

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

Panasonic

PROGRAMMABLE CONTROLLERS

FP0R

User's Manual

ACGM0475V3EN

Before beginning

Liability and copyright for the hardware

This manual and everything described in it are copyrighted. You may not copy this manual, in whole or part, without written consent of Panasonic Electric Works Europe AG (PEWEU).

PEWEU pursues a policy of continuous improvement of the design and performance of its products. Therefore we reserve the right to change the manual/product without notice. In no event will PEWEU be liable for direct, special, incidental, or consequential damage resulting from any defect in the product or its documentation, even if advised of the possibility of such damages.

We invite your comments on this manual. Please e-mail us at:

techdoc.peweu@eu.panasonic.com.

Please direct support matters and technical questions to your local Panasonic representative.

Limited warranty

If physical defects caused by distribution are found, PEWEU will replace/repair the product free of charge. Exceptions include:

- When physical defects are due to different usage/treatment of the product other than described in the manual.
- When physical defects are due to defective equipment other than the distributed product.
- When physical defects are due to modifications/repairs by someone other than PEWEU.
- When physical defects are due to natural disasters.

Warnings used in this manual

One or more of the following warnings may be used in this documentation:

DANGER	
	Indicates a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING	
	Indicates a hazardous situation which, if not avoided, could result in serious or moderate injury.

CAUTION	
	Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE
Indicates a property damage message.

Scope of this manual

The FP0R User's Manual includes:

- specifications for the CPU types and expansion units of the FP0R
- installation, wiring, and maintenance instructions
- general programming information
- troubleshooting information
- an appendix with:
  - technical specifications
  - I/O allocation tables
  - memory area tables
  - system registers
  - unit dimensions

Please refer to the FP Series Programming Manual or to the online help of Control FPWIN Pro for information on:

- system instructions
- special internal flags
- data registers
- system variables
- memory area tables
- programming examples

For documentation on other units used with the FP0R, please refer to the hardware manual for that unit.

All manuals can be downloaded from the **Panasonic** Web site (http://www.panasonic-electric-works.com).

Safety measures

Operating environment

After installing the unit, make sure to use it within the range of the general specifications:

- Ambient temperature: 0°C–+55°C
- Ambient humidity: 10%–95% RH (at 25°C, non-condensing)
- Pollution degree: 2
- Do not use the unit in the following environments:
  - Direct sunlight
  - Sudden temperature changes causing condensation
  - Inflammable or corrosive gases
  - Excessive airborne dust, metal particles or salts
  - Benzine, paint thinner, alcohol or other organic solvents or strong alkaline solutions such as ammonia or caustic soda
  - Vibration, shock or direct drop of water
  - Influence from power transmission lines, high voltage equipment, power cables, power equipment, radio transmitters, or any other equipment that would generate high switching surges. Maintain at least 100mm of space between these devices and the unit.

Static electricity

Before touching the unit or equipment, always touch some grounded metal to discharge any static electricity you may have generated (especially in dry locations). The discharge of static electricity can damage parts and equipment.

Protection of power supply

- Use a twisted power supply wire.
- Insulate the wiring systems to the CPU, input/output devices, and mechanical power apparatus.
- An insulated power supply with an internal protective circuit should be used (FP power supply). The power supply for the CPU is a non-insulated circuit, so if an incorrect voltage is directly applied, the internal circuit may be damaged or destroyed.

- If using a power supply device without an internal protective circuit, always make sure power is supplied to the unit through a protective element such as a fuse.
- Be sure to supply power to a CPU and an expansion unit from the same power supply, and turn the power on and off simultaneously for both.

Power supply sequence

Make sure the power supply of the CPU turns off before the power supply for input and output. If the power supply for input and output is turned off first, the CPU will detect the input fluctuations and may begin an unexpected operation.

Before turning on the power

When turning on the power for the first time, be sure to take the precautions given below.

- During installation, check that there are no scraps of wiring, particularly conductive fragments, adhering to the unit.
- Verify that the power supply wiring, I/O wiring, and power supply voltage are all correct.
- Sufficiently tighten the installation and terminal screws.
- Set the operation mode selector to PROG mode.

Before entering a program

Be sure to clear any existing program before entering a new program.

Procedure

1. **Online** → **Online mode** or 
2. **Online** → **Clear PLC**
3. [OK]

Request concerning program storage

To prevent the accidental loss of programs, the user should consider the following measures:

- Backing up programs: To avoid accidentally losing programs, destroying files, or overwriting the contents of a file, use the backup or export

functions of Control FPWIN Pro and store the files in a safe place. Additionally, you can print out the entire project documentation.

- Specifying passwords: The password setting is designed to avoid programs being accidentally overwritten. If the password is forgotten, however, it will be impossible to overwrite the program even if you want to. Also, if a password is forcibly bypassed, the program is deleted. Therefore, please note the password in a safe location.

Programming conventions

The programming examples in this manual are designed for Control FPWIN Pro. For FPWIN GR examples, please refer to: FP0R User's Manual ARCT1F475E

Most of the sample programs were written in Ladder Diagram. In Control FPWIN Pro, you can also program in Structured Text, Function Block Diagram, Instruction List, and Sequential Function Chart. For examples in other programming languages, please refer to the Control FPWIN Pro Online Help and the Programming Manual.

The abbreviations used in the examples signify the following:

- POU: Program Organization Unit
- DUT: Data Unit Type
- GVL: Global Variable List

These and other terms are explained in the Control FPWIN Pro Online Help and Programming Manual.

