footnote:

◎ The version V4.9 or above of ES / EX / SS series has this command (SEGL).

◎ Version 4.9 of ES / EX / SS series has only a group of 4 digits of 7-step display and use 8 points to output. It only uses SEGL command one time in the program and the usage of n operand is n=0~3.

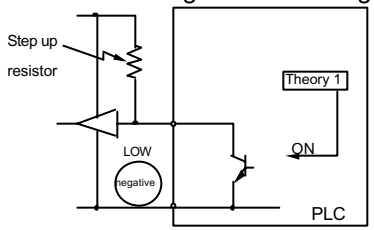
◎ Scan time must be longer than 10ms when executing this command. If scan time is shorter than 10ms, please use fixed scan time function to fix scan time on 10ms.

◎ Please use suitable 7-step display for the transistor that PLC uses to output.

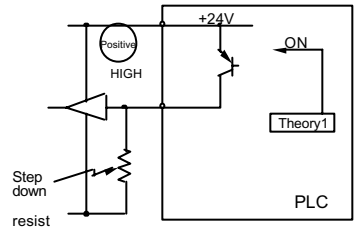
◎ Settings of n: it is used to set the polarity of transistor output loop. It can be set to positive polarity or negative polarity. what 7-step display it connects is a group of 4 digits or two groups of 4 digits.

A. Polarity of PLC output

■ Output loop of NPN transistor: when inner signal is "1", it will output low potential. This logic is called negative polarity.



■ Output loop of PNP transistor: when inner signal is "1", it will output high potential. This logic is called positive polarity.



■ 7-step display polarity

	Positive polarity	Negative polarity
Data input	When high potential, output in BCD type	When low potential, output in BCD type
Scan signal	When high potential, it will display latched.	When low potential, it will display latched.

■ Settings of parameter n

Group number of 7-step display group	A group				2 groups			
The polarity of PLC output terminal and display data input terminal	✓		×		✓		×	
The polarity of PLC output terminal and display scan siganl input terminal	✓	×	✓	×	✓	×	✓	×
n	0	1	2	3	4	5	6	7

✓: is the same. ×: is different

◎ The combination of output polarity of PLC transistor and input polaity of 7-step display can be set by settings of n.

◎ If output polarity of PLC is negative, input polarity of 7-step display is also negative and input terminal of scan signal of 7-step display is positive polarity. n will be 1 when a group of 4 digits and n will be 5 when two groups of 4 digits.

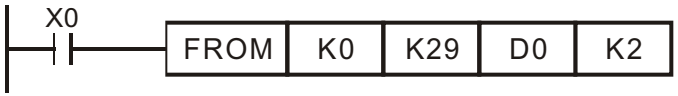
78 FROM (m1) (m2) (D) (n) Read special module CR data

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(m1)					◎	◎									
(m2)					◎	◎									
(D)								◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

Note: The usage range of operand m_1 is 0~7. The usage range of operand m_2 is 0~35. The usage range of operand n is 36-m2. Please refer to function specification chart for each device usage range.

- (m1): the number for special module. (m2): the number of CR (Control Register) of special module that will be read. (D): the location to save reading data. (n): the data number of reading one time.
- DVP PLC uses this command to read CR data of special module.
- When (D) indicates bit operand, you can use K1~K4 for 16-bit command and K1~K8 for 32-bit command.

- **Program Example**



◎ To read the content of CR#29 of special module#0 to D0 of PLC and to read the content of CR#30 of special module#0 to D1 of PLC. It can read 2 data at one time (n=2).

◎ The command will be executed when X0=ON. The command won't be executed when X0=OFF and the content of previous reading data won't change.

79 TO (m1) (m2) (S) (n) Special module CR data write in

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(m1)					◎	◎									
(m2)					◎	◎									
(S)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(n)					◎	◎									

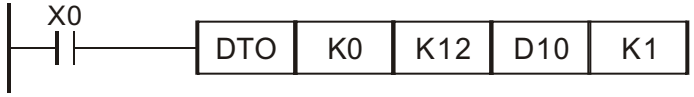
Note: The usage range of operand m_1 is 0~7. The usage range of operand m_2 is 0~35. The usage range of operand n is 36-m2. Please refer to function specification chart for every device usage range.

- (m1): the number of special module. (m2): the number of CR (Control Register) of special module that will be wrote in. (S): the data to write in CR. (n): the data number to write in one time.
- DVP-series PLC uses this command to write data into CR of special module.
- (S): When assigning bit operand, K1~K4 can be used for 16-bit and K5~K8 can be used for 32-bit.

- **Program Example**

◎ Using 32-bit command DTO, program will write D11 and D10 into CR#13 and CR#12 of special module#0. It only writes a group of data at one time (n=1).

◎ The command will be executed when X0=ON and it won't be executed when X0=OFF. The data that wrote in previous won't have any change.



- **Footnote:**

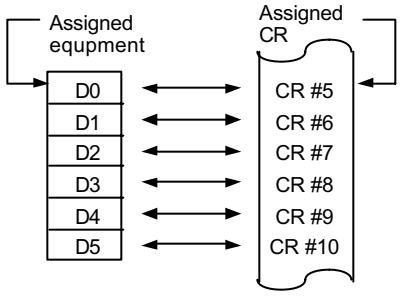
◎ The version 4.9 and above of ES / EX / SS series models support continuous execution commands (FROM, DFROM, TO, DTO). Other version won' t support these commands.

◎ The rule of command operand

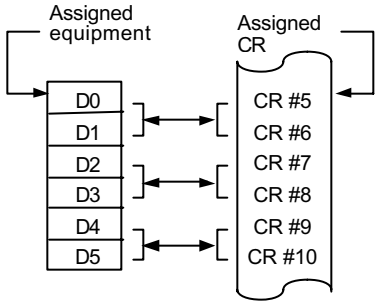
- m1: arrangement number of special module. The number of special module that connects to PLC MPU. The numbering order of special module from the near to the distant of MPU is from 0 to 7. The maximum is 8 special modules and won' t occupy I/O point.
- m2: the number of CR. Built-in 16-bit of 36 groups memory of special module is called CR (Control Register). The number of CR uses decimal digits (#0~#35). All running status and setting values of special module have included.
- If using FROM/TO command, the unit of read/write of CR is one number for one time. If using DFROM/DTO command, the unit of read/write of CR is two numbers in one time.

Upper 16-bit	Lower 16-bit	
CR #10	CR #9	← Assigned CR number

- The number of transmission groups n. The meaning of n=2 of 16-bit command and n=1 of 32-bit are the same.



16-bit command when n=6

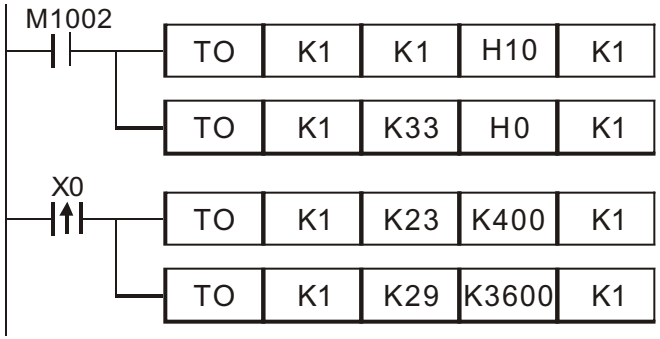


32-bit command when n=3

◎ When FROM/TO command is executed, all interrupts (including external or internal interrupt subroutines) will be prohibited. All interrupts will be executed after FROM/TO command is finished. Besides, FROM/TO command can be put in the subroutine.

■ **FROM / TO Application Program Example Explanation**

For example: Adjust A/D conversion characteristic curve by setting OFFSET value of CH1 to 0V(=$K0_{LSB}$) and GAIN value of CH1 to 2.5V(=$K2000_{LSB}$).



1. Writing H10 to CR#1 of analog input mode no. 1 and set CH2 to mode 2 (current input : +4mA ~ +20mA).
2. Writing H0 to CR#33 and allow to adjust characteristics of CH1 and CH2.
3. When X0 switches from OFF to ON, $K400_{LSB}$ of OFFSET value will be wrote in CR#23 and $K3600_{LSB}$ of GAIN value will be wrote in CR#29.

◎ Switch flag M1083 of EH series model instruction mode function:

- FROM/TO commands will be executed when M1083=OFF. All interrupts (including external or internal interrupt subroutines) will be prohibited. All interrupts will be executed after FROM/TO command is finished. Besides, FROM/TO command can be put in the subroutine.
- If there is any interrupts happen when FROM/TO command is executed during M1083=ON, FROM/TO command will be interrupted to execute interrupt signal. But FROM/TO command can' t be put in the subroutine.

80 RS (S) (m) (D) (n) Data Communication

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(m)					◎	◎							◎		
(D)													◎		
(n)					◎	◎							◎		

Note: The usage range of operand m is 0~256.The usage range of operand n is 0~256.Please refer to function specification chart for every device usage range.

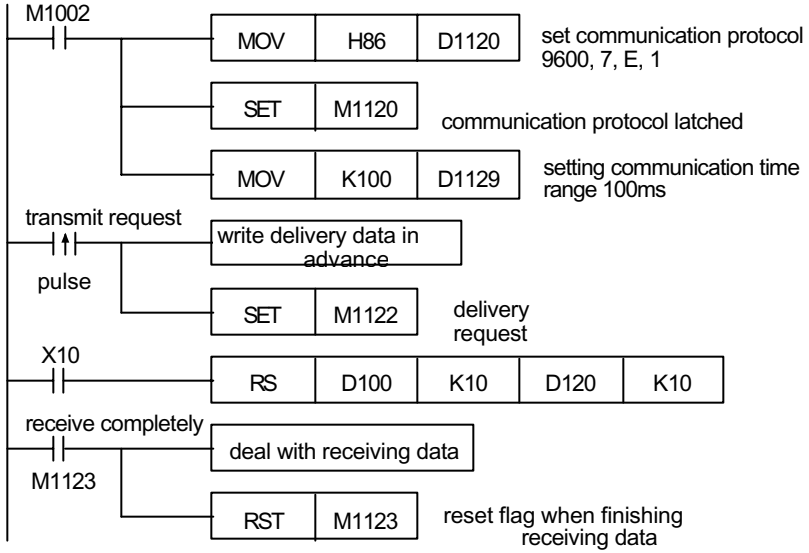
- (S): start device of transmitting data. (m): transmitting data group number. (D): start device of receiving data. (n): receiving data group numbers.
- This instruction is a convenience instruction for MPU to use RS-485 to connect communication interface in series. It saves words data in (S) source data register and sets length (m). It also sets receive data register (D) and length.
- If it doesn't need to transmit data, (m) can be indicated to K0 and if it doesn't need to receive data, (n) can be indicated to K0.
- You can use RS command in the program unlimitedly, but you can't execute two or more RS commands at the same time.
- It is invalid to change delivery data during executing RS command.
- PLC user can transmit data of PLC and peripheral equipment if peripheral equipment has RS-485 series communication and communication format of this equipment is public.
- If communication format of peripheral equipment corresponds with communication format of MODBUS, DVP series PLC provides several convenience communication commands, API 100 MODRD, API 101 MODWR and API 150 MODRW, for user to use. Please refer to individual instruction for detail.
- Please refer to following footnote for flag special auxiliary relay M1120~M1161 and special data register D1120~D1131 that relates to RS-485 communication.

■ **Program Example 1:**

◎ Writing data into the register that starts from D100 and set M1122 (delivery request flag) to ON.

◎ If RS command is executed when X10=ON, PLC will in the state of waiting for transmitting and receiving data. It will start to transmit 10 continuous data that start from D100. M1122 will be set to OFF at the end of transmitting. (Please don't execute RST M1122 by program) After 1ms, it will start to receive external 10 data and save them into continuous registers that start from D120.

◎ When finishing to receive data, M1123 will be set to ON. (Program will set M1123 to OFF when finishing to receive data and in the state of waiting transmitting and receiving. Please don't execute RST M1123 continuously by PLC program.



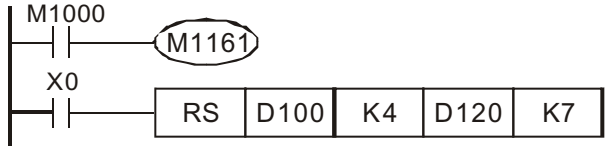
■ **Program Example 2:**

◎ 8 bits mode (M1161=ON) / 16 bits mode (M1161=OFF) switch:

《8 bits mode》:

Head code and tail code of PLC transmission data will be set by using M1126 and M1130 according to D1124~D1126. After setting, PLC will send head code and tail code that set by user automatically when executing RS command.

When M1161=ON, the conversion mode will be 8 bits. 16 bits data will be divided into upper 8 bits and lower 8 bits. Upper 8 bits will be ignored and lower 8 bits will be received and transmitted.



transmit data: (PLC→external machine)

STX	D100 lower	D101 lower	D102 lower	D103 lower	EXT1	EXT2

Head code — (S) source data register will start from lower 8 bits of D100 — tail code 1 — tail code 2

(m) length= 4

receive data: (external machine→PLC)

D120 lower	D121 lower	D122 lower	D123 lower	D124 lower	D125 lower	D126 lower

Head code — (D) receive data register will start from lower 8 bits of D120 — tail code 1 — tail code 2

(n) length= 7

PLC will receive all data that transmits from external machine, including head code and tail code. Please pay attention when setting length (n).

《16 bits mode》:

Head code and tail code of PLC transmitting data is set by using M1126 and M1130 according to D1124~D1126. After setting, PLC will send head code and tail code that set by user automatically when executing RS command.

When M1161=OFF, the conversion mode will be 16 bits. 16 bits data will be divided into upper 8 bits and lower 8 bits to receive and transmit data.

transmit data: (PLC→external machined)

STX	D100 lower	D100 upper	D101 lower	D101 upper	EXT1	EXT2

Head code — (S) Source data register will start from lower 8 bits of D100 — tail code 1 — tail code 2

(m) length= 4

receive data: (external machine→PLC)

D120 lower	D120 upper	D121 lower	D121 upper	D122 lower	D122 upper	D123 lower

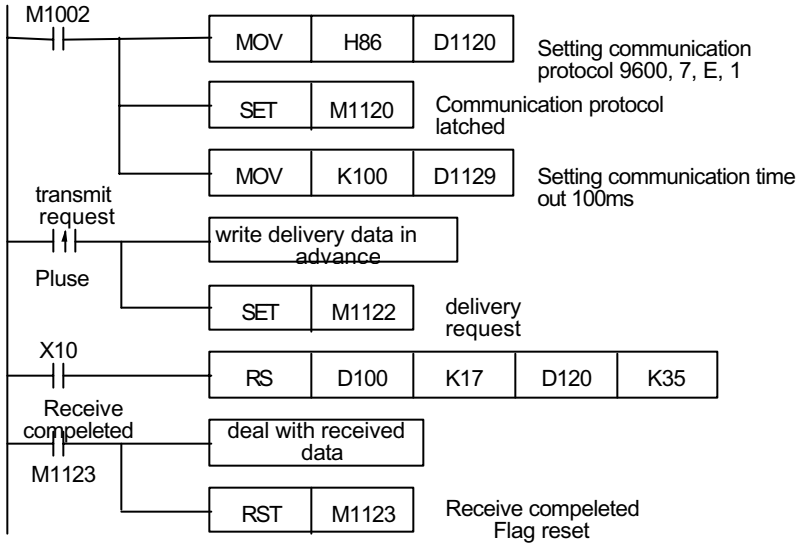
Head code — (D) receive data register will start from lower 8 bits of D120 — tail code 1 — tail code 2

(n) length= 7

PLC will receive all data that transmits from external machine, including head code and tail code. Please pay attention when setting length (n).

■ **Program Example 3:**

◎ When PLC connects to VFD-B series AC drives (ASCII Mode, M1143=OFF), (16 bits Mode, M1161=OFF), it will transmit data to read 6 continuous data that start from VFD-B reference address H2101.



PLC ⇨ VFD-B, PLC transmits: **“: 01 03 2101 0006 D4 CR LF ”**

VFD-B ⇨ PLC, PLC receives: **“: 01 03 0C 0100 1766 0000 0000 0136 3B CR LF ”**

PLC transmits data register (PLC transmits messages)

Register	DATA			
D100 lower	‘ : ’	3AH	STX	
D100 upper	‘ 0’	30 H	ADR 1	ADR (1,0) is AC drive address
D101 lower	‘ 1’	31 H	ADR 0	
D101 upper	‘ 0’	30 H	CMD 1	CMD (1,0) is command code
D102 lower	‘ 3’	33 H	CMD 0	
D102 upper	‘ 2’	32 H	Start data address	
D103 lower	‘ 1’	31 H		
D103 upper	‘ 0’	30 H		
D104 lower	‘ 1’	31 H		
D104 upper	‘ 0’	30 H	Number of data(count by word)	
D105 lower	‘ 0’	30 H		
D105 upper	‘ 0’	30 H		
D106 lower	‘ 6’	36 H		
D106 upper	‘ D’	44 H	LRC CHK 1	LRC CHK (0,1) is fault check code
D107 lower	‘ 4’	34 H	LRC CHK 0	
D107 upper	CR	A H	END	
D108 lower	LF	D H		

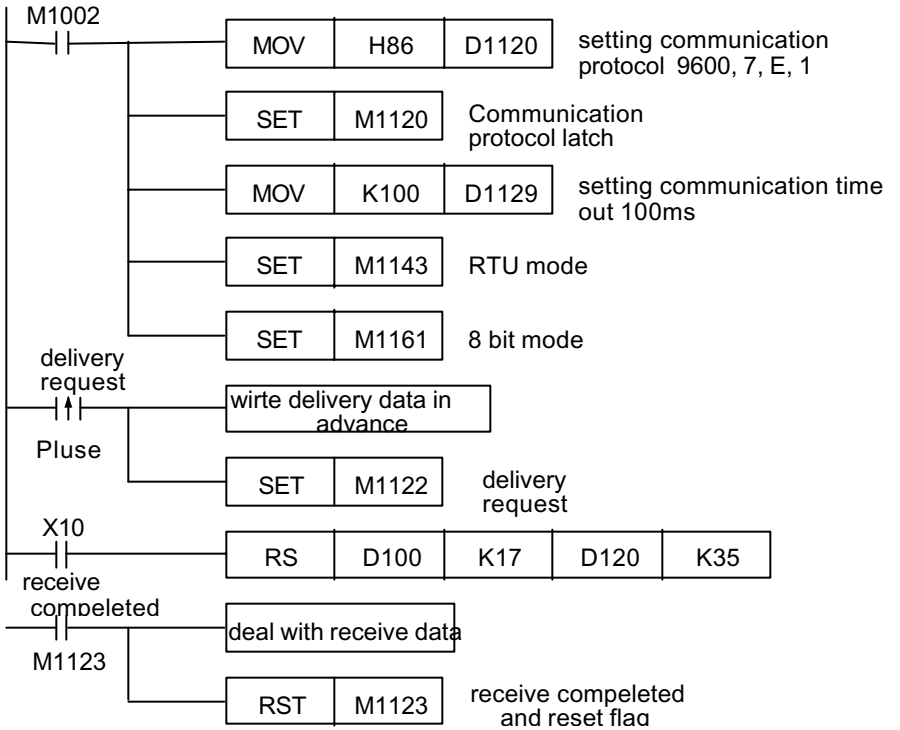
PLC receive data register (VFD-B response messages)

Register	DATA		
D120 lower	‘ : ’	3AH	STX
D120 upper	‘ 0’	30 H	ADR 1
D121 lower	‘ 1’	31 H	ADR 0
D121 upper	‘ 0’	30 H	CMD 1
D122 lower	‘ 3’	33 H	CMD 0
D122 upper	‘ 0’	30 H	Number of data (count by byte)
D123 lower	‘ C’	43 H	
D123 upper	‘ 0’	30 H	Content of address 2101 H
D124 lower	‘ 1’	31 H	
D124 upper	‘ 0’	30 H	
D125 lower	‘0’	30 H	
D125 upper	‘1’	31 H	Content of address 2102 H
D126 lower	‘7’	37 H	
D126 upper	‘ 6’	36 H	
D127 lower	‘ 6’	36 H	

Register	DATA		
D127 upper	‘ 0’	30 H	Content of address 2103 H
D128 lower	‘ 0’	30 H	
D128 upper	‘ 0’	30 H	
D129 lower	‘ 0’	30 H	
D129 upper	‘ 0’	30 H	Content of address 2104 H
D130 lower	‘ 0’	30 H	
D130 upper	‘ 0’	30 H	
D131 lower	‘ 0’	30 H	
D131 upper	‘0’	30 H	Content of address 2105 H
D132 lower	‘1’	31 H	
D132 upper	‘3’	33 H	
D133 lower	‘6’	36 H	
D133 upper	‘0’	30 H	Content of address 2106 H
D134 lower	‘0’	30 H	
D134 upper	‘0’	30 H	
D135 lower	‘0’	30 H	
D135 upper	‘3’	33 H	LRC CHK 1
D136 lower	‘B’	42 H	LRC CHK 0
D136 upper	CR	A H	END
D137 lower	LF	D H	

■ **Program Example 4**

◎ PLC connects to VFD-B AC drive (ASCII Mode, M1143=ON), (16 bits Mode, M1161=ON). To write delivery data, H12, in advance into parameter address of VFD-B H2000.



PLC ⇨ VFD-B, PLC transmits: **01 06 2000 0012 02 07**

VFD-B ⇨ PLC, PLC receives: **01 06 2000 0012 02 07**

PLC transmits data register (transmit messages)

PLC transmits data register (PLC transmits messages)

Registers	DATA	
D100 lower	01 H	Address
D101 lower	06 H	Function
D102 lower	20 H	Data address
D103 lower	00 H	

D104 lower	00 H	Data content
D105 lower	12 H	
D106 lower	02 H	CRC CHK Low
D107 lower	07 H	CRC CHK High

PLC receives data register (response messages of VFD-B)

Registers	DATA	
D120 lower	01 H	Address
D121 lower	06 H	Function
D122 lower	20 H	Data address
D123 lower	00 H	
D124 lower	00 H	Data content
D125 lower	12 H	
D126 lower	02 H	CRC CHK Low
D127 lower	07 H	CRC CHK High

■ **Footnote:**

◎ RS-485 communication RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW commands relation flag signal:

Flag	Function Explanation
M1120	It is used to set communication latch. PLC will reset communication protocol setting according to special data register D1120 after the first program scan. When second program scan starts and RS command is executed, it will reset communication protocol setting according to special data register D1120. If communication protocol is fixed, M1120 can be set to ON. At this time, communication protocol setting won't be reset as RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW is executed even if D1120 setting is changed.
M1121	It indicates that PLC can transmit data now.
M1122	Transmit request. Users need to set M1122 to ON by pulse command when using RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW command to transmit and receive data. If the command above is executing, PLC will transmit and receive data. M1122 will be clear after the commands above finish transmitting.
M1123	Receive completely. M1123 will be set to ON after RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW commands finish executing. User can deal with received data when M1123 is set to ON and clear M1123 to OFF when finish handling them.
M1124	Wait for receiving. When M1124 is set to ON, it means PLC is waiting for receiving data.
M1125	Received function disable. When M1125 is set to ON, the state of PLC transmits function disable.
M1126	Please refer to following chart for selecting user/system definition and STX/ETX.
M1127	M1127 should be clear to OFF when MODRD / RDST / MODRW commands finish receiving at ASCII mode.
M1128	transmitting / receiving indication
M1129	Receive time out. This flag will be active, if D1129 is set and receive data doesn't finish within the setting time. If the state disable, M1129 should be clear to OFF.
M1130	Please refer to following chart for selecting users/system definition and STX/ETX.
M1131	M1131=ON during MODRD / RDST / MODRW convert to HEX. Otherwise M1131 will be OFF.
M1140	MODRD / MODWR / MODRW data receive error
M1141	MODRD / MODWR / MODRW command parameter error
M1142	Data receive error of VFD-A convenience command
M1143	ASCII / RTU mode selection, ON is RTU mode and OFF is ASCII mode.
M1161	8/16 bits handle mode selection. ON is 8 bits mode and OFF is 16 bits mode.

◎ Special register of RS-485 communication RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW command relative setting

Special register	Function Explanation
D1038	Data response delay time setting when PLC MPU is slave. Time definition (0.1ms)
D1050~D1055	PLC will convert ASCII data of D1070~D1085 to HEX and save hexadecimal data to D1050~D1055.
D1070~D1085	Built-in RS-485 communication convenience command. This command will execute "send" command and receiver will return messages when it receives. These messages will be saved at D1070~D1085. User can check return data by viewing the register content.
D1089~D1099	It is PLC built-in RS-485 communication convenience command. The message that sent when this command is executed will be saved in D1089~D1099. Users can check by viewing the register.
D1120	Please refer to following chart for RS-485 communication protocol.
D1121	Communication address of PLC MPU when PLC MPU is slave.
D1122	Remainder words of delivery data.
D1123	Remainder words of receive data.
D1124	Start word definition (STX). Please refer to chart above.
D1125	The first end word definition. (ETX1) Please refer to chart above.
D1126	The second end word definition. (ETX1) Please refer to chart above.
D1129	Communication time out is abnormal. Time unit (ms). It is used to set time of time out. if it is 0, it means there is no time out. PLC will set M1129 to be ON if receiving time of the first word or between any two words is more than setting after executing RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW commands to enter received mode. Please pay attention to clear M1129 after handling.

Special register	Function Explanation
D1130	MODBUS return fault code record.
D1256~D1295	Built-in RS-485 communication convenience command MODRW. The command characters sent when this command is executed will be saved in D1256~D1295. User can check with the content of these registers. (Users only can use MOV, DMOV, BMOV to move the data in this area in version V4.9)
D1296~D1311	PLC will convert ASCII data in the register that user requests to hexadecimal. (Users only can use MOV, DMOV, BMOV to move the data in this area in version V4.9)

◎ D1120: RS-485 communication protocol. Please refer to following chart to set.

	Content	0	1
b0	Data length	7	8
b1 b2	Parity bits	00 : None	
		01 : Odd	
		11 : Even	
b3	stop bits	1 bit	2 bit
b4 b5 b6 b7	0011	(H3) : 300	
	0100	(H4) : 600	
	0101	(H5) : 1200	
	0110	(H6) : 2400	
	0111	(H7) : 4800	
	1000	(H8) : 9600	
	1001	(H9) : 19200	
	1010	(HA) : 38400	
	1011	(HB) : 57600 (only for EH series)	
	1100	(HC) : 115200 (only for EH series)	

	Content	0	1
b8	Start word selection	none	D1124
b9	The first end word selection	none	D1125
b10	The second end word selection	none	D1126
b15~b11	No definition		

◎ Start character and end character of control characters will be defined in the communication format of peripheral equipment when using RS command. Start character and end character can be set in D1124~D1125 by user or defined by machine. When users use M1126, M1130, D1124~D1125 to set start and end character, b8~b9 of D1120 of RS485 communication protocol should be set to 1. Please refer to the following chart for detail.

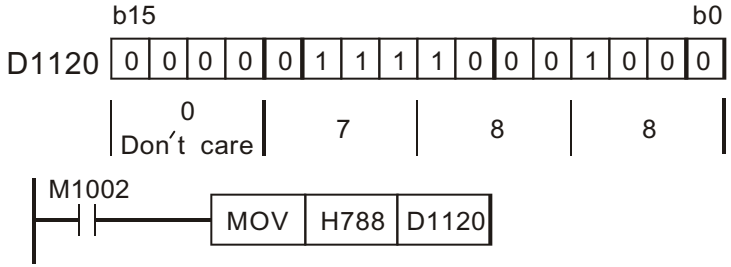
		M1130	
		0	1
M1126	0	D1124: user define D1125: user define D1126: user define	D1124: H 0002 D1125: H 0003 D1126: H 0000（no setting）
	1	D1124: user define D1125: user define D1126: user define	D1124: H 003A（':'） D1125: H 000D（CR） D1126: H 000A（LF）

◎ Example for communication format setting:

Communication format: Baud rate 9600 7, N, 2

STX : ": "
ETX1 : "CR"
EXT2 : "LF"

You can get the communication format H788 via check with chart and write into D1120.



You should pay attention to special auxiliary relay M1126 and M1130 when using STX, EXT1 and EXT2.

◎ M1143: ASCII / RTU mode selection. ON is RTU mode and OFF is ASCII mode.

Take standard MODBUS format to explanation.

ASCII mode (M1143=OFF):

STX	Start word = ': ' (3AH)
Address Hi	Communication address:
Address Lo	8-bit address consists of 2 ASCII codes
Function Hi	Function code:
Function Lo	8-bit function code consists of 2 ASCII codes
DATA (n-1)	Data content:
…….	nx8-bit data content consists of 2n ASCII codes
DATA 0	
LRC CHK Hi	LRC check sum:
LRC CHK Lo	8-bit check sum consists of 2 ASCII code
END Hi	End character:
END Lo	END Hi = CR (0DH), END Lo = LF(0AH)

Communication protocol is made of MODBUS ASCII(American Standard Code for Information Interchange). Each byte consists of 2 ASCII characters. For example: a 1-byte data 64 Hex shown as '64' in ASCII, consists of '6' (36Hex) and '4' (34Hex).

character	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII code	30H	31H	32H	33H	34H	35H	36H	37H
character	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII code	38H	39H	41H	42H	43H	44H	45H	46H

Start character (STX): ': ' (3AH)

Communication Address:

'0' '0': broadcast for all driver
'0' '1': toward the drive at the 01 address
'0' 'F': toward the drive at the 15 address
'1' '0': toward the drive at the 16 address . . . . . .
and consequently, the Max. to be reached is 255 ('F' 'F').

Function code:

'0' '3': read the contents of the register.

'0' '6': write one WORD into the register.

'1' '0': write contents of the register.

Data Characters:

The data characters that user transmits.

LRC check:

The LRC check is the added sum from "Address" to "Data Contents". For example, the 01H + 03H + 21H + 02H + 00H + 02H = 29H, then take the complementary of 2, D7H.

End character:

END Hi = CR (0DH), END Lo = LF(0AH)

For example: when the address of the drive is set as 01H, read 2 data contents that exist successively within the register, as shown follows: the address of the start register is 2102H.

Inquiry message:

STX	': '
Address	'0'
	'1'
Function	'0'
	'3'
Start address	'2'
	'1'
	'0'
	'2'
Number of data (count by word)	'0'
	'0'
	'0'
	'2'
LRC Check	'D'
	'7'
END	CR
	LF

Response message:

STX	': '
Address	'0'
	'1'
Function	'0'
	'3'
Number of data (count by byte)	'0'
	'4'
Content of start address 2102H	'1'
	'7'
	'7'
	'0'
Content of address 2103H	'0'
	'0'
	'0'
	'0'
LRC Check	'7'
	'1'
END	CR
	LF

The RTU Mode (M1143=ON):

START	Please refer to following explanation
Address	Communication address: 8-bit binary
Function	Function code: 8-bit binary
DATA (n-1)	Data characters: n×8-bit data
…….	
DATA 0	
CRC CHK Low	CRC check: 16-bit CRC consists of 2 8-bit binary
CRC CHK High	
END	Please refer to following explanation

START:

ES / EX / SS / EP series: keep none input signal to be greater or equal to 10 ms.

EH series:

Baud Rate(bps)	RTU Timeout Timer(ms)	Baud Rate(bps)	RTU Timeout Timer(ms)
300	40	9600	2
600	21	19200	1
1200	10	38400	1
2400	5	57600	1
4800	3	115200	1

Communication Address:

00 H: broadcast all drives
01 H: toward the drive at the 01 address
0F H: toward the drive at the 15 address
10 H: toward the drive at the 16 address…..,
and consequently, the Max. to be reached is 255 (FF H).

Function code:

03 H: read the contents of the register

06 H: write one WORD into the register

01 H: write the contents of the register

Data Characters:

The data contents that user transmits

CRC check:

The CRC check starts from "Address" and ends in "Data Content". Its calculation is as follows:

Step 1: Load the 16-bit register (the CRC register) with FFFFH.

Step 2: Exclusive OR the first 8-bit byte message command with the 16-bit CRC register of the lower bit, then save the result into the CRC register.

Step 3: shift the CRC register one bit to the right and fill in 0 to the higher bit.

Step 4: check the value that shifts to the right. If it is 0, save the new value from step 3 into the CRC register, otherwise, Exclusive OR A001H and the CRC register, then save the result into the CRC register.

Step 5: repeat step 3 and 4 and calculates the 8-bit.

Step 6: Repeat Steps 2~5 for the next 8-bit message command, till all the message commands are processed. And finally, the obtained CRC register value is the CRC check value. What should be noted is that the CRC check must be placed interchangeably in the check sum of the message command.

END:

ES / EX / SS / EP series: keep none input signal to be greater or equal to 10 ms

EH series:

Baud Rate(bps)	RTU Timeout Timer(ms)	Baud Rate(bps)	RTU Timeout Timer(ms)
300	40	9600	2
600	21	19200	1
1200	10	38400	1
2400	5	57600	1
4800	3	115200	1

For example: when the address of the drive is set as 01H, read 2 data contents that exist successively within the register, as shown follows: the address of the start register is 2102H.

Inquiry:

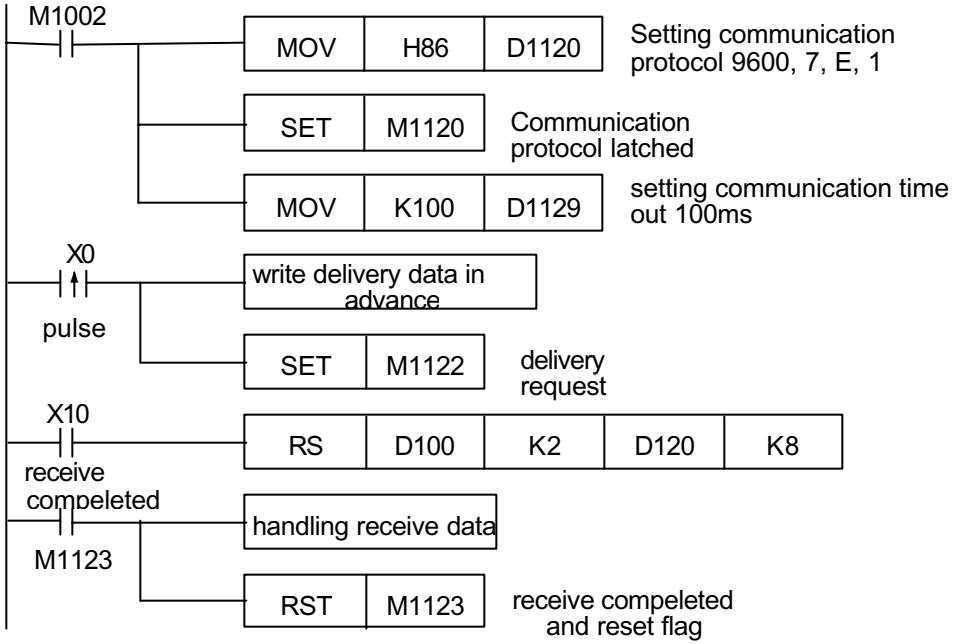
Address	01 H
Function	03 H
Start data address	21 H
	02 H
Number of data (count by word)	00 H
	02 H
CRC CHK Low	6F H
CRC CHK High	F7 H

Response:

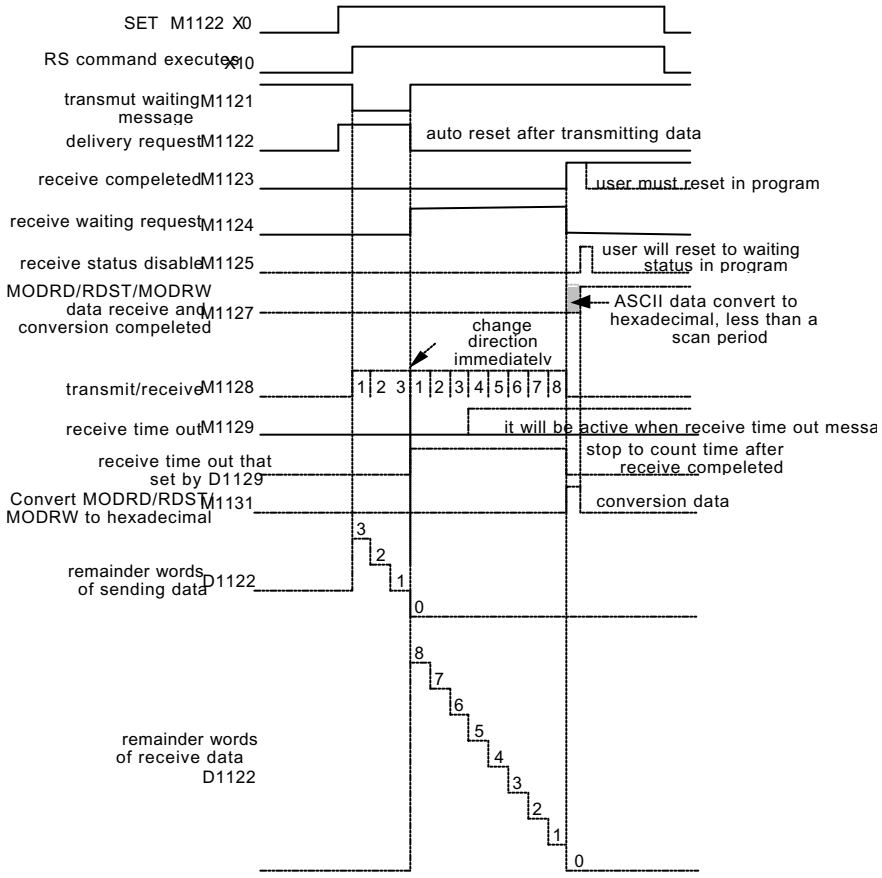
Address	01 H
Function	03 H
Number of data (count by byte)	04 H
Content of data address 8102H	17 H
	70 H
Content of data address 8103H	00 H
	00 H
CRC CHK Low	FE H
CRC CHK High	5C H

1. Timing chart of RS-485 communication program flag:

Timing chart:



Time chart:



82 ASCI (S) (D) (n) Converts HEX into ASCII

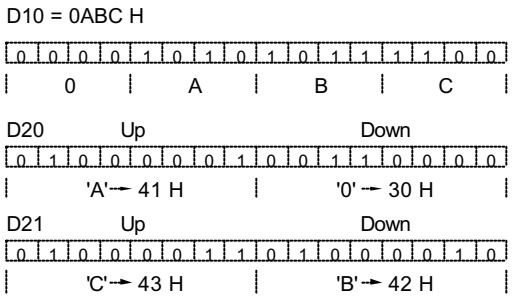
Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎					◎	◎	◎		
(D)											◎	◎	◎		
(n)	1~256														

- Converts the data of lower "n" digits (one digit consists of four bits) of 16-bit data that begins with the device (S) into the ASCII code and stores the result into four point devices that begin with the device (D).
- **Program Example**

- 16-bit conversion (M1161 OFF)

(D10) = 0ABC H '0' = 30H '1' = 31H '5' = 35H
(D11) = 1234 H 'A' = 41H '2' = 32H '6' = 36H
(D12) = 5678 H 'B' = 42H '3' = 33H '7' = 37H
'C' = 43H '4' = 34H '8' = 38H

When n is 4, the bit structure is:



When n=1 to 9:

D * \ n	K1	K2	K3	K4	K5	K6	K7	K8	K9
D20 down	'C'	'B'	'A	'0'	'4'	'3'	'2'	'1'	'8'
D20 up		'C'	'B	'A'	'0'	'4'	'3'	'2'	'1'
D21 down			'C	'B'	'A'	'0'	'4'	'3'	'2'
D21 up				'C'	'B'	'A'	'0'	'4'	'3'
D22 down					'C'	'B'	'A'	'0'	'4'
D22 up						'C'	'B'	'A'	'0'
D23 down							'C'	'B'	'A'
D23 up								'C'	'B'
D24 down									'C'

- 8-bit conversion (M1161 ON)

(D10) = 0ABC H '0' = 30H '1' = 31H '5' = 35H
(D11) = 1234 H 'A' = 41H '2' = 32H '6' = 36H
(D12) = 5678 H 'B' = 42H '3' = 33H '7' = 37H
'C' = 43H '4' = 34H '8' = 38H

When n=2:

When n=1 to 9:

D * \ n	K1	K2	K3	K4	K5	K6	K7	K8	K9
D20	'C'	'B'	'A	'0'	'4'	'3'	'2'	'1'	'8'
D21		'C'	'B	'A'	'0'	'4'	'3'	'2'	'1'
D22			'C	'B'	'A'	'0'	'4'	'3'	'2'
D23				'C'	'B'	'A'	'0'	'4'	'3'
D24					'C'	'B'	'A'	'0'	'4'
D25						'C'	'B'	'A'	'0'
D26							'C'	'B'	'A'
D27								'C'	'B'
D28									'C'

83 HEX (S) (D) (n) Converts ASCII to HEX

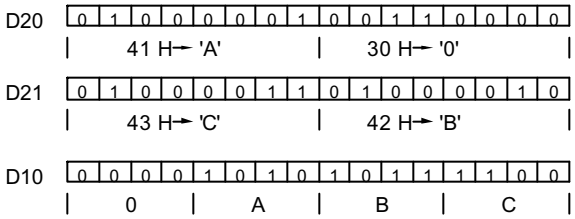
Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎					◎	◎	◎		
(D)											◎	◎	◎		
(n)	1~256														

■ Converts the data of lower "n" digits (one digit consists of four bits) of 16-bit data that begins with the device (S) into the HEX code and stores the result into four point devices that begin with the device (D).

■ **Program Example**

- 16-bit conversion (M1161 OFF)

When n=4,



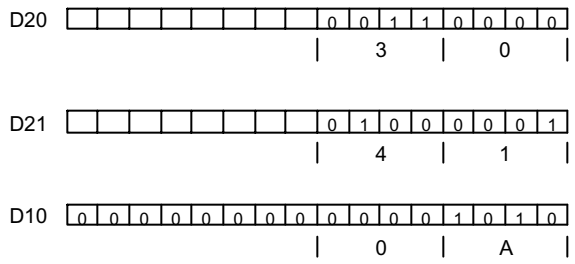
When n=1 to 9:

S *	ASCII code	HEX conversion
D10 down	30 H	0
D10 up	41 H	A
D11 down	42 H	B
D11 up	43 H	C
D12 down	31 H	1
D12 up	32 H	2
D13 down	33 H	3
D13 up	34 H	4
D14 down	35 H	5

n \ D *	D22	D21	D20
1			. . . 0 H
2			. . 0 A H
3			. 0 A B H
4			0 A B C H
5		. . . 0 H	A B C 1 H
6		. . 0 A H	B C 1 2 H
7		. 0 A B H	C 1 2 3 H
8		0 A B C H	1 2 3 4 H
9	. . . 0 H	A B C 1 H	2 3 4 5 H

- 8-bit conversion (M1161 ON)

When n=2:



When n=1 to 9:

S *	ASCII Code	HEX Conversion
D10	30 H	0
D11	41 H	A
D12	42 H	B
D13	43 H	C
D14	31 H	1
D15	32 H	2
D16	33 H	3
D17	34 H	4
D18	35 H	5

n \ D *	D22	D21	D20
1			. . . 0 H
2			. . 0 A H
3			. 0 A B H
4			0 A B C H
5		. . . 0 H	A B C 1 H
6		. . 0 A H	B C 1 2 H
7		. 0 A B H	C 1 2 3 H
8		0 A B C H	1 2 3 4 H
9	. . . 0 H	A B C 1 H	2 3 4 5 H

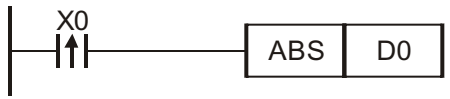
D 87 ABS (D) Absolute value

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(D)								◎	◎	◎	◎	◎	◎	◎	◎

■ When the command is executed, take the absolute value of the assigned device, (D).

■ It is generally advised to use the pulse wave contact to drive this command, otherwise, it will keep going on and on.

■ **Program Example**



◎ When X0 goes from OFF→ON, take the absolute value of the D0 contents.

88 PID (S1) (S2) (S3) (D) PID calculation

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)													◎		
(S2)													◎		
(S3)													◎		
(D)													◎		

Note: S3 operand occupies continuous 6 devices. Please refer to function specification charts for usage range of each device. Please refer to footnote for the use time of PID command.

■ (S1): target value(SV). (S2): present value(PV). (S3): parameter. (D): output value(MV).

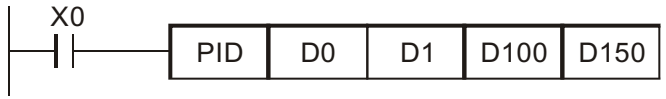
■ Specific command for PID operation control. This scan will execute PID operation when sampling time reaches. PID means Proportion, Integration and Differential. PID control has wide application on mechine equipment, pneumatic equipment and electric eqipment.

■ (S1): target value (SV), (S2): present value (PV), (S3)~(S3) +5: it will start to execute PID command after finishing all parameters setting and save the result to (D). Please give no latch register area for (D) content. (if you want to give (D) content a latch register, please clear latch to 0 when program runs.)

■ **Program Example**

◎ Please finish parameters setting before executing PID command.

◎ The command will be executed when X0=ON and the result will be saved in D150. The command won't be executed when X0=OFF and the previous data won't have any change.



■ **Footnote:**

◎ PID command is only in version V4.9 or above of ES / EX / SS series.

◎ PID command can be used one time in ES / EX / SS / EP series.

◎ It is unlimited for using times of PID command of EH series. But the register number that S3 indicates can't repeat.

◎ (S3) has 6 registers. In above program, the parameter setting area of PID command that (S3) indicates are D100~D105. You should use MOV command to transmit settings to the indication register to set before PID command executes. If the registers that parameters indicate are latch area, please execute MOVP to execute transmitting.

◎ Parameters setting is as follows.

Device No.	Function	Setting range	Explanation
(S3) :	Sampling time (T_S) (unit: 10ms)	1~2,000	If T_S is small than a scan time, PID command will execute a scan time. If T_S=0, it won't act.
(S3) +1:	Propotion gain (K_P)	0~100	
(S3) +2:	Integration gain (K_I)	0~100	

S3 +3:	Differential gain（K_D）	0~100	
S3 +4:	Action direction（Dir）	0: auto control direction 1: forward action (SV→PV) 2: reserve action (PV→SV)	
S3 +5:	deviation (E) range	0~100,	For example: if the range of deviation (E) is 5, output value MV of E between is 0.

◎ If parameter setting exceeds range, the setting will use Max. and Min. value to be settings.

◎ PID commands can be used in interrupt subroutine, step point and CJ command.

◎ Max. range of sampling time T_S is -（a scan time+1ms）~+（a scan time）. If error value has influence on output, please keep the time fixable or execute PID command in interrupt subroutine.

◎ If the settings of sampling time $T_S \leqq$ a scan time, CPU will have error code K6740(PID operation error). At this time, CPU will reset T_S = a scan time to execute PID operand. In this situation, please execute PID command in time interrupt subroutine (I6□□~I8□□).

◎ PV of PID must be a stable value before PID executes operation. Please take note of A/D of these modules conversion time if using input value of DVP-04AD / DVP-04XA / DVP-04PT / DVP-04TC module to do PID operation.

■ Calculation formula for PID command

◎ This command executes PID operation according to speed and test value differential type.

◎ PID operation has two operations, forward and reverse operation. The direction of operation is set by S3 +4. Besides, the settings that have relation to PID operation is set by S3 ~ S3 +5.

◎ Basic operation of PID

Move direction	Calcultaion method of PID
Forward operation automatically	$MV = K_P * E(t) + K_D * PV(t)S + K_I * E(t)\frac{1}{S}$ $E(t) = SV - PV$
Reverse operation	$MV = K_P * E(t) + K_D * PV(t)S + K_I * E(t)\frac{1}{S}$ $E(t) = PV - SV$

Symbols explanations:

MV : Output value

K_P : Porprotion gain

$E(t)$: Deviation value. Forward operation $E(t) = PV - SV$, reverse operation $E(t) = PV - SV$

PV : Test value

SV : Target value

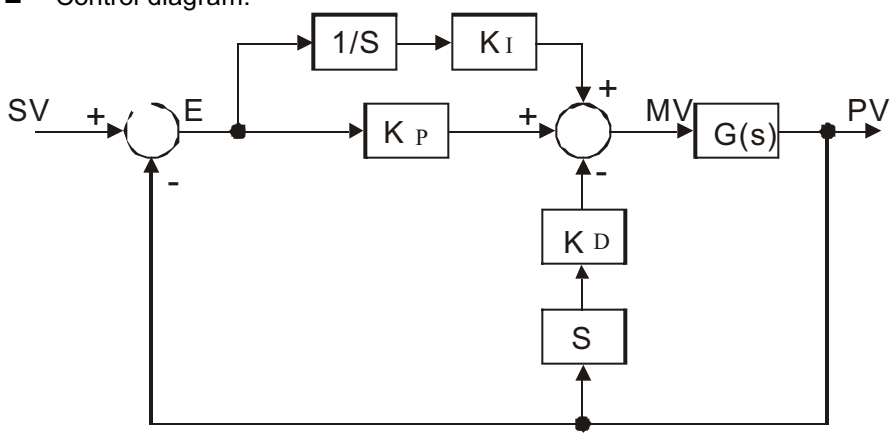
K_D : Differential gain

$PV(t)S$: Differential of $PV(t)$

K_I : Integration gain

$E(t)\frac{1}{S}$: Integration value of $E(t)$

■ Control diagram:



100 MODRD (S1) (S2) (n) MODBUS data read

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(n)					◎	◎									

■ MODRD is a command for the MODBUS ASCII mode communication. (Version 3.3 and above contain RTU mode, controlled by M1143). The Delta VFD series drives have build-in MODBUS communication. Please refer to the Delta VFD Series Manual for more details.

■ (S1) Communication address: K00000~K00254.

■ (S2) Read out address. If the address setting is illegal, the user will be informed by an error message. The error code will be saved in D1130, at the same time, M1141 will turn ON. For example, 4000H is an illegal address to VFD-S, M1141 will turn ON, D1130=2. Refer to Delta VFD-S series AC drive manual on fault information.

■ (n) Data length,n≦6.

■ The feedback data from peripherial equipment will be saved in D1070 to D1085. PLC will check the data after SAVE function is complete. If there is an error, then M1140 will be ON.

■ Because the feedback data are all ASCII characters, PLC will convert the feedback data to value data and store them in D1050 to D1055.

101 MODWR (S1) (S2) (n) MODBUS Data write

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(n)					◎	◎									

■ MODWR is a command for the MODBUS ASCII mode communication. (Version 3.3 and above contain RTU mode, controlled by M1143).

■ (S1) Communication address: K00000 to K00254.

■ (S2) Write address. If the address setting is illegal, the user will be informed by an error message. The error code will be saved in D1130, at the same time, M1141 will ON.

■ (n) Write data.

■ The feedback data from perpherial equipment will be saved in D1070 to D1076. PLC will check the data after the SAVE function is complete. If there is an error, M1140 will be ON.

102 FWD (S1) (S2) (n) VFD-A series drive forward command

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(n)					◎	◎									

103 REV (S1) (S2) (n) VFD-A series drive reverse command

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(n)					◎	◎									

104 STOP (S1) (S2) (n) VFD-A series drive stop command

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(n)					◎	◎									

- FWD/REV/STOP are communication commands for Delta A/H series drive, make sure to use the communication overtime setting (D1129) when applying these commands.
- (S1) Communication address: K00000~K00031.
- (S2) ACdrive master frequency setting for VFD-A series: setting of K0000 to K4000 represents 0.0Hz to 400.0Hz. For H series AC drive, the setting of K0000 to K1500 represent 0Hz to 1500Hz.
- (n) command object, n=1 is for one drive. n=2 communicates to all drives connected.
- The feedback data from perpherial equipment will be saved in D1070 to D1080. PLC will check the data after the SAVE function is complete. If there is an error, M1142 will be ON.

105 RDST (S1) (n) VFD-A series drive status read

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(n)					◎	◎									

- RDST is a read status instruction used with the VFD-A series drive.
- (S1), Communication address: K00000 to K00031.
- (n) Status object.

  n = 0 Frequency command
  n = 1 Output frequency
  n = 2 Output current
  n = 3 Operation command

There are 11 words in the feedback command message saved in the low byte of address D1070 to D1080.

106 RSTEF (S1) (n) VFD-A series drive reset command

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(n)					◎	◎									

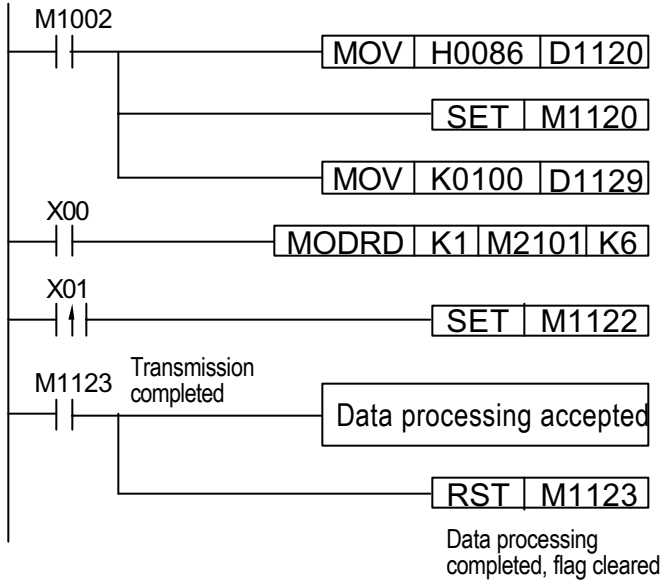
- RSTEF is a drive reset instruction for the VFD-A series drive.
- (S1) Communication address: K00000 to K0031.
- (n) Command object, n=1 is for one drive, n=2 communicates to all drives connected.
- The feedback data from perpherial equipment will be saved in D1070 to D1089. If n=2, PLC will not receive any data.
- Communication example:

1. Connect RS-485 communication between the Delta VFD-S and PLC.

2. Preset the following VFD-S series parameters.

Parameter	Setting Value	Explanations
2-00	4	Master frequency determined by RS-485
2-01	3	Operation command determined by RS-485
9-00	1	Communication address is 01
9-01	1	Transmission speed (baud rate) : 9600 bps
9-04	1	Communication protocol ASCII mode: 7 data bits, Even parity, 1 stop bit (7, E, 1)

3. Use DVP programming tools to input the following program.



4. After the PLC executes a RUN command, input point X00 remains ON, and the input point X01 changes from OFF to ON, PLC will transfer the command MODRD K1 M2101 to the VFD-S series AC drive, and the data will be saved in D1089 to D1095.

5. When PLC receives feedback data, the data will be placed in D1070 to D1076, the ASCII codes will be converted into HEX, and saved in D1050~D1055. Refer to the following example:

PLC ⇨ VFD-S
“010321010006D4”

VFD-S ⇨ PLC
“01 03 0C 0100 1766 0000 0000 0136 0000 3B”

Command Messages :			
D1089 down	‘0’	30 H	ADR 1
D1089 up	‘1’	31 H	ADR 0
D1090 down	‘0’	30 H	CMD 1
D1090 up	‘3’	33 H	CMD 0
D1091 down	‘2’	32 H	Starting Address
D1091 up	‘1’	31 H	
D1092 down	‘0’	30 H	
D1092 up	‘1’	31 H	
D1093 down	‘0’	30 H	Date (Words)
D1093 up	‘0’	30 H	
D1094 down	‘0’	30 H	
D1094 up	‘6’	36 H	
D1095 down	‘D’	44 H	LRC CHK 1
D1095 up	‘4’	34 H	LRC CHK 0

Feedback Messages:				
D1070 down	‘0’	30 H	ADR 1	
D1070 up	‘1’	31 H	ADR 0	
D1071 down	‘0’	30 H	CMD 1	
D1071 up	‘3’	33 H	CMD 0	
D1072 down	‘0’	30 H	Date (Words)	
D1072 up	‘C’	43 H		
D1073 down	‘0’	30 H	Content of address 2101H	PLC will automatically convert ASCII codes and save in D1050 = 0100H
D1073 up	‘1’	31 H		
D1074 down	‘0’	30 H		
D1074 up	‘0’	30 H		
D1075 down	‘1’	31 H	Content of address 2102H	PLC will automatically convert ASCII codes and save in D1051 = 1766H
D1075 up	‘7’	37 H		
D1076 down	‘6’	36 H		
D1076 up	‘6’	36 H		

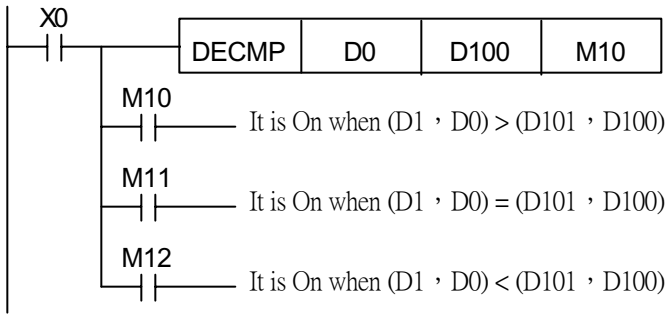
※ ADR (1,0): AC drive
※ CMD (1,0): Command code
※ LRC CHK (0,1): check fault code. Refer to the communication parameters of Delta AC drive user manual for more details.