To illustrate the use of positioning instructions, the chapter on high-speed counters and pulse output contains numerous examples. Some of the sample programs can be opened directly in Control FPWIN Pro. The Control FPWIN Pro projects in LD and ST code can be downloaded from the Panasonic Web site (http://www.panasonic-electric-works.com/eu/downloadcenter.htm).

Table of contents

1. Overview 15
1.1 Features 15
1.2 Unit types 18
1.2.1 CPU 18
1.2.2 FP0/FP0R I/O expansion units 19
1.2.3 FP0 intelligent units 20
1.2.4 FP series link units 20
1.2.5 Power supply unit 21
1.2.6 Accessories 21
1.3 Restrictions on unit combinations 22
1.4 Programming tools 23
1.5 FP0 program compatibility 24

2. CPU types 28
2.1 CPU parts and functions 28
2.2 CPU input specifications 31
2.3 CPU output specifications 32
2.4 Terminal layout 35
2.5 Backup and clock/calendar functions 38
2.5.1 Backup function 40
2.5.2 Clock/calendar function 41
2.5.2.1 Memory area for clock/calendar function 41
2.5.2.2 Settings for clock/calendar function 42
2.5.2.3 Sample program for fixed schedule and automatic start 43
2.5.2.4 Sample program for 30-second compensation 44

3. Expansion 45
3.1 Expansion method 45
3.2 Expansion unit parts and functions 46
3.3 Expansion input specifications 47
3.4 Expansion output specifications 48
3.5 Terminal layout 51

4. I/O allocation 55
4.1 General 55
4.2 CPU 56
4.3 FP0/FP0R expansion units 56

5. Installation and wiring 58
5.1 Installation 58
5.1.1 Installation environment and space 58
5.1.2 Using DIN rails 60
5.1.3 Using optional mounting plates 61
5.1.3.1 Slim type mounting plate 61
5.1.3.2 Flat type mounting plate 63
5.2 Connecting FP0/FP0R expansion units 65
5.3 Safety instructions for wiring 66
5.4 Wiring the power supply 68
5.4.1 Grounding 70
5.5 Input and output wiring 71
5.5.1 Input wiring 72
5.5.1.1 Photoelectric and proximity sensors 72
5.5.1.2 Input wiring precautions 75
5.5.2 Output wiring 77
5.5.2.1 Protective circuit for inductive loads 77
5.5.2.2 Protective circuit for capacitive loads 78
5.6 Wiring the MIL connector 79
5.7 Wiring the terminal block 81
5.8 Wiring the COM port 83
5.8.1 Transmission cables 86

6. Communication 87
6.1 Communication modes 87
6.1.1 Terminology in FPWIN Pro and FPWIN GR 87
6.1.2 MEWTOCOL-COM master/slave 88
6.1.3 Program controlled communication 89
6.1.4 PLC Link 90
6.1.5 Modbus RTU master/slave 91
6.2 Ports: names and principle applications 92
6.2.1 TOOL port 92
6.2.2 COM port 93
6.2.3 USB port 93
6.2.3.1 Installing the USB driver 95
6.2.3.2 Communication with the programming tool 97
6.2.3.3 Reinstalling the USB driver 98
6.3 Communication specifications 98
6.4 Communication parameters 102

6.4.1 Setting system registers in PROG mode 102
6.4.2 Changing communication mode in RUN mode 105
6.5 MEWTOCOL-COM 105
6.5.1 Operation outline for MEWTOCOL-COM slave 108
6.5.2 Command and response format 109
6.5.3 Commands 111
6.5.4 Setting communication parameters 112
6.5.4.1 FP0 compatibility mode 113
6.5.5 1:1 slave communication 114
6.5.5.1 1:1 communication with a computer 115
6.5.5.2 1:1 communication with GT touch panels 116
6.5.6 1:N slave communication 117
6.5.7 Sample program for master communication 118
6.6 Program controlled communication 119
6.6.1 Setting communication parameters 121
6.6.1.1 FP0 compatibility mode 124
6.6.2 Sending data 124
6.6.3 Receiving data 126
6.6.3.1 Setting receive buffer for CPU 128
6.6.4 Format of send and receive data 131
6.6.5 Flag operation 133
6.6.5.1 Start code: no-STX, end code: CR 135
6.6.5.2 Start code: STX, end code: ETX 136
6.6.6 1:1 communication 139
6.6.7 1:N communication 139
6.6.8 Programming in FP0 compatibility mode 140
6.7 PLC Link 141
6.7.1 Setting communication parameters 142
6.7.2 Link area allocation 144
6.7.2.1 Example for PLC link 0 145
6.7.2.2 Example for PLC link 1 147
6.7.2.3 Partial use of link areas 148
6.7.2.4 Precautions for allocating link areas 150
6.7.3 Setting the highest station number for a PLC link 151
6.7.4 PLC link 0 and 1 allocation setting 152
6.7.5 Monitoring 152
6.7.6 PLC link response time 155
6.7.6.1 Reducing transmission cycle times 158
6.7.6.2 Error detection time for transmission assurance flags 159

6.8 Modbus RTU communication 160
6.8.1 Setting communication parameters 164
6.8.2 Sample program for master communication 165