Feedback:				
D1077 down	‘0’	30 H	Content of address 2103H	PLC will automatically convert ASCII codes and save in D1052 = 0000H
D1077 up	‘0’	30 H		
D1078 down	‘0’	30 H		
D1078 up	‘0’	30 H		
D1079 down	‘0’	30 H	Content of address 2104H	PLC will automatically convert ASCII codes and save in D1053 = 0000H
D1079 up	‘0’	30 H		
D1080 down	‘0’	30 H		
D1080 up	‘0’	30 H		
D1081 down	‘0’	30 H	Content of address 2105H	PLC will automatically convert ASCII codes and save in D1054 = 0136H
D1081 up	‘1’	31 H		
D1082 down	‘3’	33 H		
D1082 up	‘6’	36 H		
D1083 down	‘0’	30 H	Content of address 2106H	PLC will automatically convert ASCII codes and save in D1055 = 0000H
D1083 up	‘0’	30 H		
D1084 down	‘0’	30 H		
D1084 up	‘0’	30 H		
D1085 down	‘3’	33 H	LRC CHK 1	
D1085 up	‘B’	42 H	LRC CHK 0	

D **110** ECMP (S1) (S2) (D) Comparison of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(D)		◎	◎	◎											

- (S1): the comparison value 1 of decimal of binary system. (S2): the comparison value 2 of decimal of binary system. (D): comparison result, occupies continuous 3 points.
- The comparison result (>, =, <) of value 1 of decimal of binary system and value 2 of decimal of binary system will be showed in (D).
- If the source operand (S1) or (S2) designates constant K or H, command will convert the constant to decimal of binary system to compare.
- If designated device is M10, it will occupy M10~M12.
- **Program Example**



◎ When X0=On and execute DECMP command, one of M10~M12 will be On. When X0=Off and not to execute DECMP command, the state of M10~M12 will be in the state before X0= Off.

- If you need to get the result of ≧, ≦, ≠, you could get by series connection or parellet connection of M10~M12.
- If you want to clear the result, please use RST or ZRST command.
- Please refer to page 7-4 Handling of Decimal for detail.

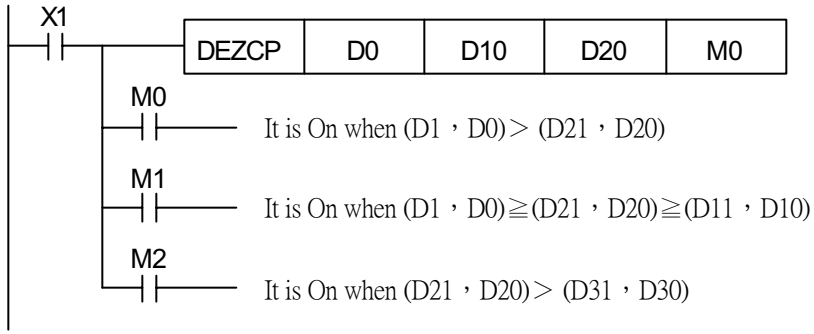
D **111** EZCP (S1) (S2) (D) (D) Comparison of the area of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(D)					◎	◎							◎		
(D)		◎	◎	◎											

- (S1): lower bound of decimal of binary of area comparison. (S2): upper bound of decimal of binary of area comparison. (S): comparison value of decimal of binary system. (D): comparison result, it will occupy continuous 3 points.
- The compared result of (S), (S1) and (S2) will be saved in (D).
- If source operand (S1) or (S2) designates constant K or H, the command will convert the constant to decimal of binary system to compare.
- When (S1) > (S2), this command will use (S1) to be upper bound

and lower bound for comparison.

- If designated device is M0, it will auto occupy M0~ M2.
- **Program Example**



◎ When X1=On and DEZCP command is executed, one of M0~M2 will be On. When X0=Off and ZCP command is not executed, the state of M0~M2 will be in the state before X1=Off.

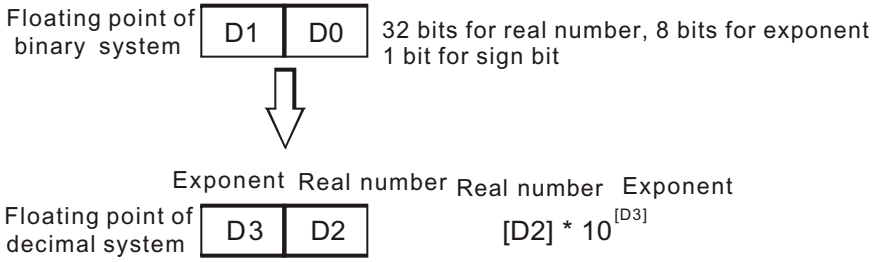
- If you want to clear the result, please use RST or ZRST command.
- Please refer to Page 7-4 Handling of Decimal for detail.

D	118	EBCD (S) (D)	Decimal of binary number→ decimal of decimal system

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

- (S): data resource. (D): the result of exchange.
- To convert the value of register that (S) designates from decimal of binary system to decimal of decimal system to save in the register that (D) designates.
- The PLC decimal is operated by decimal of binary system. The DEBCD command is the specific command for converting from decimal of binary system to decimal of decimal system.
- **Program Example**

◎ When X0=On, the decimal of binary system in D1, D0 will be converted to decimal of decimal system to save in D3, D2.



- Please refer to Page 7-4 Handling of Decimal for detail.

D	119	EBIN (S) (D)	Decimal of decimal system →decimal of binary system

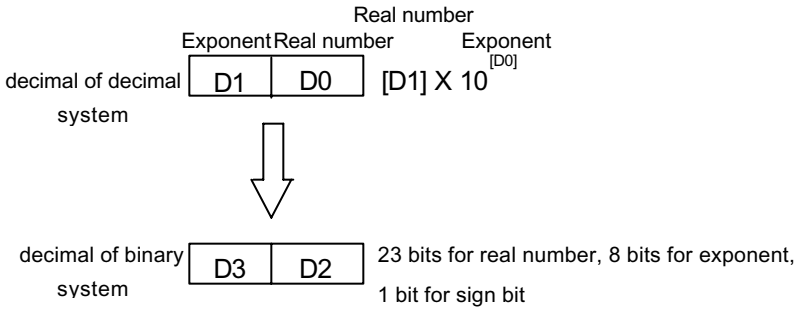
Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

- (S): data resource. (D): the exchange result.
- To convert the value of decimal of decimal system in the register that

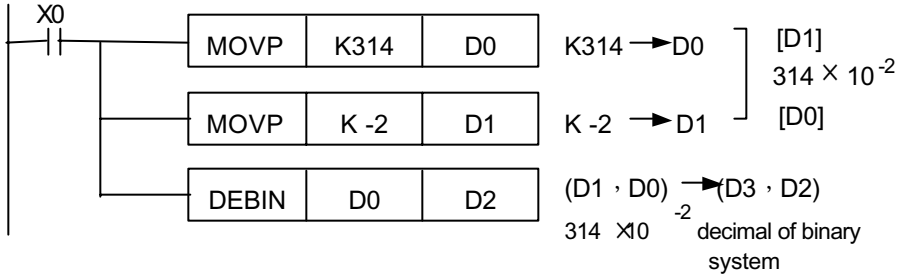
(S) designates to decimal of binary system and save the result in the register that (D) designates.

- DEBIN command is the specific command that used to convert the value from decimal of decimal system to decimal of binary system.
- **Program Example**

◎ When X1=On, the decimal of decimal system in D1, D0 is converted to decimal of binary system to save in D3, D2.



- Before doing decimal operation, you should use FLT（API 49）BIN integer to convert to decimal of binary system. The value that is exchanged must be BIN integer. However, DEBIN command can convert decimal to decimal of binary system.
- **Program Example**



◎ When X0=On, move K314 to D0 and move K-2 to D1 to make up decimal of decimal system ($3.14 = 314 \times 10^{-2}$).

- Please refer to page 7-4 decimal handing for detail.

D	120	EADD (S1) (S2) (D)	Addition of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(D)													◎		

- (S1): augend. (S2): addend. (D): sum.
- The content of register that (S1) designates adds the content of register that (S2) indicates and save the sum in the register that (D) designates. The all process of addition operation uses decimal of binary system.
- If source operand (S1) or (S2) designates constant K or H, the command will convert the constant to decimal of binary system for addition operation.

■ (S1) and (S2) can designate the same number register. In this situation, when using "continuous" command the register will be added one time in the every scan during the condition contact is On. In general, it uses pulse execution command. (DEADDP).

■ **Program Example**

◎ When X0=On, add the decimal of binary system (D1, D0) and the decimal of binary system (D3, D2) and save the sum in (D11, D10).

■ **Program Example**

■ When X2=On, add the decimal of binary system (D11, D10) and K1234 (auto convert to decimal of binary system) and save the sum in (D21, D20).

■ Please refer to page 7-4 Handling of Decimal for detail.

D	121	ESUB (S1) (S2) (D)	Subtraction of decimal of binary system

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(D)													◎		

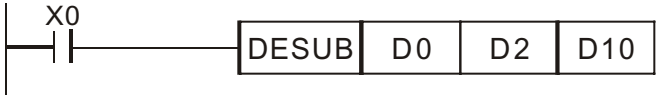
■ (S1): minuend. (S2): subtrahend. (D): difference.

■ Using the content of register that (S1) designates minus the content of register that (S2) designates and save the result in the register that (D) designates. All process of subtraction uses the type of decimal of binary system.

■ If the source operand (S1) or (S2) designates the constant K or H, the command will be convert to decimal of binary system to subtract.

■ (S1) and (S2) can designate the same number register. In this situation, when using "continuous" command the register will be added one time in the every scan during the condition contact is On. in general, it uses pulse execution command (DESUBP).

■ **Program Example**



◎ When X0=On, decimal of binary system (D1, D0) minus decimal of binary system (D3, D2) and save the result to (D11, D10).

■ **Program Example**

◎ When X2=On, K1,234 (auto convert to decimal of binary system) minus the decimal of binary system (D1, D0) and save the result to (D11, D10).

■ Please refer to page 7-4 Handling of Decimal for detail.

D **122** EMUL (S1) (S2) (D) Multiplication of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(D)													◎		

■ (S1): multiplicand. (S2): multiplicator. (D): product of multiplication.

■ The content of register that (S1) designates multiplied by the content of register that (S2) designates and save the result in the register that (D) designates. All process of multiplication operation uses decimal of binary system.

■ If source operand (S1) or (S2) designates the constant K or H, the command will convert the constant to decimal of binary system.

■ (S1) and (S2) can designate the same number register. In this situation, when using "continuous" command the register will be added one time in the every scan during the condition contact is On. It uses pulse execution command in general (DEMULP).

■ **Program Example**

◎ When X0=On, the decimal of binary system (D1,D0) multiplies the decimal of binary system (D11,D10) and save the result in the register that (D21,D20) designates.

■ **Program Example**

◎ When X2=On, K1,234(auto convert to decimal of binary system)×the decimal of binary system (D1, D0) and save the result in (D11, D10).

■ Please refer to page 7-4 Handling of Decimal for detail.

D **123** EDIV (S1) (S2) (D) Division of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(D)													◎		

■ (S1): dividend. (S2): divisor. (D): quotient and remainder.

■ The content of register that (S1) designates divided by the content of register that (S2) designates and save the result in the register that (D) designates. All process of division operation uses decimal of binary system.

■ If source operand (S1) or (S2) designates the constant K or H, the command will convert the constant to decimalof binary system.

■ If the content of divisor (S2) is 0, it will be regarded as "operand error" and this command won' t be executed.

■ **Program Example**

◎ When X1=On, the decimal of binary system (D1,D0) divided by the decimal of binary system (D11,D10) and save the remainder in (D21,D20).

■ **Program Example**

◎ When X2=On, the decimal of binary system (D1, D0) ÷ K1234 (auto convert to decimal of binary system) and save the result in (D11, D10).

■ Please refer to page 7-4 Handling of Decimal for detail.

D	127	ESQR (S) (D)	Square root of decimal of binary system

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)					◎	◎							◎		
(D)													◎		

■ (S): the source device for getting the square root. (D): the result of the square root.

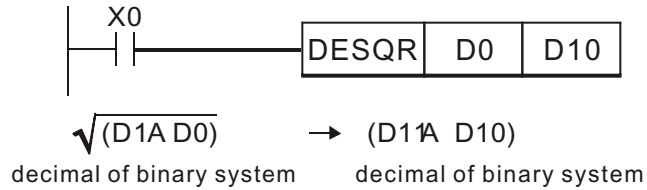
■ Getting the square root of the register that (S) designates and save the result in the rgister that (D) designates. All process uses decimal of binary system.

■ If the source operand (S1) or (S2) designates the constant K or H, the command will convert the constant to decimal of binary system.

If the result of square root is 0, flag M1020=On

■ Source operand is valid when the value is positive. If the value is negative, it will be regarded as "operand error", the command won't be executed and flag M1067=On.

■ **Program Example**



◎ When X0=On, get the square root of decimal of binary system (D1,D0) to save in the register that (D11,D10) designates.

■ **Program Example**

◎ When X2=On, get the square root of K1,234 (auto convert to decimal of binary system) and save the result in (D11, D10).

■ Please refer to page 7-4 Handling of Decimal for detail.

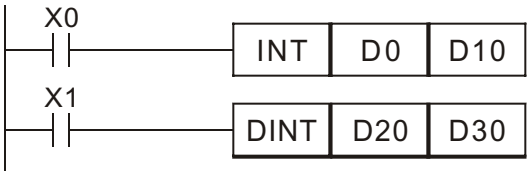
D	129	INT (S) (D)	Decimal of binary system→BIN integer

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

■ (S): the source device for converting. (D): conversion result.

■ The register that (S) designates convert from decimal of binary system to BIN integer and save in the register that (D) designates. The decimal of Bin integer will be discarded.

- The function of this command is opposite to API 49 (FLT).
- If the result after converting is 0, zero flag M1020=On.
  If there is any decimal discarded, M1021=On.
  If the result exceeds the following range, M1022=On.
  16-bit command: -32,768~32,767
  32-bit command: -2,147,483,648~2,147,483,647
- **Program Example**



◎ When X0=On, the decimal of binary system (D1, D0) will convert to BIN integer and save the result in (D10). The decimal of BIN integer will be discarded.

◎ When X1=On, the decimal of binary system (D21, D20) will convert to BIN integer and save the result in (D31, D30). The decimal of BIN integer will be discarded.

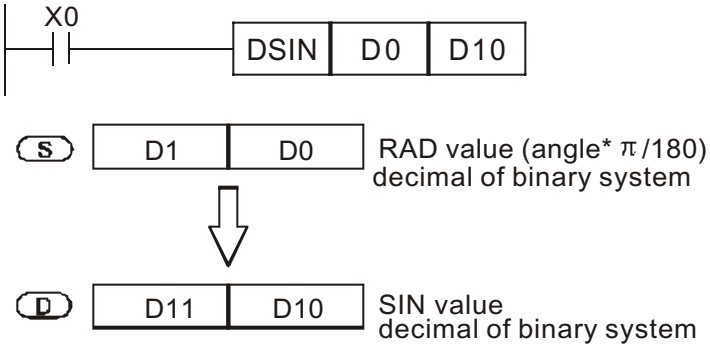
- Please refer to page 7-4 Handling of Decimal for detail.

D 130	SIN (S) (D)	SIN operation of decimal of binary system

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

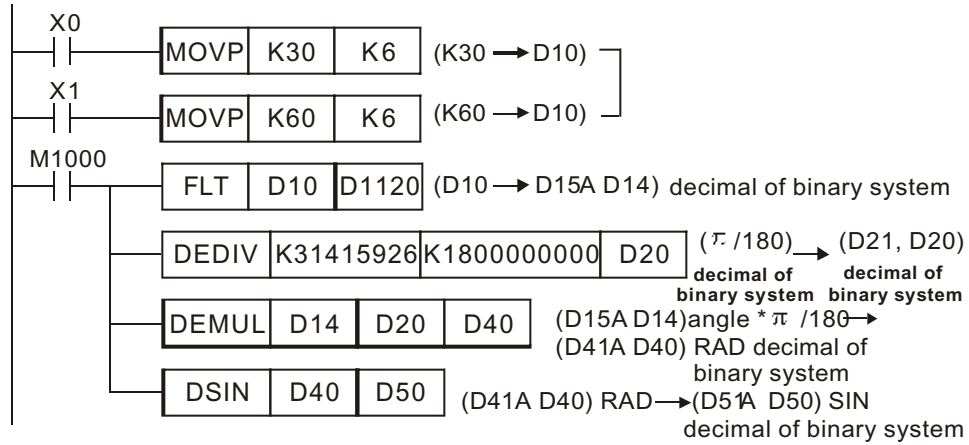
- (S): designated RAD value. (D): the result after converting to SIN.
- RAD value that (S) designates = angle $\times\pi$/180. Save the result of converting to SIN value in the register that (D) designates.

- **Program Example**



◎ When X0=On, save the result after converting the RAD value of decimal of binary system of (D1, D0) to SIN value in (D11, D10). The content is decimal of binary system.

- Selecting angle from input terminal X0 and X1 and convert it to RAD value. Then convert to SIN value.

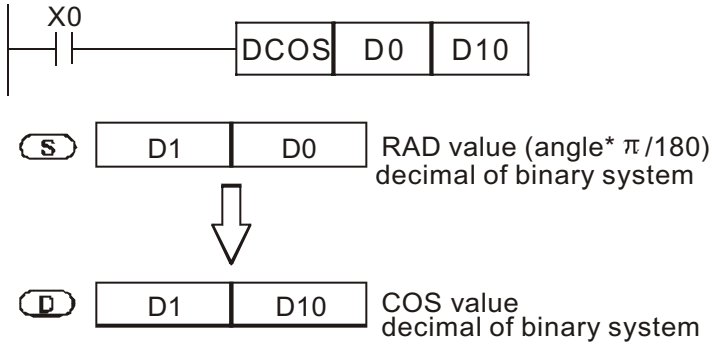


- Please refer to page 7-4 Handling of Decimal for detail.

D 131 COS (S) (D) COS operation of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

- (S): designated RAD. (D): the result of COS value.
- RAD value that (S) designates = angle × π /180. To get COS value and save in the register that (D) designates.
- **Program Example**



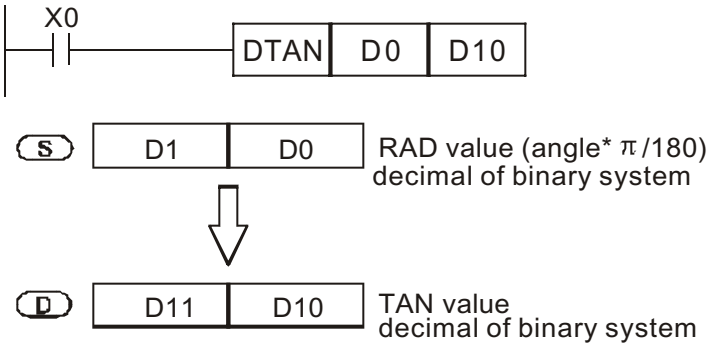
◎ When X0=On, Get the COS value of RAD value of decimal of binary system (D1, D0) and save it in (D11, D10). The content is decimal of binary system.

- Please refer to page 7-4 Handling of Decimal for detail.

D 132 TAN (S) (D) TAN operation of decimal of binary system

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

- (S): designated RAD value. (D): the result of TAN value.
- RAD value that (S) designates = angle × π /180. Get TAN value and save the result in the register that (D) designates.
- **Program Example**



◎ When X0=On, RAD value of decimal of binary system of (D1, D0) and convert to TAN value to save in (D11, D10). The content is decimal of binary system.

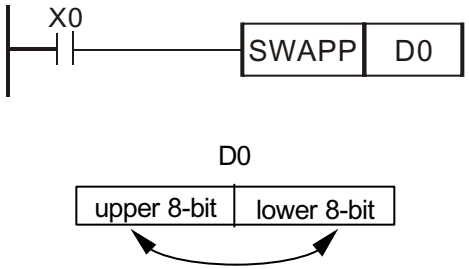
- Please refer to page 7-4 Handling of Decimal for detail.

147 **SWAP** (S) Swap upper and lower 8-bit

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)								◎	◎	◎	◎	◎	◎	◎	◎

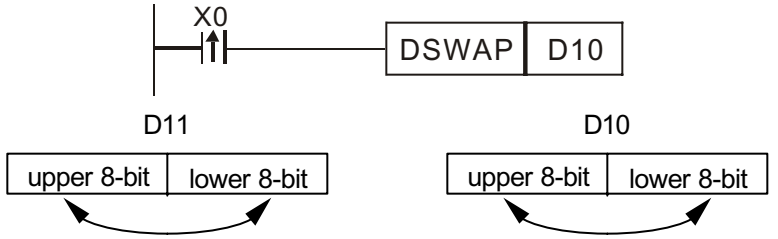
- Note: When operand D is used with equipment F, it can only use 16-bit command. Please refer to function specification charts for usage range of each device.
- (S): the equipment for swapping upper and lower 8-bit
- When being 16-bit command, swapping the content of upper and lower 8-bit.
- When being 32-bit command, swapping the content of upper and lower 8-bit of two registers separately.
- This command is usually pulse execution (SWAPP, DSWAPP).
- **Program Example 1:**

When X0=ON, swapping the content of upper and lower 8-bit of D0.



- **Program Example 2:**

When X0=ON, swapping upper 8-bit and lower 8-bit of D11 and swapping upper 8-bit and lower 8-bit of D10.



- **Footnote:**

The version V4.9 and above of ES / EX / SS series support Continuous execution command (SWAP, DSWAP).

150 **MODRW** (S1) (S2) (S3) (S4) (D) MODBUS data read/write

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎							◎		
(S2)					◎	◎							◎		
(S3)					◎	◎							◎		
(S4)													◎		
(D)					◎	◎							◎		

Note: usage range of S1 operand K0~K255. The limitation of S2 operand indication content K3(H3), K6(H6), K16(H10). The usage of n: n=K1~K16. Please refer to function specification charts for usage range of each device.

- (S1): connection device address. (S2): function code. (S3): address of being read or write. (S4): register of being read/write. (n): length of read/write data.

■ (S1): UNIT ADDRESS. The usage range is K0~K255.

■ (S2): FUNCTION CODE. For example: the command of AC drive or DVP-PLC to read many items is H03. Write command of AC drive or DVP-PLC is H06 and the command of write many items is H10.

■ (S3): device address that being read/write data, inner device address of connection device. If address is illegal to the assigned equipment, there will be fault code save in D1130 and at the same time, M1141 will be ON. For example, 4000H is illegal to VFD-S, M1141 will be ON and D1130 = 2. Please refer to VFD-S for fault code.

■ (S3): device address of being read/write

■ (S4): source or destination of being read/write. User can set register to write data length in advance or save data after reading.

■ (n): read/write data length. Assigned range K1~K16(WORD).

■ **Program Example 1:**

◎ Function code K3(H3): read many items data.

1. PLC connects to VFD-S AC drive. (ASCII Mode when M1143=OFF)

2. PLC connects to VFD-S AC drive. (ASCII Mode when M1143=ON)

◎ Receiving data saves in 16 continuous registers that start from D0 with ASCII form when in ASCII mode. PLC will convert the content to Hexadecimal and save into registers D1296~D1311 automatically. M1131=ON when it starts converting to hexadecimal and M1131 will be OFF after finishing converting.

◎ User can MOV, DMOV or BMOV commands to move D1296~D1311 that save hexadecimal data to general register to use. Other command is invalid to this area.

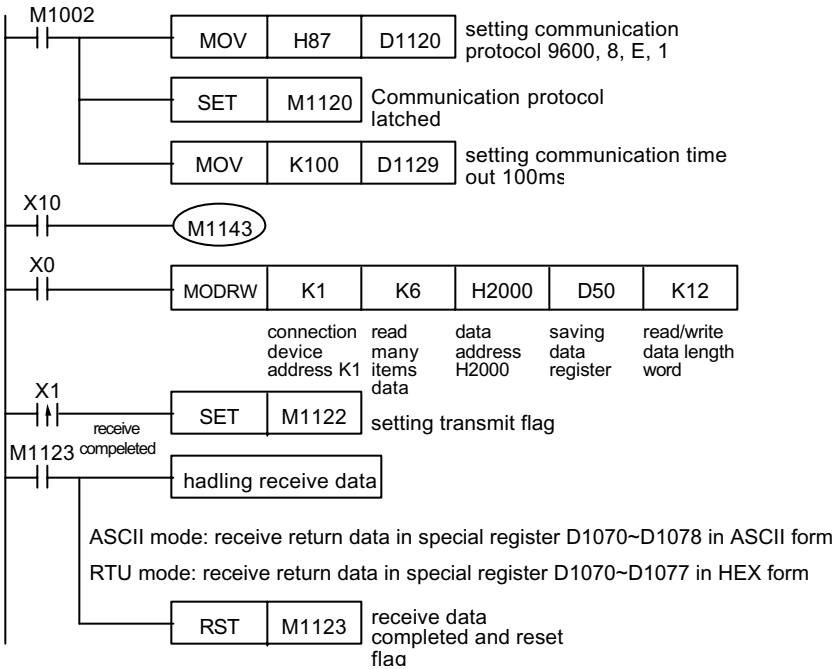
◎ Received data saves in the 16 continuous registers that starts from D0 and designated by users in hexadecimal type in RTU mode. At the same time, D1296~D1311 is invalid.

◎ In ASCII mode or RTU mode, PLC will save the transmission data in D1256~D1295. Users can move these register data to general register by MOV, DMOV or BMOV commands. Other commands are invalid to this area.

◎ Data, return from AC drive, is saved in registers that designate by users. After finishing, PLC will check if the received data is correct automatically. If having faults, M1140 will be set to ON.

◎ Inner data address of AC drive. If address is illegal to assigned equipment, it will have fault code. Fault code will be saved in D1130 and M1141 will be on. For example, 4000H is illegal to VFD-S and M1141=ON and D1130=2. Please refer to VFD-S user manual to fault code.

◎ After M1140= ON or M1141=ON, it will transmit a correct data to AC drive. If return data is correct, M1140 and M1141 will be reset.



■ **Program Example 2:**

◎ Function code K6(H6): write a WORD to register

A. PLC connects to VFD-S AC drive. (ASCII Mode when M1143=OFF)

B. PLC connects to VFD-S AC drive. (ASCII Mode when M1143=ON)

◎ When in ASCII mode, users save data that will be wrote to AC drive in ASCII form in assigned register D0. Data that return from AC drive will be saved in registers D1070~D1076.

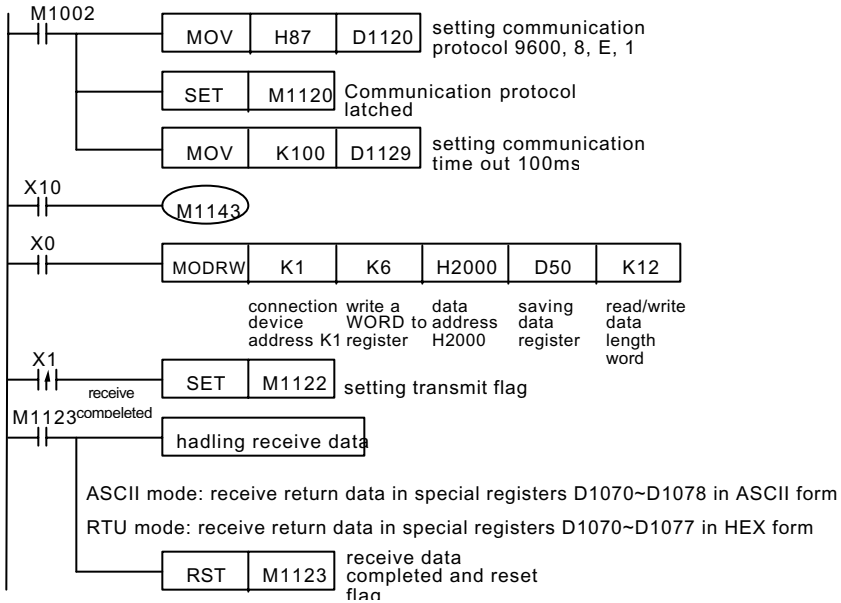
◎ When in RTU mode, users save data that will be wrote to AC drive in HEX form in assigned register D0. Data that return from AC drive will be saved in register D1070~D1076.

◎ When in ASCII mode or RTU mode, PLC will save data that will transmit in transmission registers D1256~D1295. Users can move these data to general registers by using MOV, DMOV or BMOV commands.

◎ After receiving return data from AC drive, PLC will check the receiving data automatically. If having fault, M1140 will be ON.

◎ Inner data address of AC drive. If address is illegal to assigned equipment, it will have fault code. Fault code will be saved in D1130 and M1141 will be ON. For example, 4000H is illegal to VFD-S, M1141 will be ON and D1130=2. Please refer to VFD-S user manual for detail.

◎ After M1140 is ON or M1141 is ON, it will transmit a correct data to AC drive. If return data is correct, M1140 and M1141 will be reset.



■ **Program Example 3:**

◎ Function code K16(H10): write many WORD to register

A. PLC connects to VFD-S AC drive (when M1143=OFF, ASCII Mode)

B. PLC connects to VFD-S AC drive (when M1143=ON, RTU Mode)

◎ When in ASCII mode, users will save data that being wrote to AC drive in 12 continuous registers that start from D0 and designated by user in ASCII form. Data that AC drive return will save in registers D1070~D1076.

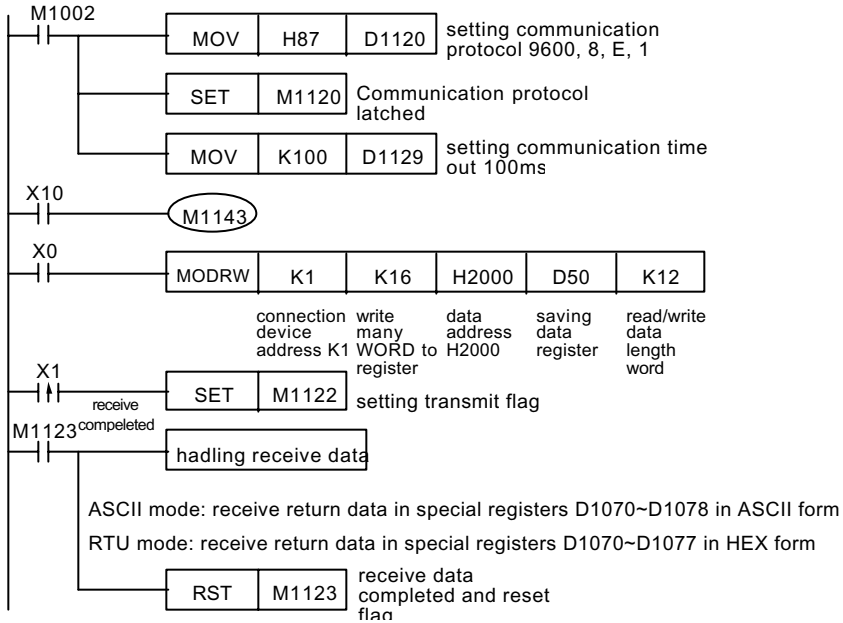
◎ When in RTU mode, users will save data that being wrote to AC drive in 12 continuous registers that start from D0 and designated by user in HEX form. Data that AC drive return will save in registers D1070~D1078.

◎ When in ASCII mode or RTU mode, PLC will save data that being transmitted in registers D1256~D1295. Users can move these data to general registers by using MOV, DMOV or BMOV commands. Other commands are invalid to this area.

◎ After receiving data that return from AC drive, PLC will check it. If there is fault, M1140 will be ON.

◎ Inner data address of AC drive. If address is illegal to assigned equipment, it will have fault code. Fault code will be saved in D1130 and M1141 will be on. For example, 4000H is illegal to VFD-S, M1141 is ON and D1130=2. Please refer to VFD-S user manual for detail.

◎ After M1140 is ON or M1141 is ON, it will transmit a correct data to AC drive. If return data is correct, M1140 and M1141 will be reset.



■ **Footnote:**

1. V4.9 and above of ES / EX / SS series have this command MODRW.
2. Relative flag signal and special register of RS-485 communication MODRW command: please refer to footnote of API 80 RS command for detail.

Flag	Function explanation
M1120	Communication setting latched
M1121	Transmit waiting message
M1122	Delivery request

Flag	Function explanation
M1123	Receive completed
M1124	Receive waiting message
M1125	Receive status disable
M1126	STX/ETX system definition selection
M1127	MODRD / RDST / MODRW commands data receive completed
M1128	Transmitting/receiving indication
M1129	Receive time out
M1130	Users/system definition STX/ETX
M1131	MODRD / MODWR / MODRW data convert to HEX, M1131=ON
M1140	MODRD / MODWR / MODRW data receive error
M1141	MODRD / MODWR / MODRW command parameter error
M1142	VFD-A convenience command data receive error
M1143	ASCII / RTU mode selection, ON is RTU mode

Special register	Function Explanation
D1038	Time setting for data response delay when PLC is slave. Time unit is 0.1ms.
D1070~ D1085	It is PLC built-in RS-485 communication convenience command. This command will send messages during executing and if the receiver receives, it will return messages and save it in D1070~D1085. Users can view return data by this register content.
D1120	RS-485 communication protocol
D1121	PLC communication address
D1122	Remainder characters of delivery data

Special register	Function Explanation
D1123	Remainder characters of received data
D1124	Start text definition（STX）
D1125	Definition of the first end character（ETX1）
D1126	Definition of the second end character（ETX2）
D1129	Communication time out abnormal. Time unit:（ms）
D1130	Return fault code record of MODBUS
D1256~ D1295	This is PLC built-in RS-485 communication convenience command MODRW. The message that this command sends during executing will be saved in D1256~D1295. User can check according to this register content. (In version 4.9, you can use MOV, DMOV, BMOV to move the data in this area.
D1296~ D1311	PLC will convert ASCII saved in the register that users indicate to hexadecimal. (In version 4.9, you can use MOV, DMOV, BMOV to move the data in this area.)

D 224 ↓ 230

LD＊ (S1) (S2) The contact type comparison **LD＊**

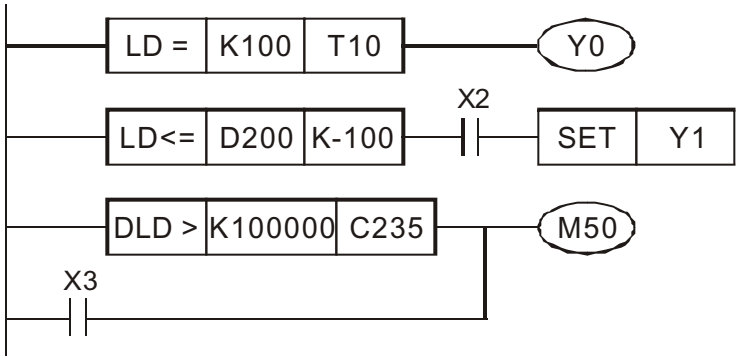
Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
＊ : =, >, <, <>, ≦, ≧															

- Compare the contents of (S1) and of (S2). To take "LD=" as an example, if the comparison result is "=", the contact is in continuity, and if it is "≠", the contact is in discontinuity. The LD＊ command could connect directly with the BUS.

- When the left most bit, MSB (the 16-bit command: b15, the 32-bit command: b31), from (S1) and (S2) is 1, this comparison value will be viewed as a negative value for comparison.
- If the 32-bit length counter (C235~) is put into this command for comparison, be sure to use the 32-bit command (DLD∗). If the 16-bit command (LD∗) is utilized, CPU will determine it as "Program Error", and the red "ERROR" indicator on the MPU panel will be blinking, and the CPU will not berunning.
- Motion Conditions of LD∗:

API No.	16-bit command	32-bit command	Continuity condition	Discontinuity condition
224	LD=	DLD=	(S1) = (S2)	(S1) ≠ (S2)
225	LD>	DLD>	(S1) > (S2)	(S1) ≦ (S2)
226	LD<	DLD<	(S1) < (S2)	(S1) ≧ (S2)
228	LD<>	DLD<>	(S1) ≠ (S2)	(S1) = (S2)
229	LD≦	DLD≦	(S1) ≦ (S2)	(S1) > (S2)
230	LD≧	DLD≧	(S1) ≧ (S2)	(S1) < (S2)

- **Program Example**



◎ If the content of counter T10 is equal to K100, Y0=ON.

◎ When the content of D200 is smaller or equal to K –100, and that X2=ON, Y1 will be set as "ON".

◎ If the content of C235 is smaller than K100,000, or when X3=ON, M50=ON.

D 232 ↓ 238

AND∗ (S1) (S2) The series connection contact type comparison AND∗

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎

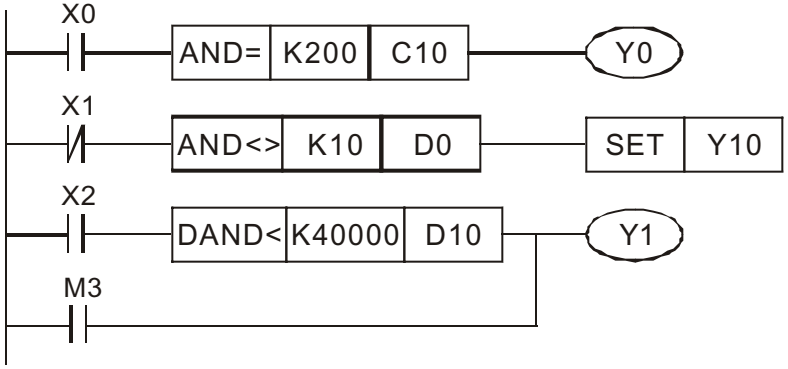
∗ : =, >, <, <>, ≦, ≧

- Compare the contents of (S1) and of (S2), To take "AND=" as an example, if the comparison result is "=", the contact is in continuity, and if it is "≠", the contact is in discontinuity. The AND∗ command is the comparison command that connects with the series connection contact.
- When the left most bit, MSB (the 16-bit command: b15, the 32-bit command: b31), from (S1) and (S2) is 1, this comparison value will be viewed as a negative value for comparison.
- If the 32-bit length counter (C235~) is put into this command for comparison, be sure to use the 32-bit command (DAND∗). Or if the 16-bit command (AND∗) is utilized, CPU will determine it as "Program Error", and the red "ERROR" indicator on the MPU panel will be blinking, and the CPU will not berunning.