7. High-speed counter and pulse output 166
7.1 Overview 166
7.2 Function specifications and restrictions 169
7.2.1 High-speed counter function 169
7.2.2 Pulse output function 171
7.2.3 PWM output function 173
7.2.4 Maximum counting speed and output frequency 173
7.3 High-speed counter function 176
7.3.1 Count input modes 176
7.3.2 Minimum input pulse width 178
7.3.3 I/O allocation 178
7.3.4 Instructions and system variables 179
7.3.4.1 Writing the high-speed counter control code 180
7.3.4.2 Writing and reading the elapsed value of the high-speed counter 184
7.3.4.3 Target value match ON 185
7.3.4.4 Target value match OFF 186
7.3.4.5 Input pulse measurement 187
7.3.5 Sample programs 187
7.3.5.1 Positioning operations with a single-speed inverter 188
7.3.5.2 Positioning operations with a double-speed inverter 189
7.4 Pulse output function 191
7.4.1 Pulse output methods and position control modes 192
7.4.2 I/O allocation 195
7.4.3 Instructions and system variables 196
7.4.3.1 Writing the pulse output control code 199
7.4.3.2 Writing and reading the elapsed value of the pulse output 203
7.4.3.3 Target value match ON (pulse output) 204
7.4.3.4 Target value match OFF 205
7.4.3.5 Trapezoidal control 206
7.4.3.6 JOG operation and positioning 208
7.4.3.7 JOG operation 211
7.4.3.8 Data table control 213
7.4.3.9 Linear interpolation 214
7.4.3.10 Home return 215
7.5 PWM output function 216

8. Security functions 218
8.1 Security function types 218
8.2 Security settings in Control FPWIN Pro 218
8.2.1 Upload protection 218
8.2.2 PLC protection (password protection) 219
8.3 FP Memory Loader 220
8.3.1 Upload protection 220
8.3.2 Download protection 221

9. Other functions 223
9.1 F-ROM backup (P13_EPWT) 223
9.2 Sampling trace 223
9.3 Input time constants 224

10. Troubleshooting 225
10.1 LED display for status condition 225
10.2 Operation on error 226
10.3 ERROR/ALARM LED is flashing 226
10.4 ERROR/ALARM LED is ON 227
10.5 All LEDs are OFF 228
10.6 Diagnosing output malfunction 228
10.7 Password protection error message 229
10.8 PROG mode does not change to RUN 230

11. Appendix 231
11.1 Specifications 231
11.1.1 General specifications 231
11.1.2 Performance specifications 232
11.1.3 Communication specifications 234
11.1.4 Power supply specifications 237
11.1.5 Current consumption 238
11.2 Dimensions 239
11.2.1 C10/C14 CPU (terminal block) 239
11.2.2 C16 CPU (MIL connector) 240
11.2.3 C32 CPU (MIL connector) 241
11.2.4 Power supply unit 242
11.2.5 Using DIN rails 242
11.3 I/O allocation 242
11.4 Flags and memory areas for FP0R 244
11.5 System registers 247

11.5.1 Precautions when setting system registers 247
11.5.2 Types of system registers 247
11.5.3 Checking and changing system registers 248
11.5.4 Table of system registers 249
11.6 Error codes 256
11.6.1 Error codes E1 to E8 256
11.6.2 Self-Diagnostic error codes 257
11.6.3 MEWTOCOL-COM error codes 258
11.7 MEWTOCOL-COM communication commands 259
11.8 Data types 260
11.8.1 Elementary data types 260
11.8.2 Generic data types 261
11.9 Hexadecimal/binary/BCD 262
11.10 ASCII codes 263

Chapter 1

Overview

1.1 Features

The FP0R is an ultra compact PLC (programmable logic controller) with high-speed processing capabilities and a large memory capacity. The controller uses the comprehensive F instruction set and is programmed with Control FPWIN Pro or FPWIN GR. With Control FPWIN Pro, programming according to IEC 61131-3 is possible.

USB 2.0 TOOL port

The TOOL port supports USB 2.0 Full Speed and enables ultra high-speed communication with programming tools. Large programs with up to 32k steps can now be downloaded in as fast as 5s.

For details, see p. 93.

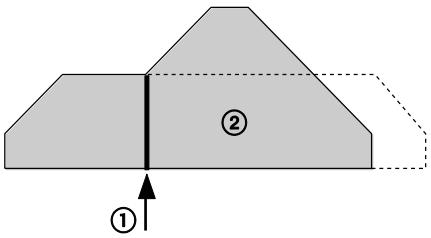
Separate large-capacity comment memory

The unit's comment memory area is separate from the program area, and can store I/O comments for 100 000 points. Program management and maintenance is easy. Thanks to the separate comment area, programs can now be developed without concern for comment memory capacity.

Positioning control using high-speed counter and pulse output

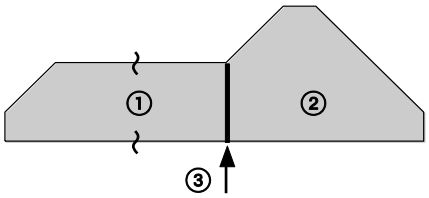
A high-speed counter and a pulse output function are provided as standard features.

- Target speed change



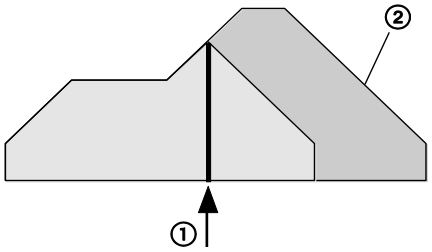
①	Target speed change
②	Number of pulses

- JOG operation



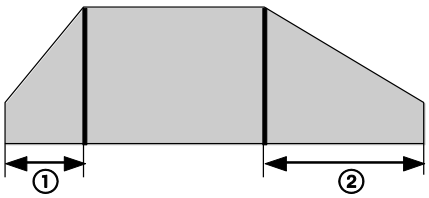
①	JOG operation
②	Number of pulses
③	Position control trigger input

- Decelerated stop



①	Trigger for decelerated stop
②	Number of pulses

- Individual setting for acceleration/deceleration time



①	Acceleration time
②	Deceleration time

For details, see p. 169.