■ Motion Conditions of AND∗:

API No.	16-bit command	32-bit command	Continuity condition	Discontinuity condition
232	AND=	DAND=	(S1) = (S2)	(S1) ≠ (S2)
233	AND>	DAND>	(S1) > (S2)	(S1) ≦ (S2)
234	AND<	DAND<	(S1) < (S2)	(S1) ≧ (S2)
236	AND<>	DAND<>	(S1) ≠ (S2)	(S1) = (S2)
237	AND≦	DAND≦	(S1) ≦ (S2)	(S1) > (S2)
238	AND≧	DAND≧	(S1) ≧ (S2)	(S1) < (S2)

■ **Program Example**



◎ If X0=ON and that the current value of counter C10 equals K200, Y0=ON.

◎ If X1=OFF and that the content of register D0 not equal to K –10, Y10 will be set as "ON".

◎ If X2=ON and that the contents of the 32-bit registers D11 and D10 are equal to K40,000, Y1=ON.

D 240 ↓ 246

OR∗ (S1) (S2) The parallel connection contact type comparison OR∗

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎
(S2)					◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎

∗: =, >, <, <>, ≦, ≧

■ Compare the contents of (S1) and of (S2). Take "OR=" as an example, if the comparison result is "=", the contact is in continuity, and if it is "≠", the contact is in discontinuity. The OR∗ command is the comparison command that connects with the parallel connection contact.

■ When the left most bit, MSB (the 16-bit command: b15, the 32-bit command: b31), from (S1) and (S2) is 1, this comparison value will be viewed as a negative value for comparison.

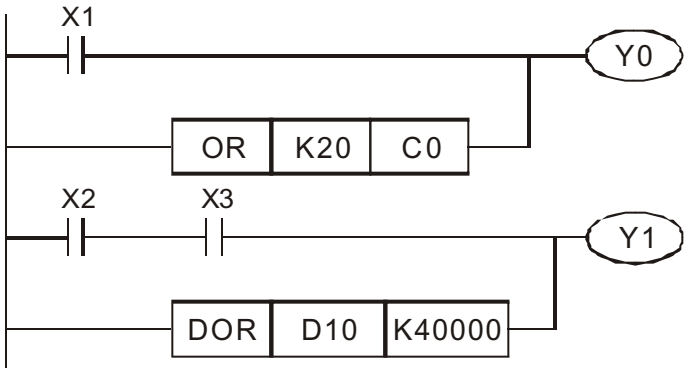
■ If the 32-bit length counter (C235~) is put into this command for comparison, be sure to use the 32-bit command (DOR∗). Or if the 16-bit command (OR∗) is utilized, CPU will determine it as "Program Error", and the red "ERROR" indicator on the MPU panel will be blinking, and the CPU will not be running.

■ Motion Conditions of OR∗:

API No.	16-bit command	32-bit command	Continuity condition	Discontinuity condition
240	OR=	DOR=	(S1) = (S2)	(S1) ≠ (S2)
241	OR>	DOR>	(S1) > (S2)	(S1) ≦ (S2)
242	OR<	DOR<	(S1) < (S2)	(S1) ≧ (S2)

API No.	16-bit command	32-bit command	Continuity condition	Discontinuity condition
244	OR< >	DOR< >	(S1) ≠ (S2)	(S1) = (S2)
245	OR≦	DOR≦	(S1) ≦ (S2)	(S1) > (S2)
246	OR≧	DOR≧	(S1) ≧ (S2)	(S1) < (S2)

■ **Program Example**



◎ If X1=ON, or that the current value of counter C0 is equal to K20, Y0=ON.

◎ If both X2 and X3 are "ON", or that the contents of the 32-bit registers D11 and D10 are greater or equal to K40,000, Y1=ON.

■ EX MPU

EX MPU is a main processing unit with 4 analog inputs and 2 analog outputs. (Refer to Chapter 2 for detailed specifications), methods to be adopted are as follows:

◎ Analog/Digital (A/D)

Analog Input:

Monotonicity with no miss code

Overall Precision:

Non-linearity: ±1% of full scale over temperature.

Maximum error: ±1% of full scale of +10V and +20mA over temperature.

Data format returned to the application program: Binary.

Value of LSB (Least Significant Bit):

Voltage input: 19.53125 mV (10V/512)

Current input: 39.0625 μA (20mA/512)

Input mode: differential

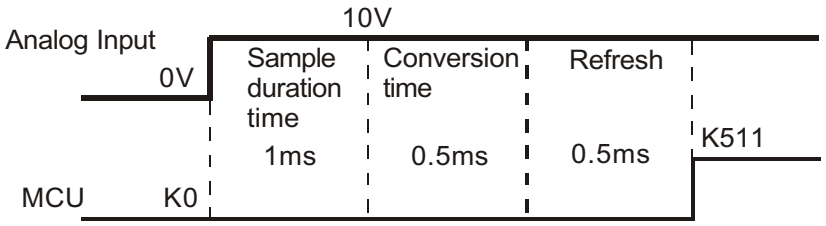
Common mode characteristic (dc 50Hz 60Hz) if applicable: 70dB

Total input system transfer time (TAID + TAIT): 2ms

Sample duration time (including setting time): 1ms

Sample repetition time: 0.5ms

Refresh time: 0.5ms



Input characteristics: Third order

Maximum transition frequency: 200Hz

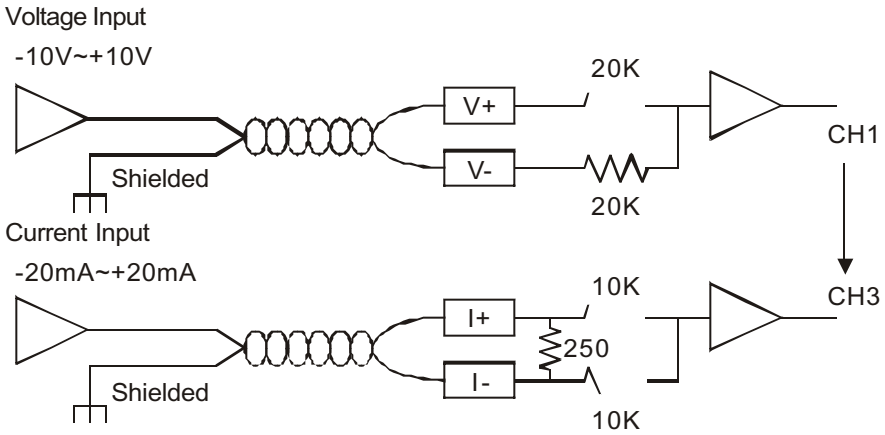
Conversion method: SAR (Successive Approximation Register)

Operating modes: Self-scan

*Please use a twisted pair shielded cable for the analog input/output, this cable should be wired away from powers lines or any other lines which induce noise. (Suggested cable length: under 3m)

*No need for this device to be verified by the factory, and should any problem occurred, please return this device to the original factory or the agent.

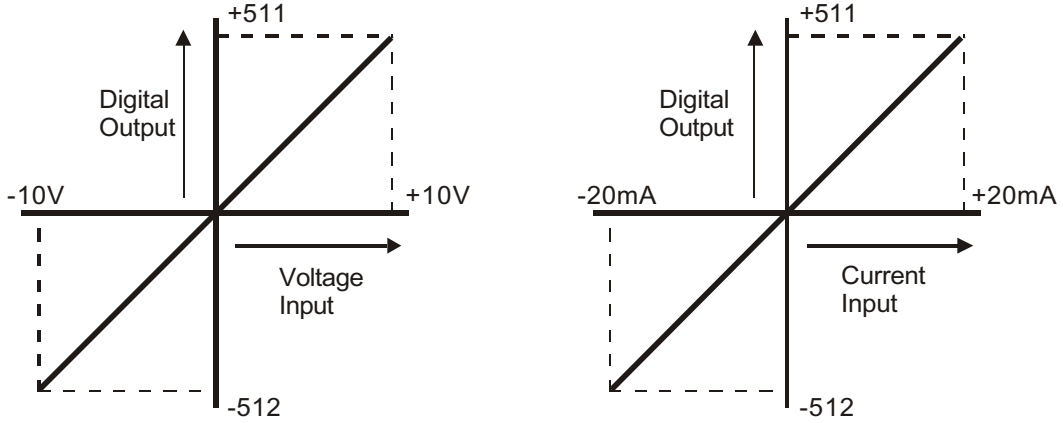
External Wiring Diagram：



1. The analog input is received through a twisted pair shield cable. This cable should be wired separately from power line or any other lines that may induce electrical noise.
2. Connect the ground terminal on the DVP20EX-Series with the grounded terminal on the unit; use class 3 grounding on the unit.

Either voltage or current input can be selected with your choice of input terminal.

Analog input



There are four channels (CH0~CH3) that accept analog inputs. The PLC will convert the analog into a digital format and save it in the corresponding data registers D1110~D1113.

Channel	Analog signals	Reacting time	Input terminals	Resolution (bits)	Accuracy	Analog conversion		Example: Added in +5V to Ch1, and added –5mA to Ch2, the Analog/Digital (A/D) conversions are: D1111 = 256, D1112 = -128
						Reg.	Conversion range	
CH0	Voltage: –10V~+10V Input impedance: 40KΩ Current: -20mA~+20mA Input impedance: 250Ω	5ms	A_0V+~A_0V- A_0I+~A_0I-	10	±1% at full scale of ±10V and ±20mA	D1110	-512~+511	
CH1		5ms	A_1V+~A_1V- A_1I+~A_1I-	10		D1111	-512~+511	
CH2		5ms	A_2V+~A_2V- A_2I+~A_2I-	10		D1112	-512~+511	
CH3		5ms	A_3V+~A_3V- A_3I+~A_3I-	10		D1113	-512~+511	

*This unit may be damaged by input voltages in excess of ±15V or ±30mA.

*If the voltage or current exceeds ±15V or ±30mA during the operation, it will then result in permanent damage to this unit. Users should pay special attention to avoid the above-mentioned incident.

◎ **Digital/Analog (D/A)**

Analog Output:

Monotonicity with no miss code

Overall Precision:

Non-linearity: ±1% of full scale over temperature.

Maximum error: ±1% of full scale of +10V and +20mA over temperature.

Data format returned to the application program: Binary.

Value of LSB (Least Significant Bit):

Voltage output: 78.125 mV

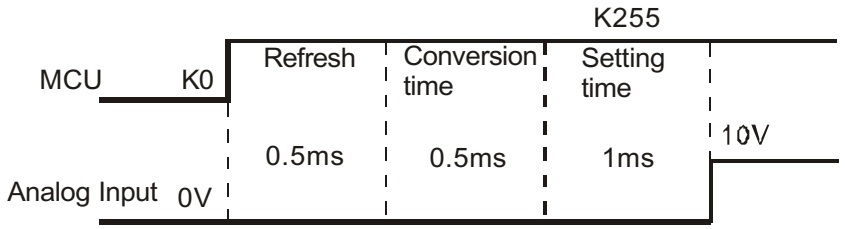
Current output: 78.125 μA

Total input system transfer time (TAID + TAIT) ：2ms

Refresh time:0.5ms

Conversion time:0.5ms

Setting time for full range change = 1ms

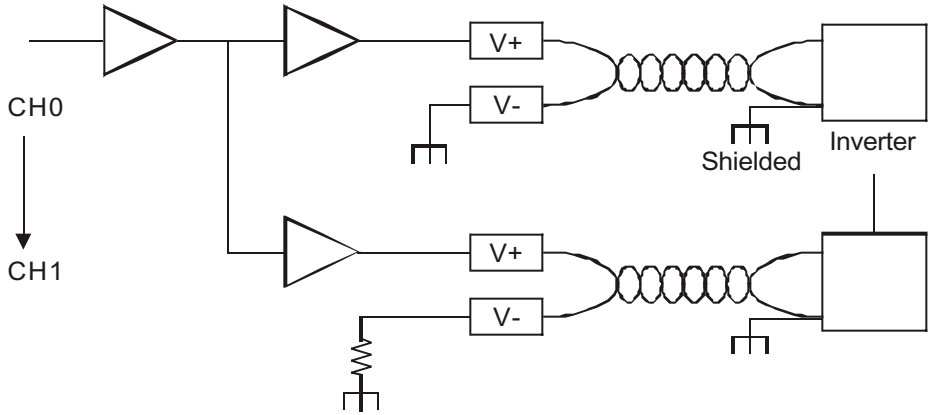


Overshoot : ±1% of full scale

1. The analog output is received through a twisted pair shield cable. This cable should be wired separately from power line or any other lines which may induce electrical noise.

*No need for this device to be verified by the factory, and should any problem occurred, please return this device to the original factory or the agent.

External Wiring Example Diagram：



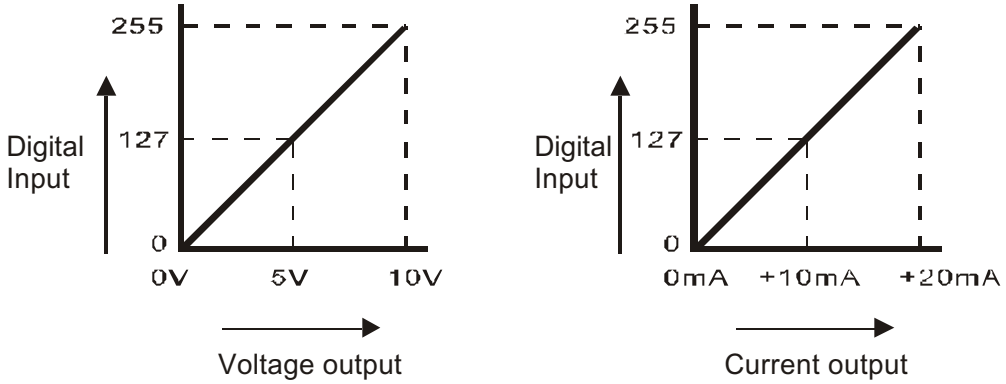
*If the external wiring is not conducted properly, erroneous motions or damages might result, and consequently, if there is the condition of short-circuits for the analog voltage output, it is then very likely that a permanent damage will take place.

Allowed type of loads: floating

Maximum capacitive load (for voltage outputs): 100PF

The responding time from the point where the power is supplied till an output signal is generated: 4 sec

Output ripple: less than 0.1%



There are two channels (CH0~CH1) that convert digital signals saved in D1116~D1117 and output analog signals on specified output terminals.

Channel	Analog signals	Reacting time	Input terminals	Resolution (bits)	Accuracy	Analog conversion		Example:
						Reg.	Conversion range	
CH0	Voltage: 0V~+10V Current: 0mA~+20mA External load resistance: 2K~1MΩ(V), 0~500Ω(I)	5ms	D_0V+~D_0V- D_0I+~D_0I-	8	±1% at full scale of ±10V and 20mA	D1116	0~255	Use MOV command, Let D1116 = 50, D1117 = 90, The signal output will be: CH0 (D0V+,D0V-) ≅ 1.953125V CH1(D1V+, D1V-) ≅ 3.515625 V CH0 (D0I+, D0I-) ≅ 3.90625mA CH1(D1I+, D1I-) ≅ 7.03125mA
		5ms						
CH1		5ms	D_1V+~D_1V- D_1I+~D_1I-	8		D1117	0~255	
		5ms						

* Value 0-255 correspond to current signal 0-20mA so that value 128 corresponds to 10.039mA (20/255 * 128). Value 200 corresponds to 15.686 (20/255 * 200).

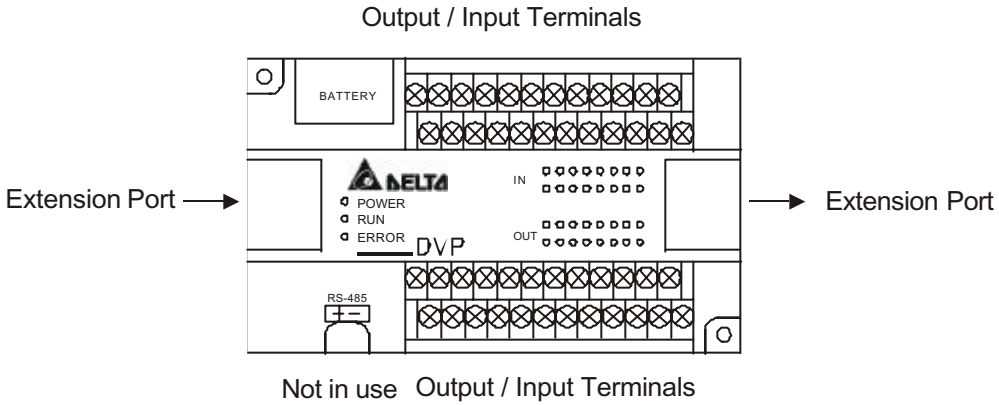
■ I/O Extension Units

The DVP series provides different extension units with specific I/O (please refer to Section 1.1 for specifications). The total input and output points can extend to 256 points. If 256 points are exceeded, the ERROR LED of the MPU will flash.

■ General Specification

Items \ Models	DVP08XN11□ DVP08XP11R	DVP08XM11N DVP16XM01N	DVP-16XN01□	DVP24XP01□	DVP24XN01□ DVP32XP01□	DVP24XP00□ DVP24XN00□ DVP32XP00□
Power Supply Voltage	24VDC (-15%~20%)					100~240VAC (-15%~20%) 50/60Hz ± 5%
Fuse Capacity	2A/250VAC					2A/250VAC
Power consumption (MAX)	5W	5W	6.5W	6.5W	8W	30VA
DC24V supply current	—	—	—	—	—	400mA
Power Protection	—					DC24V output with short-circuit protection
Withstand Voltage	1500VAC(Primary-secondary)，1500VAC(Primary-PE)，500VAC(Secondary-PE)					
Retentive Power Interruption	Continues operation within 5ms					Continues operation within 10ms
Insulation Resistance	>5 MΩ at 500VDC (Between all inputs/outputs and earth)					
Noisy Immunity	ESD: 8KV Air Discharge EFT: Power Line: 2KV, Digital I/O: 1KV, Analog & Communication I/O: 250V Damped-Oscillatory Wave: Power Line: 1KV Digital I/O: 1KV RS: 26MHz~1GHz, 10V/m					
Grounding	The diameter of grounding wire cannot be smaller than the wire diameter of terminals L and N (All DVP units should be grounded directly to the ground pole).					
Operation/Storage Environment	Operation: 0℃~55℃ (Temperature), 50~95% (Humidity); Storage: -25℃~70℃ (Temperature), 5~95% (Humidity) Pollution degree 2					
Vibration /Shock resistance	Standard IEC1131-2, IEC 68-2-6（TEST Fc）/ IEC1131-2 & IEC 68-2-27 (TEST Ea)					
Weight (g)	170/165	160/270	280	434	462/442	600/580

■ I/O Extension Unit Parts and Labels



◎ Status Indicator

1. POWER LED

There is a Power indication LED on the front of the I/O extension unit. When power is on, the POWER LED will light up. If the I/O extension unit LED does not light up and the extension unit is AC power input, please conduct the following test. Remove the **+24V** wire and recheck the LED. If the LED is now ON, then the DC power supply is overloaded and cannot be used. Please use another 24V source.