Additional unit with battery free backup function (F32 type)

The F32 type offers a battery-free automatic backup function for all operation memories (internal flags, data registers, timers/counters). Maintainability has been significantly improved, as there is no need to change a battery.

For details, see p. 40.

Full range of communication functions

- PLC Link (supports MEWNET-W0)
- MEWTOCOL-COM Master/Slave
- MODBUS RTU Master/Slave
- Program controlled communication via TOOL or COM port

For details, see p. 87.

Extended online editing functions

Additional functions now ensure that programs can be corrected without stopping the system. The online edit mode is no longer limited to 512 steps. Instead, entire programs can be downloaded to the program memory during RUN mode. Project information is written to the comment memory. Please refer to the Control FPWIN Pro online help for detailed information.

Enhanced security

The FP0R supports 8-digit passwords (alphanumeric), and offers an upload protection function as well as security functions for the FP Memory Loader.

For details, see p. 218.

FP0 compatibility

The FP0 compatibility mode enables programs that have been used on an existing FP0 to be activated on the FP0R with no further modifications. Also, since both units have an identical shape and terminal layout, there is no need to check the installation space or change the wiring.

For details, see p. 24.

1.2 Unit types

1.2.1 CPU

The operating voltage and the rated input voltage is 24V DC for all CPU types.

16k types (program capacity: 16k steps)

Type	I/Os [1]	Output	Connection	COM port	Product no.
C10	10 (6/4)	Relay	Terminal block	—	AFP0RC10RS
				RS232C	AFP0RC10CRS
				RS485	AFP0RC10MRS
C14	14 (8/6)			—	AFP0RC14RS
				RS232C	AFP0RC14CRS
				RS485	AFP0RC14MRS
C16	16 (8/8)	Transistor (NPN): 0.2A	MIL connector	—	AFP0RC16T
		Transistor (PNP): 0.2A		—	AFP0RC16P
		Transistor (NPN): 0.2A		RS232C	AFP0RC16CT
				RS485	AFP0RC16MT
		Transistor (PNP): 0.2A		RS232C	AFP0RC16CP
				RS485	AFP0RC16MP

[1] Total number (input points/output points)

32k types (program capacity: 32k steps)

Type	I/Os [1]	Output	Connection	COM port	Product no.
C32	32 (16/16)	Transistor (NPN): 0.2A	MIL connector	—	AFP0RC32T
		Transistor (PNP): 0.2A		—	AFP0RC32P
		Transistor (NPN): 0.2A		RS232C	AFP0RC32CT
				RS485	AFP0RC32MT
		Transistor (PNP): 0.2A		RS232C	AFP0RC32CP
				RS485	AFP0RC32MP
T32 (built-in battery)		Transistor (NPN): 0.2A		RS232C	AFP0RT32CT
				RS485	AFP0RT32MT
		Transistor (PNP): 0.2A		RS232C	AFP0RT32CP
				RS485	AFP0RT32MP
F32 (built-in FRAM)		Transistor (NPN): 0.2A		RS232C	AFP0RF32CT
				RS485	AFP0RF32MT
		Transistor (PNP): 0.2A		RS232C	AFP0RF32CP
				RS485	AFP0RF32MP

[1] Total number (input points/output points)

1.2.2 FP0/FP0R I/O expansion units

Type	I/Os	Power supply	Input	Output	Connection	Product no.
E8	8 (8/–)	–	24V DC ±COM terminal	–	MIL connector	FP0R-E8X
	8 (4/4)	24V DC	24V DC ±COM terminal	Relay: 2A	Terminal block	FP0R-E8RS
	8 (–/8)	24V DC	–	Relay: 2A	Terminal block	FP0R-E8YRS
	8 (–/8)	–	–	Transistor (NPN): 0.3A	MIL connector	FP0R-E8YT
	8 (–/8)	–	–	Transistor (PNP): 0.3A	MIL connector	FP0R-E8YP
E16	16 (16/–)	–	24V DC ±COM terminal	–	MIL connector	FP0R-E16X
	16 (8/8)	24V DC	24V DC ±COM terminal	Relay: 2A	Terminal block	FP0R-E16RS
	16 (8/8)	–	24V DC ±COM terminal	Transistor: (NPN) 0.3A	MIL connector	FP0R-E16T
	16 (8/8)	–	24V DC ±COM terminal	Transistor: (PNP) 0.3A	MIL connector	FP0R-E16P
	16 (–/16)	–	–	Transistor: (NPN) 0.3A	MIL connector	FP0R-E16YT
	16 (–/16)	–	–	Transistor: (PNP) 0.3A	MIL connector	FP0R-E16YP
E32	32 (16/16)	–	24V DC ±COM terminal	Transistor: (NPN) 0.3A	MIL connector	FP0R-E32T
	32 (16/16)	–	24V DC ±COM terminal	Transistor: (PNP) 0.3A	MIL connector	FP0R-E32P

1.2.3 FP0 intelligent units

Type	Description	Product no.	Manual
FP0 thermocouple unit	Thermocouple types: K, J, T, R (Resolution 0.1°C)	FP0-TC4	ARCT1F366
	Thermocouple types: K, J, T, R (Resolution 0.1°C)	FP0-TC8	
FP0 analog I/O unit	No. of input channels: 2 Input range (Resolution 1/4000): • Voltage: 0–5V, -10–+10V • Current: 0–20mA	FP0-A21	ARCT1F390
	No. of output channels: 1 Output range (Resolution 1/4000): • Voltage: -10–+10V • Current: 0–20mA		
FP0 A/D conversion unit	No. of input channels: 8 Input range (Resolution 1/4000): • Voltage: 0–5V, -10–+10V, -100–100mV • Current: 0–20mA	FP0-A80	ARCT1F321
FP0 D/A conversion unit	No. of output channels: 4 Output range (Resolution 1/4000): • Voltage: -10–+10V • Current: 4–20mA	FP0-A04V	ARCT1F382
		FP0-A04I	
FP0 RTD unit	Pt100, Pt1000, Ni1000 Resolution: 0.1°C/0.01°C (depending on switch setting)	FP0-RTD6	ARCT1F445

1.2.4 FP series link units

Type	Description	Power supply	Product no.	Manual
FP0 I/O link unit	Designed to make the FP0 function as a MEWNET-F slave unit (remote I/O system).	24V DC	FP0-IOL	FAF35E5
FP0 DP Slave unit	Designed to connect the PLC to PROFIBUS-DP, or it can stand alone as a remote I/O unit.	24V DC	FP0-DPS2	ACGM0123
C-NET adapter S2	RS485 adapter for connecting PLC and host via C-NET using MEWTOCOL-COM. Supplied with a 30cm FP0 TOOL port cable. A power supply is not required.	–	–	ARCT1F96
FP Web-Server 2	Designed to connect FP series PLCs to the Ethernet, to send e-mails, and present PLC data as HTML pages.	–	FP-WEB2	ARCT1F446
FP Web Expansion unit	Must be connected to FP Web-Server 2. Supplied with a USB and RS485 port.	–	FPWEBEXP	ARCT1F446

1.2.5 Power supply unit

Product name	Description	Product no.
FP power supply	Input voltage range: 100–240V DC Max. output current: 1A (24V DC)	FP-PS24-024E
	Input voltage range: 100–240V DC Max. output current: 2.5A (24V DC)	FP-PS24-060E
	Input voltage range: 100–240V DC Max. output current: 5A (24V DC)	FP-PS24-120E

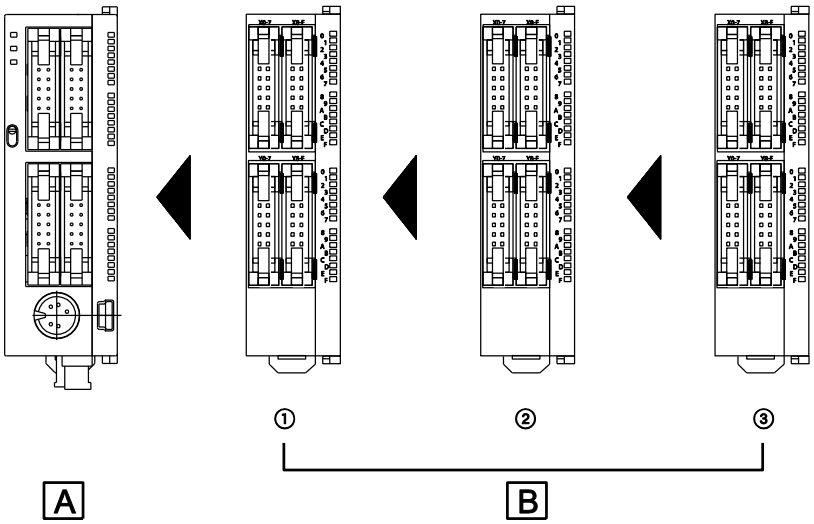
1.2.6 Accessories

Name	Description		Product no.
I/O cable	10-pin MIL wire-press socket on one side, 2 pieces (blue, white, or multi-colored)	1m	AFP0521D AFP0521BLUED AFP0521COLD
		3m	AFP0523D AFP0523BLUED
FP0 power supply cable for expansion units	Maintenance part (packed with FP0/FP0R expansion units)	1m	AFP0581
FP0R/FPΣ power supply cable	Maintenance part (packed with CPU)	1m	AFPG805
Phoenix connector (2 pcs)	Terminal block socket; maintenance parts (packed with relay output type)		AFP0802
MIL connector (2 pcs)	10-pin MIL wire-press socket; maintenance parts (packed with transistor output type)		AFP0807
Pressure connection tool	For wiring transistor output type connections		AXY5200FP
FP0 slim type mounting plate (10 pcs)	For vertical mounting of FP0/FP0R expansion units		AFP0803
Flat type mounting plate (10 pcs)	For horizontal mounting of the CPU		AFP0804
FP Memory Loader	For reading/writing programs from/to PLC	Data clear type	AFP8670
		Data hold type	AFP8671

1.3 Restrictions on unit combinations

By adding expansion units, the number of I/O points can be increased. However, the maximum number of expansion units per CPU is limited.

A maximum of three expansion units can be connected on the right side of the FP0R CPU, these expansion units being either I/O expansion units or intelligent units. A combination of relay output types and transistor output types is also possible.