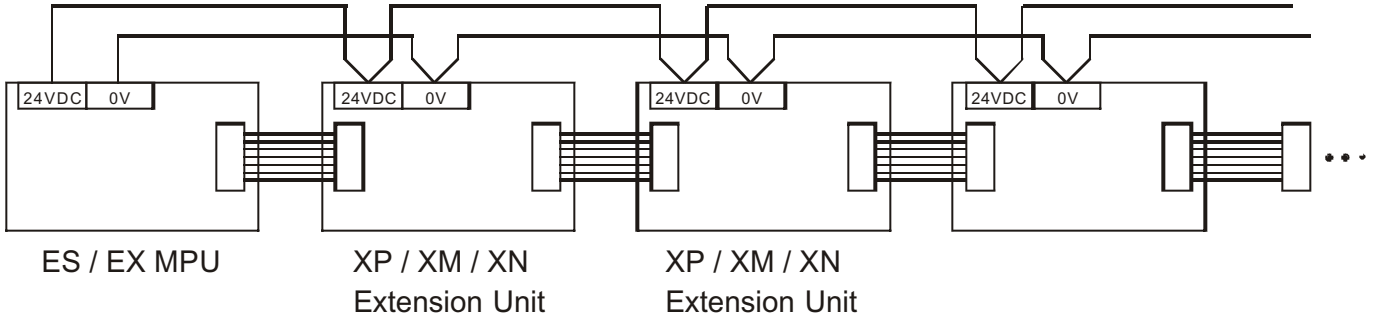
2. LOW V. LED

When the I/O +24V power supply is lower than 17.5V, the extension unit LED of LOW V will light up. At this moment, do not use the **+24V** DC output terminal of the extension unit. Please check your power source.

■ Combined System

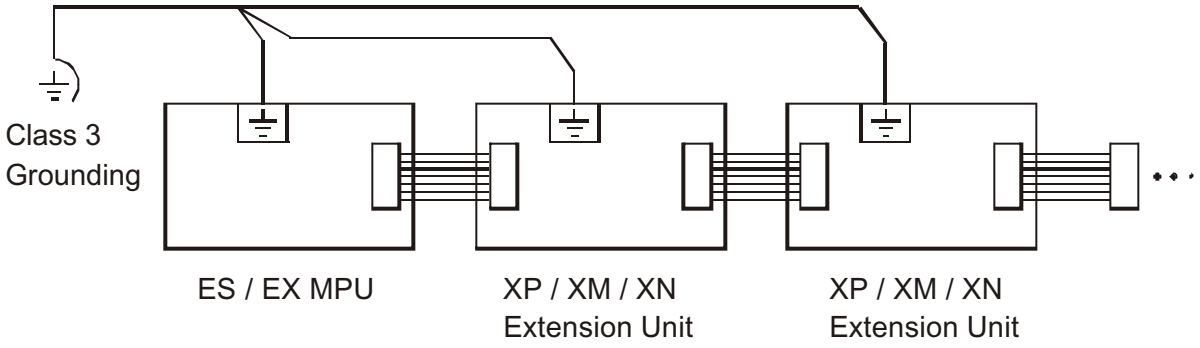
1. Power Terminal

There are two types of power inputs: AC and DC. We can use the +24V output provided by the MPU and supply it to the serial extension units (assuming the DC power supply input is selected). The total current consumed by the extension units cannot exceed the capacity provided by the MPU. Please see the specifications.



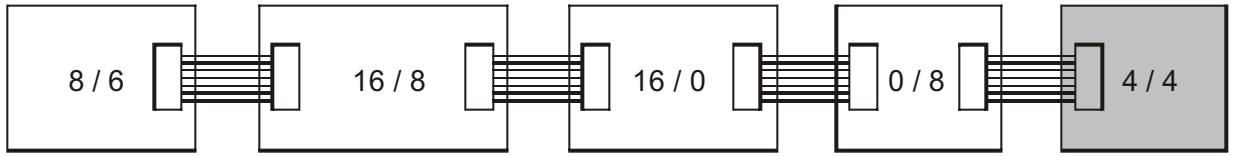
If you have the AC power input Extension Units, the connection method is the same as the AC input on the MPU.

2. Grounding



■ Input / Output points numbering order

No matter how many points the MPU has, the input of the first I/O extension unit will start from X20 and the output will start from Y20.



DVP14ES00R2 DVP24XP01R DVP16XM01N DVP08XN11R DVP08XP11R

Models	Input Points	Output Points	Input Numbering	Output Numbering
DVP14ES00R2	8	6	X00~X07	Y0~Y5
DVP24XP01R	16	8	X20~X37	Y20~Y27
DVP16XM01N	16	0	X40~X57	—
DVP08XN01R	0	8	—	Y30~Y37
DVP08XP11R	4	4	X60~X63	Y40~Y43

- When MPU (DVP60ES00R) connects to a extension unit (DVP24XP01R), the input of the extension unit will start from X50 and the output will start from Y40.

Models	Input Points	Output Points	Input Numbering	Output Numbering
DVP60ES00R2	36	24	X00~X43	Y0~Y27
DVP24XP01R	16	8	X50~X67	Y30~Y37

■ Fault Indication From Panel

Common Problems and Solutions:

◎ POWER LED Indication

There is a Power indication LED on the front of the MPU unit. When power is on, the POWER LED (Green) will light up. If the MPU unit LED does not light up when power is on, please remove the +24V wire and recheck the LED. If the LED is now ON, it means the DC power supply is overloaded and cannot be used. Please use another DC24V source.

If the POWER LED still does not light up when the power is on after the above corrective actions, the PLC should be sent back to the dealer or the distributor whom you purchased the product from.

◎ PLC RUN LED

When PLC is operating, the RUN LED will light up. Users can use an HPP or the Ladder Diagram to enter commands of RUN and STOP.

◎ ERROR LED

The LED will flash if the program sent to the PLC is incorrect or too large. The user can check both the error codes saved in the MPU data register D1004 and the fault codes of this chapter to correct the programs. Then, send the corrective programs back to the MPU unit again.

If the connections between the PLC are failed and the LED will flash rapidly, this indicates the DC24V power supply is down and please check for possible DC24V overload.

The LED will be steady if the program loop execution time is over the preset time (D1000 preset value), check the programs or the WDT (Watch Dog Timer). When the LED lights up, switch the power ON and OFF to see if the RUN LED is off. If not, please check if there is any noise interference or any foreign object in the PLC.

◎ Input point Indication LED

Input point ON/OFF status can be seen from the LED lights, which could also be retrieved from the HPP device monitoring function for the monitoring purpose. Once the input point ON/OFF status is valid, the indicator will be ON. Therefore, if errors are detected, utilize HPP, the indicator and the input signal circuit to check whether everything is normal. Special care should be taken with those electronic switches with great current leakage, for it might result in unexpected motions in the input point.

◎ Output Point LED Indication

Output LED indicates if the output signals are ON or OFF. Please check the following items when the LED ON/OFF indication does not correspond to the commands.

- ☞ Output contacts may be melted and stuck together due to a short circuit or current overload.
- ☞ Check wiring and verify screws are tight.
- ☞ Fault Codes

If the ERROR LED is flashing, the problem may be an invalid commands, communication error, invalid operation, or missing instructions, error indication is given by self-checking function and corresponding error code and error step are stored in special registers. This section gives description of cause and corrective action for each error.

If an error occurred, corresponding error codes can be read from the PC or HPP. The following table shows the error messages, description and cause of error. Error codes and error steps are stored in the following special registers.

Error code : D1004
Error step : D1137

Fault Code	Description
0001	Operand bit device S exceeds the usage range
0002	Label P exceeds the usage range or duplicated
0003	Operand KnSm exceeds the usage range
0102	Interrupt pointer I exceeds the usage range or duplicated
0202	Instruction MC exceeds the usage range
0302	Instruction MCR exceeds the usage range
0401	Operand bit device X exceeds the usage range
0403	Operand KnXm exceeds the usage range
0501	Operand bit device Y exceeds the usage range
0503	Operand KnYm exceeds the usage range
0601	Operand bit device T exceeds the usage range

Fault Code	Description
0604	Operand word device T register usage exceeds limit
0801	Operand bit device M exceeds the usage range
0803	Operand KnMm exceeds the usage range
0D01	DECO Misuse operand
0D02	ENCO Misuse Operand
0D03	DHSCS Misuse Operand
0D04	DHSCR Misuse Operand
0D05	PLSY Misuse Operand
0D06	PWM Misuse Operand
0D07	FROM / TO Misuse Operand
0D08	PID Misuse Operand

Fault Code	Description
0E01	Operand bit device C exceeds the usage range
0E04	Operand word device C register usage exceeds limit
0E05	DCNT misuse operand C
0E18	BCD Conversion Error
0E19	DIVISION (divisor=0)
0F04	Operand word device D register usage exceeds limit
0F05	DCNT misuse operand D
0F06	SFTR misuse operand
0F07	SFTL misuse operand
0F08	REF Misuse Operand
1000	ZRST misuse operand
C400	An unrecognized instruction code is being used
C401	Loop Error
C402	LD / LDI continuously use more than 9 times
C403	MPS continuously use more than 9 times
C404	FOR-NEXT exceed 6 levels
C407	STL continuously use more than 9 times

Fault Code	Description
C405	STL / RET used between FOR and NEXT SRET / IRET used between FOR and NEXT MC / MCR used between FOR and NEXT END / FEND used between FOR and NEXT
C408	Use MC / MCR in STL, Use I / P in STL
C409	Use STL / RET in Subroutine, Interrupt Service Routine STL / RET
C40A	Use MC / MCR in Subroutine, Interrupt Service Routine MC / MCR
C40B	MC / MCR does not begin from N0 or discontinuously
C40C	MC / MCR corresponding value N is different
C40D	Use I / P incorrectly
C40E	IRET does not follow by the last FEND command SRET does not follow by the last FEND command
C41C	The number of input/output points of I/O extension unit is larger than the specified limit
C4EE	No END command in the program

Error Check Devices	Description	Drop Latch	STOP → RUN	RUN → STOP
M1067	Program execution error flag	None	Reset	Latch
M1068	Execution error latch flag	None	Latch	Latch
D1067	Algorithm error code	None	Reset	Latch
D1068	Step value of algorithm errors	None	Latch	Latch

Device D1067 Error Code	Description
0E18	BCD Conversion Error
0E19	DIVISION (divisor=0)
0E1A	Operand bit device exceeds the usage range (including index register E, F)
0E1B	The value of square root is negative

■ Periodic Inspection

Preventive maintenance is required to operate this DVP series PLC in its optimal condition, and to ensure a long life. Be sure to observe the following precautions when selecting a mounting location. Failure to observer these precautions may void the warranty!

☞ Do not mount the DVP near heat-radiating elements or in direct sunlight.

☞ Do not install the DVP in a place subjected to high temperature, high humidity, excessive vibration, corrosive gasses, liquids, airborne dust or metallic particles.

☞ Periodically check if the wiring and terminals are tight.

10.1. New Special M and D Devices

■ Double Frequency Select Function of High Speed Counter

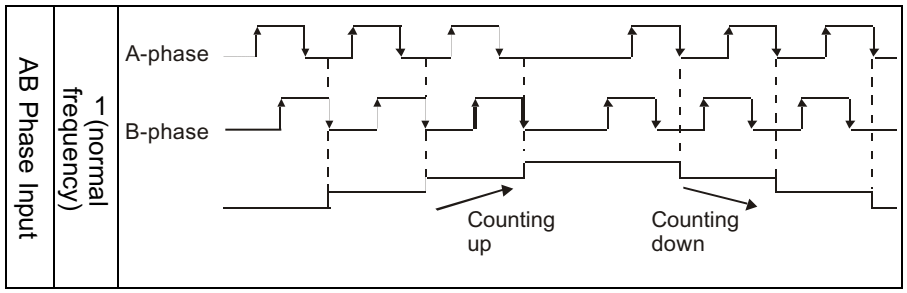
The version 5.5 (D1005=K5301) and above of ES / EX / SS series models support this function.

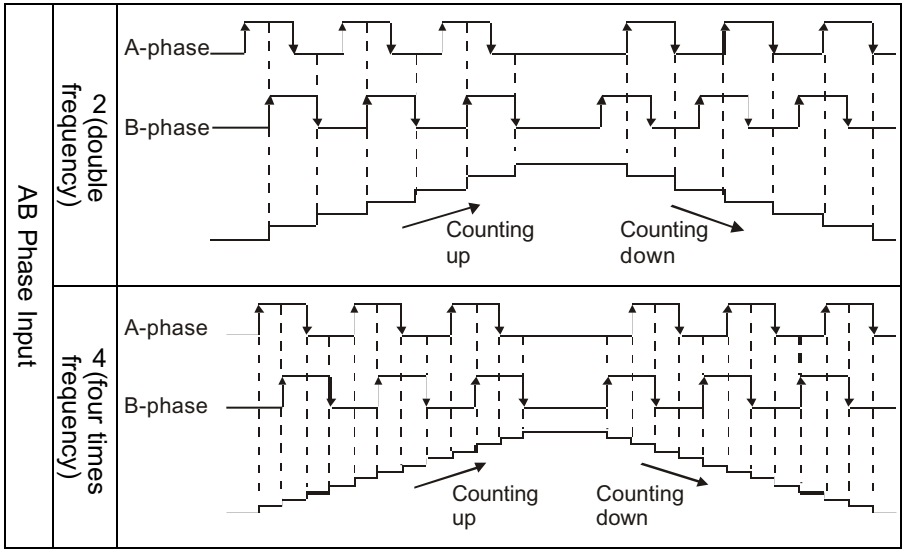
1. New Special Data Registers:

The content value of data register, D1022 will load in the first scan time when PLC switches from STOP to RUN.

Device No.	Function Description
D1022	Use counting method of counter setting double frequency
D1022=K1	Select (normal frequency) mode
D1022=K2	Select (double frequency) mode
D1022=K4	Select (4 times frequency) mode

2. Double Frequency Mode





Only AB phase high speed counter provide the double frequency selection function to set double frequency.

■ Pulse Input with Acceleration / Deceleration Functions Explanation: The version 5.5 (D1005=K5301) and above of ES / EX / SS series models support this function.

1. Meanings of Special M and D Devices:

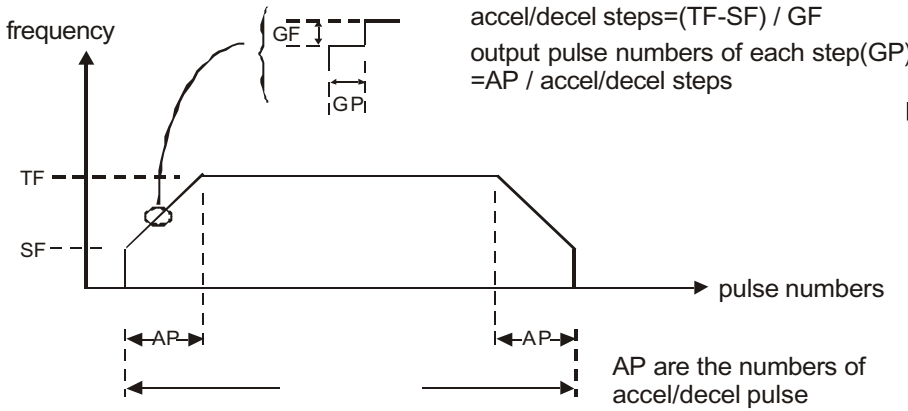
Device No.	Function Description
M1115	Accel/Decel pulse input start switch
M1116	Acceleration flag
M1117	Arrival target frequency flag
M1118	Deceleration flag
M1119	Completed function flag
D1104	Used parameter index value (for D devices)

2. Parameter List (The available range of using D device is D0 to D596 for ES series models and the frequency range is 25Hz~10KHz.)

Index value +	Function Description
0	Starting frequency (SF)
1	Gap frequency (GF)
2	Target frequency (TF)
3	Lower byte of total output pulse numbers amount
4	Higher byte of total output pulse numbers amount
5	Lower byte of total accel/decel interval output pulse numbers
6	Higher byte of total accel/decel interval output pulse numbers

3. Instruction of Functions:

It is not necessary to use commands. After user complete the parameter list, set up M1115 to start. (This step must be executed in RUN mode). The function only can use Y0 output and the timing chart is shown as follow:



accel/decel steps=(TF-SF) / GF

output pulse numbers of each step(GP) =AP / accel/decel steps

4. Note:

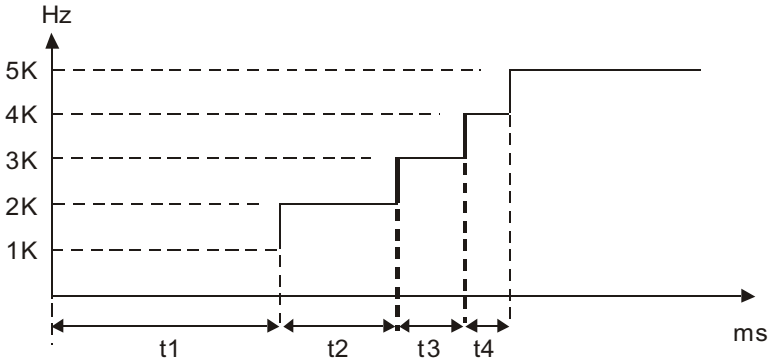
These additional functions will not be executed if anyone of the limit conditions below is not compatible:

1. Starting frequency (SF) < Target frequency (TF)
2. Target frequency (TF) < Target frequency (TF) – Starting frequency (SF)
3. Total pulse numbers amount > numbers of accel/decel pulse x 2
4. During RUN execution, if there is a PLSY command to assign Y0 output, please first start the command of Y0 output and keep on executing and not execute the other.
5. Minimum of starting and target frequency: 25Hz
   Maximum of starting and target frequency: 10KHz
6. Numbers of of accel/decel pulse > accel/decel steps

When M1115 goes from ON to OFF, M1119 will be reset and M1116, M1117 and M1118 will has no change. When PLC goes from STOP to RUN, M1115 to M1119 will be reset to OFF and when PLC goes from RUN to STOP, M1115 to M1119 will also be reset to OFF. D1104 will be reset to 0 only when it goes OFF to ON but in other conditions, D1104 will has no change.