A	FP0R CPU
B	Maximum expansion: 3 units
①	Expansion unit 1
②	Expansion unit 2
③	Expansion unit 3

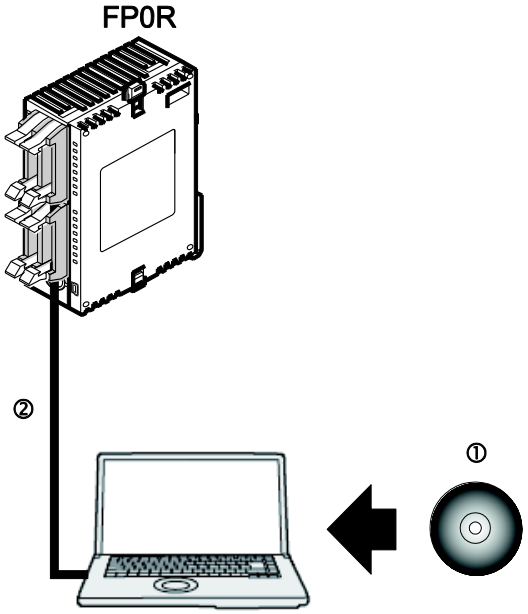
Maximum number of I/Os

C10	106
C14	110
C16	112
C32/T32/F32	128

Note

- Install the FP0 thermocouple unit to the right of other expansion units. If it is installed on the left side, overall precision will deteriorate. For details, refer to the FP0 thermocouple unit manual.
- Install the FP0 RTD unit to the right of the other expansion units.

1.4 Programming tools



①	Programming software
②	RS232C programming cable or USB cable

Programming software

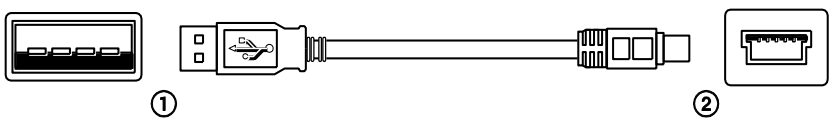
You can use the following programming software to program the FP0R:

- Control FPWIN Pro Version 6 or later
- FPWIN GR Version 2 or later
- FP Memory Loader (AFP8670/AFP8671) to transfer programs and system registers

PC connection cable

You can connect your PC to the FP0R via USB or RS232C.

Cable	Connector	Description	Product no.
USB cable	5-pin Mini-B type	USB 2.0 Full Speed (or 1.1), 2m	CABMINIUSB5D
RS232C programming cable	9-pin Sub-D to 5-pin Mini-DIN (round)	Programming cable for FP and GT series	AFC8513D



①	A type (male), PC side
②	5-pin Mini-B type (male), PLC side

Instead of Panasonic's USB cable any commercial USB cable meeting the above specifications may be used. The maximum permissible cable length is 5m.

1.5 FP0 program compatibility

Programs from the existing FP0 can only be used on the FP0R, if they:

1. conform to FP0R specifications, or
2. are executed in FP0 compatibility mode

Using programs that conform to FP0R specifications

This allows you to make maximum use of FP0R performance and functions. However, the following modifications to the FP0 program must be made before downloading the program to the PLC:

1. Change the PLC type from FP0 to FP0R using the programming software.
2. Since the system registers will be initialized when the PLC type is changed, reconfigure the system registers if necessary.
3. Modify the programs according to the FP0R specifications if necessary.

Executing programs in FP0 compatibility mode

The FP0 compatibility mode allows you to use existing FP0 programs as they stand. With a few exceptions, the same specifications apply as for the FP0.

To enter the FP0 compatibility mode, use your programming tool to download the FP0 program. A confirmation message will appear, and the mode will automatically change to the FP0 compatibility mode. The FP0 program may either have been uploaded from an FP0 or it may have been created on an FP0R in FP0 mode (PLC type is FP0).

The FP0 compatibility mode is supported by FPWIN Pro V6.10 or higher, and FPWIN GR V2.80 or higher.

Note

Due to the FP0R's higher operation speed, the scan time in FP0 compatibility mode may be shorter than the original FP0 scan time. If you require a scan time close to the original conditions, set a constant scan time in the system registers or add a dummy program, e.g. a loop operation, to increase the scan time.

An FP0 program can only run in FP0 compatibility mode, if the PLC types (C10, C14, C16, C32, and T32) match exactly. FP0 compatibility mode is not available for the F32 type FP0R.

In most respects, the FP0 programs do not need to be modified to be executable in FP0 compatibility mode. Please note, however, the following differences between the specifications, and change the programs as necessary:

1. P13_EPWT, EEPROM write instruction

   The execution times for this instruction will vary, depending on the number of write blocks.

No. of write blocks (words)	FP0 [ms]	FP0 compatibility mode [ms]
1 (64)	≈5	≈100
2 (128)	≈10	≈100
4 (256)	≈20	≈100
8 (512)	≈40	≈100
16 (1024)	≈80	≈100
32 (2048)	≈160	≈100
33 (2112)	≈165	≈200
41 (2624)	≈205	≈200
64 (4096)	≈320	≈200
96 (6144)	≈480	≈300
256 (16320)	≈800	≈800

2. F170_PulseOutput_PWM, PWM output instruction

   The frequency settings differ. In particular, the setting for the low-frequency band cannot be defined.

K	FP0		FP0 compatibility mode	
	Frequency [Hz]	Period [ms]	Frequency [Hz]	Period [ms]
8	0.15	6666.7	Cannot be specified (error occurs)	
7	0.3	3333.3		
6	0.6	1666.7		
5	1.2	833.3		
4	2.4	416.7		
3	4.8	208.3	6	166.7
2	9.5	105.3	10	100
1	19	52.6	20	50
0	38	26.3	40	25
16	100	10.0	100	10
15	200	5.0	200	5
14	400	2.5	400	2.5
13	500	2.0	500	2
12	714	1.4	750	1.3
11	1000	1.0	1000	1

3. Data size differs for elapsed value and target value

   FP0: 24 bits
   FP0 compatibility mode: 32 bits

4. F144_TRNS, serial data communication

   When sending data, note the following differences:
   FP0: The send buffer stores the number of bytes to be sent. This number is decremented during transmission.
   FP0 compatibility mode: The number of bytes to be sent remains unchanged during transmission. It is set to 0 after transmission has been completed. The number of bytes that can be sent is restricted to 2048.

5. F169_PulseOutput_Jog, JOG operation

   There are two differences between the FP0 and the FP0R specifications:
   Count mode: The FP0R does not support the "no counting" setting. Instead, incremental counting is performed with the FP0 pulse output instructions set to "no counting".
   Pulse width specification: In FP0 compatibility mode, the duty ratio is fixed at 25%. Differing settings in the FP0 programs will be ignored.

6. F168_PulseOutput_Home, Home return

   In FP0 compatibility mode, the elapsed value is counted during home return operations. With the FP0, the elapsed value is indefinite. In both cases, the elapsed value will be reset to 0 when home return has been completed.

7. Real number calculation process

   Since the accuracy of real number calculation has been improved, the calculation results obtained in the FP0 compatibility mode may differ from the results obtained in the existing FP0 program.

8. If the secondary battery installed in the T32 type is out of charge, the next power-on process will be different:

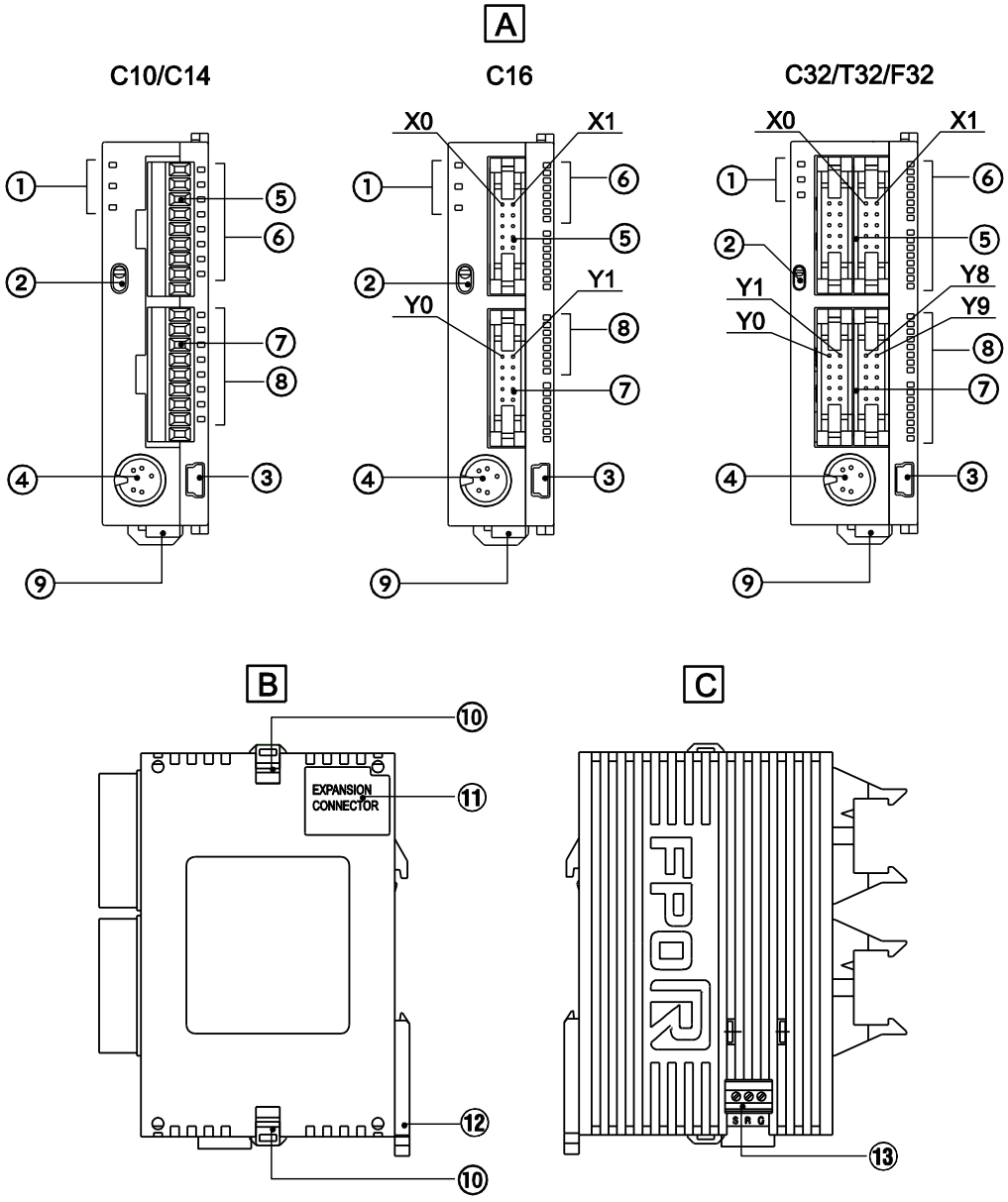
   FP0: The value in the hold area of the data memory will be unstable.
   FP0 compatibility mode: The value in the hold area of the data memory will be cleared to 0.