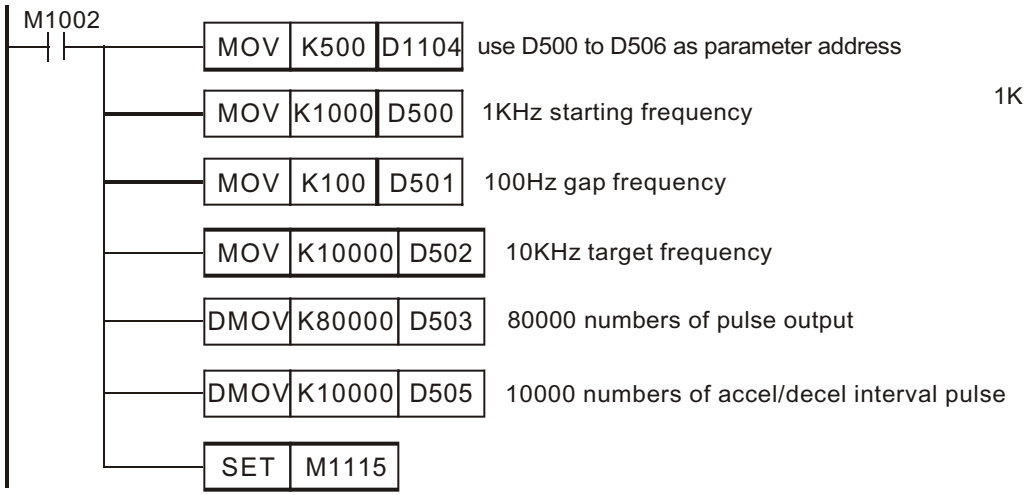
■ How to count the action time of each interval

For example, if the user set the starting frequency is 1KHz, gap frequency is 1KHz, target frequency is 5KHz, total pulse numbers amount is 100 and numbers of accel/decel pulse is 40, then the timing chart of accel interval will be shown as below:



Due to the above conditions, we can know the accel/decel steps is (5K – 1K) / 1K = 4 and pulse output numbers of each step is 40 / 4 = 10. Therefore, from the above timing chart, we can obtain that t1 = (1 / 1K) x 10 = 10ms, t2 = (1 / 2K) x 10 = 5ms, t3 = (1 / 3K) x 10 = 3.33ms and t4 = (1 / 4K) x 10 = 2.5ms.

■ Program example: REV/FWD operating acceleration /deceleration of step motor control



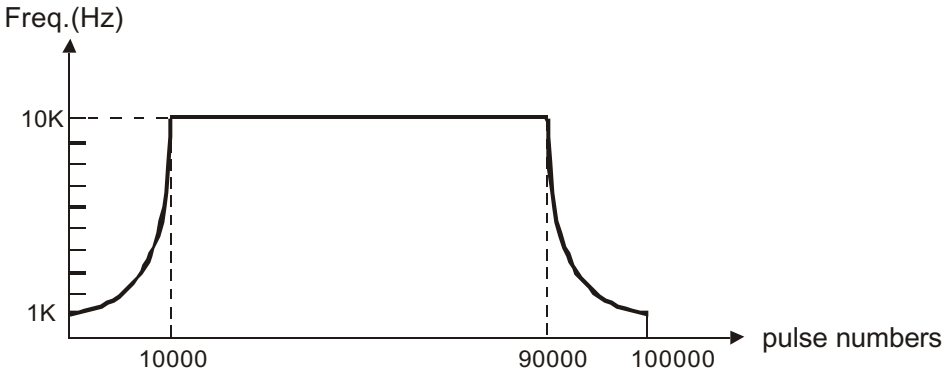
◎ When PLC is running, each parameter setting is stored in the register assigned by D1104.

◎ When M1115 is in the status of acceleration/deceleration, pulse output will start.

◎ M1116 is ON during the process of acceleration, M1117 is ON when desired speed attained and M1118 is ON during the process of deceleration. After the program is completed, M1119 is ON.

◎ M1115 will not return automatically. Users have to conclude the condition during that operating period and reset it by self-decision.

◎ Actual pulse output curve is shown as follow:



10.2. New Application Instructions

The version V5.5 or above of ES / EX / SS series has the following new commands.

D			
	124	**EXP** (S) (D)	Convert decimal of binary number system to perform exponent operation

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

■ (S) : the source device for operation

(D) : operation result device

■ Take e =2.71828 as the base and use (S) as exponent to perform the EXP operation.

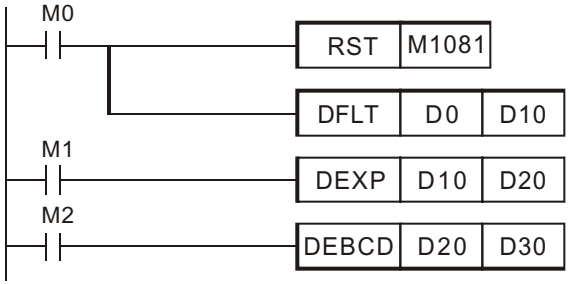
exp[(S)+1, (S)]=[(D)+1, (D)]

■ The positive and negative values of the content of (S) are all valid. Be sure to use 32-bit data format to assign the (D) register. Because using the floating point format to perform the EXP operation is necessary, (S) has to be converted to the floating point value.

■ The content value of D operand= e S ; e=2.71828, S is the assigned source data.

■ (Zero flag, Carry flag, Overflow flag are valid and Error flag M1067, M1068 read D1067, D1068)

■ Program example:



◎ When M0 is ON, the data of (D0, D1) is converted to decimal of binary number system and the result is stored in the (D10, D11) register.

◎ When M1 is ON, use (D10, D11) as exponent to perform the EXP operation. The result is decimal of binary number system and will be stored in the (D20, D21) register.

◎ When M2 is ON, the data of (D20, D21) is converted from binary to decimal and the result is stored in (D30, D31) register. (At that time, the result=D30 x 10^{D31})

D			
	125	**LN** (S) (D)	Convert decimal of binary number system to perform natural logarithm operation

Device / Operand	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S)													◎		
(D)													◎		

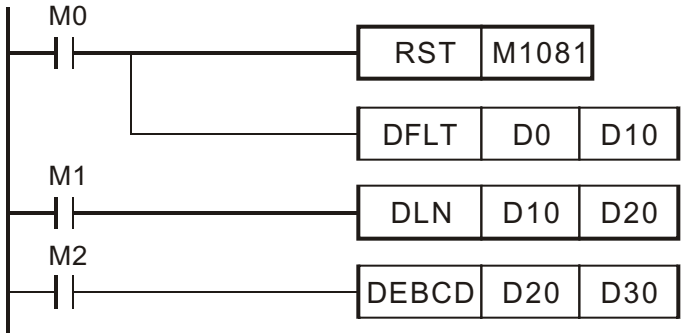
■ (S) : the source device for operation

(D) : operation result device

■ Use (S) as operand to perform the LN operation.

ln[(S)+1, (S)]=[(D)+1, (D)]

- Only the positive value of the content of (S) is valid. Be sure to use 32-bit data format to assign the (D) register. Because using the floating point format to perform the LN operation is necessary, (S) has to be converted to the floating point value.
- e^{D}=S => content value of D operand= lnS ; S is the assigned source data.
- (Zero flag, Carry flag, Overflow flag are valid and Error flag M1067, M1068 read D1067, D1068)
- Program example:



◎ When M0 is ON, the data of (D0, D1) is converted to decimal of binary number system and the result is stored in the (D10, D11) register.

◎ When M1 is ON, use (D10, D11) as real number to perform the LN operation. The result is decimal of binary number system and will be stored in the (D20, D21) register.

◎ When M2 is ON, the data of (D20, D21) is converted from binary to decimal and the result is stored in (D30, D31) register. (At that time, the result=D30 x 10^{D31})

D **126** **LOG** (S1) (S2) (D) — Convert decimal of binary number system to perform logarithm operation

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)													◎		
(S2)													◎		
(D)													◎		

- (S1) : the base device for operation

  (S2) : the source device for operation

  (D) : operation result device

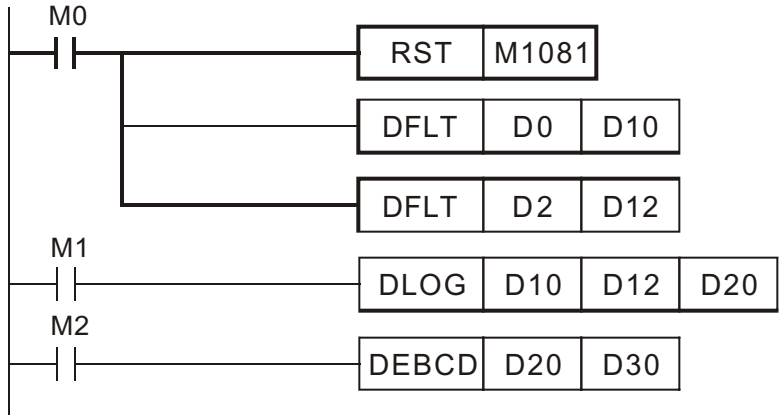
- Use the content of (S1) and (S2) as operand to perform the LOG operation and store the operation result in (D) device.
- Only the positive value of the content of (S2) is valid (The positive and negative values of content of (S1) are all valid). Be sure to use 32-bit data format to assign the (D) register. Because using the floating point format to perform the LOG operation is necessary, (S1) and (S2) have to be converted to the floating point value.

  $S1^{D}$=S2, obtain the value of D=>Log $_{S1}{}^{S2}$=D

  When knowing S1=5,S2=125, obtain D=log $_{5}{}^{125}$=?

  $S1^{D}$=S2=>5^{D}=125=>D=log $_{5}{}^{125}$=3
- (Zero flag, Carry flag, Overflow flag are valid and Error flag M1067, M1068 read D1067, D1068)

■ Program example:



◎ When M0 is ON, the data of (D0, D1) and (D2, D3) are converted to decimal of binary number system and the result are stored in the 32-bit register, (D10, D11) and (D12, D13).

◎ When M1 is ON, use the 32-bit register, (D10, D11) and (D12, D13) to perform the LOG operation. The result will be the decimal of binary number system and stored in the 32-bit register, (D20, D21).

◎ When M2 is ON, the data of (D20, D21) is converted from binary to decimal and the result is stored in (D30, D31) register. (At that time, the result=D30 x 10^{D31})

D **128** **POW** (S1) (S2) (D)

Convert decimal of binary number system to perform power operation

Operand \ Device	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
(S1)													◎		
(S2)													◎		
(D)													◎		

■ (S1) : the base device (S2) : the exponential device

(D) : operation result device

■ The floating point data of (S1) and (S2) are multiplied in an exponential manner and the result is stored in (D) device.

pow [(S1)+1, (S1)]^[(S2)+1, (S2)]=(D)

■ Only the positive value of the content of (S1) and (S2) are valid. Be sure to use 32-bit data format to assign the (D) register. Because using the floating point format to perform the LOG operation is necessary, (S1) and (S2) have to be converted to the floating point value.

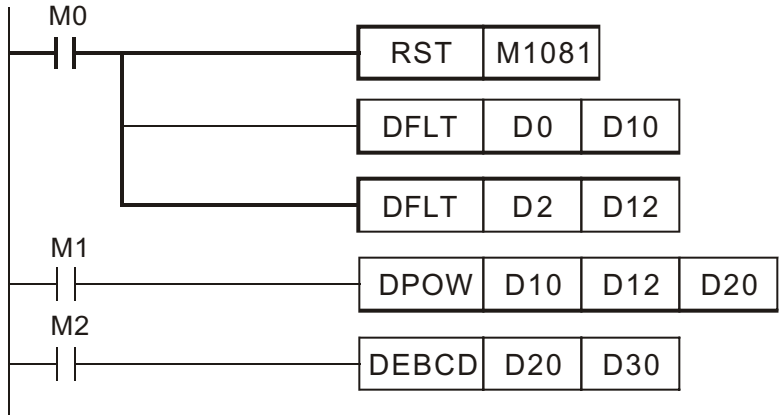
$S1^{S2}$=D, obtain the value of D

When knowing S1=5,S2=3, obtain $D=5^3$=?

$D=5^3$=125

■ (Zero flag, Carry flag, Overflow flag are valid and Error flag M1067, M1068 read D1067, D1068)

■ Program example:



◎ When M0 is ON, the data of (D0, D1) and (D2, D3) are converted to decimal of binary number system and the result are stored in the 32-bit register, (D10, D11) and (D12, D13).

◎ When M1 is ON, use the 32-bit register, (D10, D11) and (D12, D13) to perform the POW operation. The result will be the decimal of binary number system and stored in the 32-bit register, (D20, D21).

◎ When M2 is ON, the data of (D20, D21) is converted from binary to decimal and the result is stored in (D30, D31) register. (At that time, the result=D30 x 10^{D31})

Appendix A: Communication Function Explanation

1. Introduction

This chapter explains the details and the methods of DVP series PLC communication function. When DVP series PLC communication port is used for the communication protocol of slave, it can read and write the interior device of PLC to be the operation reference used for connecting the master and HMI (or other upper bit equipments) of PLC.

2. Communication Interface

PLC Models		Communication Interface
DVP-ES/EX/SS	Version 4.7 (included) or less	COM1: RS-232C COM2: RS-485 Either of them, they cannot be the slave station at the same time.
	Version 5.1 (included) or more	COM1: RS-232C COM2: RS-485 They can be the slave station at the same time.

3. Communication Protocol

Communication Protocol and Exterior Communication of MOSBUS ASCII mode

Communication format	Specifications
Baud rate	9600 bps
Start bit	1
Data length	7
Parity	Even parity
Stop bit	1

4. Communication Connection Method

Communication port	Communication connection method
RS-232	Use connection cables DVPACAB215, DVPACAB230 or DVPACAB2A30
RS-485	Master must be with RS-485 interface and use shielded twisted-pair cables.

5. Communication Message Format

STX	Start bit ‘ : ’ (3AH)
ADR 1	Communication address:
ADR 0	Tw0 ASCII word bit, total 16 bit
CMD 1	Command code:
CMD 0	Tw0 ASCII word bit, total 16 bit
DATA（0）	Data content:
DATA（1）	“n” numbers of ASCII word bit, total n*16 bit
…….	Limit n≦74 ASCII code
DATA（n-1）	
LRC CHK 1	Detection error value:
LRC CHK 0	Tw0 ASCII word bit, total 16 bit
END 1	End word device:
END 0	END 1 = CR（0DH），END 0 = LF（0AH）

6. Command Code

Code	Description	Applicable device
01	Read coil status	S, Y, M, T, C
02	Read input status	S, X, Y, M,T, C
03	Read data of hold register	T, C, D
05	Force single coil output	S, Y, M, T, C
06	Change single register data	T, C, D
15	Force multi coil output	S, Y, M, T, C
16	Change multi register data	T, C, D
17	Response Slave ID	None

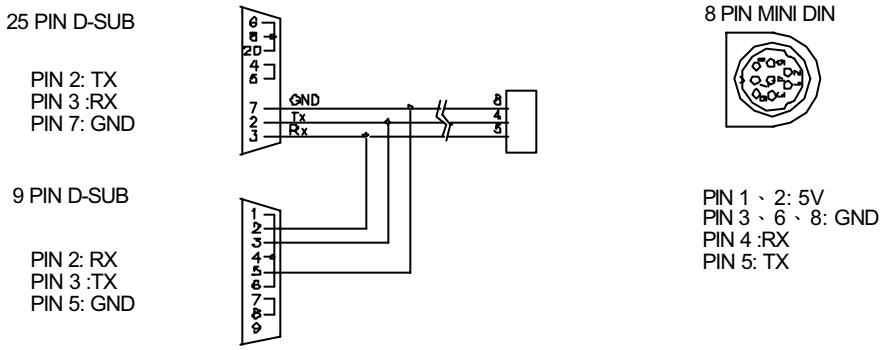
7. Device Numbers

Models	ES/EX/SS				
Device	Range		Device Type	Device address (Hex)	Quality
S	000~127		Bit	0000~007F	128
X	000~177 (Octal number system)		Bit	0400~047F	256
Y	000~177 (Octal number system)		Bit	0500~057F	256
T	000~127		Bit/Bit group	0600~067F	128
M	000~1279		Bit	0800~0CFF	1280
C	0~127	16-bit	Bit	0E00~0EC7	128
	235~254	32-bit	Bit/Double bit group	0EC8~0EFF	13
D	0~1311		Bit group	1000~151F	1322

8. Communication Wiring Diagram

Use these DVPACAB215 (1.5M), DVPACAB230 (3.0M) or DVPACAB2A30 (without 25 Pin D-SUB, 3.0M) cables when connecting DVP series PLC with the upper bit equipment (such as computer or HMI). Users must connect wiring according to the circuit diagram shown below.

Connecting with PC or TP04G



⚠ Note: Never wire to the unused terminals to prevent short circuit and burn out.

DELTA ELECTRONICS, INC.

EC Declaration of Conformity
According to the Low Voltage Directive 73/23/EEC and the Amendment Directive 93/68/EEC

For the following equipment:

Programmable Logic Controller

(Product Name)

Control Unit : DVP14ES00R, DVP14ES00T, DVP14ES01R, DVP14ES01T, DVP24ES00R, DVP24ES00T, DVP24ES01R, DVP24ES01T, DVP32ES00R, DVP32ES00T, DVP32ES01R, DVP32ES01T, DVP20EX00R, DVP20EX00T

Expansion Unit: DVP08XN11R, DVP08XN11T, DVP08XP11R, DVP08XP11T, DVP08XM11N, DVP16XN11R, DVP16XN11T, DVP16XM11N , DVP16XM01N, DVP16XN01R, DVP16XN01T, DVP24XN00R, DVP24XN00T, DVP24XP01R, DVP24XP01T, DVP24XN01R, DVP24XN01T, DVP24XP11R, DVP24XP11T, DVP24XN11R, DVP24XN11T, DVP32XP00R, DVP32XP00T, DVP32XP01R, DVP32XP01T, DVP32XP11R, DVP32XP11T

Handheld Programmable Panel : DVPHPP01

(Model Name)

is herewith confirmed to comply with the requirements set out in the Council Directive 73/23/EEC for electrical equipment used within certain voltage limits and the Amendment Directive 93/68/EEC. For the evaluation of the compliance with this Directive, the following standard was applied:

EN61131-2

The following manufacturer/importer is responsible for this declaration:

Delta Electronics, Inc.

(Company Name)

DELTA ELECTRONICS,INC.

EC Declaration of Conformity
According to the Electromagnetic Compatibility 89/336/EEC and the Amendment Directive 93/68/EEC

For the following equipment:

Programmable Logic Controller

(Product Name)

Control Unit : DVP14ES00R, DVP14ES00T, DVP14ES01R, DVP14ES01T, DVP24ES00R, DVP24ES00T, DVP24ES01R, DVP24ES01T, DVP32ES00R, DVP32ES00T, DVP32ES01R, DVP32ES01T, DVP20EX00R, DVP20EX00T,

Expansion Unit : DVP08XN11R, DVP08XN11T, DVP08XP11R, DVP08XP11T, DVP08XM11N, DVP16XN11R, DVP16XN11T, DVP16XN01R, DVP16XN01T, DVP24XN00R, DVP24XN00T, DVP24XP01R, DVP24XP01T, DVP24XN01R, DVP24XN01T, DVP24XP11R, DVP24XP11T, DVP24XN11R, DVP24XN11T, DVP32XP00R, DVP32XP00T, DVP32XP01R, DVP32XP01T, DVP32XP11R, DVP32XP11T,

Handheld Programmable Panel : DVPHPP01

(Model Name)

is herewith confirmed to comply with the requirements set out in the Council Directive 89/336/EEC for electrical equipment used within certain voltage limits and the Amendment Directive 93/68/EEC. For the evaluation of the compliance with this Directive, the following standard was applied:

EN61131-2

The following manufacturer/importer is responsible for this declaration:

Delta Electronics, Inc.

(Company Name)