9. The sampling trace function is not available in FP0 compatibility mode.

Chapter 2

CPU types

2.1 CPU parts and functions



A	Front view
B	Right side view
C	Left side view

① Operation status LEDs

Display the current operation mode or the occurrence of an error.

LED	Description
RUN (green)	Lights when in RUN mode and indicates that the program is being executed.
	Flashes during forced input/output (RUN and PROG. LEDs flash alternately).
PROG. (green)	Lights when in PROG mode and indicates that operation has stopped.
	Flashes during forced input/output (RUN and PROG. LEDs flash alternately).
ERROR/ALARM (red)	Flashes when an error is detected by the self-diagnostic function (ERROR).
	Lights if a hardware error occurs, or if operation slows because of the program, and the watchdog timer is activated (ALARM).

② Operation mode selector

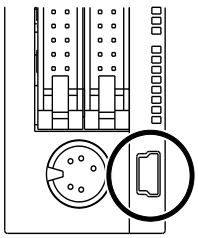
Used to change the operation mode of the PLC.

Switch position	Operation mode
RUN (upward)	Sets RUN mode. The program is executed and operation begins.
PROG. (downward)	Sets PROG mode. Operation stops. In this mode, programming via the TOOL port is possible.

When performing remote switching with the programming tool, the position of the operation mode selector and the actual operation mode may differ. Verify the mode with the operation status LED. Otherwise, restart the FP0R and set the operation mode using the operation mode selector.

③ USB port (5-pin Mini-B type)

Used to connect a programming tool.
Panasonic's USB cable CABMINIUSB5D or a commercial USB2.0 AB type cable can be used.



To use the USB port, you must install the USB driver (see p. 93).

④ TOOL port (RS232C)

Used to connect a programming tool.

For details, see p. 92.

⑤ Input connector

⑥ Input status LEDs

⑦ Output connector

⑧ Output status LEDs

⑨ Power supply connector (24V DC)

Use the power supply cable provided. Product no.: AFPG805

⑩ Expansion hook

Used to secure an expansion unit. The hook is also used for installation on the flat type mounting plate (part no. AFP0804).

⑪ Connector for FP0/FP0R expansion units

Connects an FP0/FP0R expansion unit to the internal circuit. The connector is located under the seal.

⑫ DIN rail attachment lever

Used for easy attachment to a DIN rail. The lever is also used for installation on a slim type mounting plate. See "Using optional mounting plates" on p. 61.

⑬ COM port (RS232C or RS485)

Used to enable communication with external devices, e.g. a programmable display.

2.2 CPU input specifications

The input specifications below apply to all FP0R CPU types.

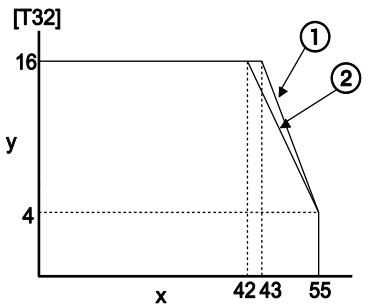
Item		Description
Insulation method		Optical coupler
Rated input voltage		24V DC
Operating voltage range		21.6–26.4V DC
Rated input current		≈2.6mA
Input points per common		C10: 6 C14, C16: 8 C32, T32, F32: 16 (Either the positive or negative pole of the input power supply can be connected to the common terminal.)
Min. ON voltage/min. ON current		19.2V DC/2mA
Max. OFF voltage/max. OFF current		2.4V DC/1.2mA
Input impedance		9.1kΩ
Response time	FALSE → TRUE	≤20μs (see note) An input time constant (0.1ms–64ms) can be set using the system registers.
	TRUE → FALSE	
Operation indicator		LEDs

Note

This specification applies when the rated input voltage is 24V DC and the temperature is 25°C.

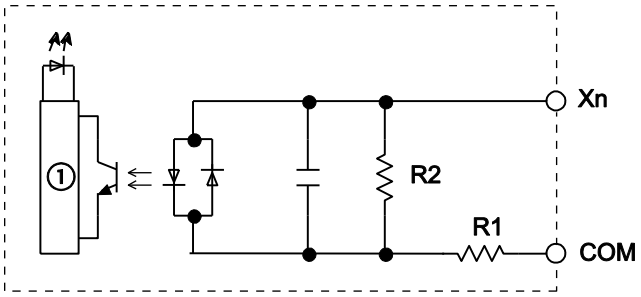
Inputs that are TRUE simultaneously

Keep the number of inputs per common which are simultaneously TRUE within the following range as determined by the ambient temperature.



x	Ambient temperature [°C]
y	Number of inputs per common that are TRUE simultaneously
①	At 24V DC
②	At 26.4V DC

Internal circuit diagram



①	Internal circuit
R1	9.1kΩ
R2	1kΩ

2.3 CPU output specifications

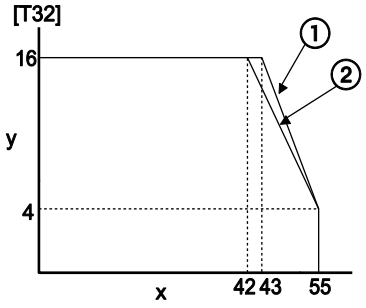
Transistor types

These output specifications apply to the CPU types C32 and C28.

Item		Description	
		NPN	PNP
Insulation method		Optical coupler	
Output type		Open collector	
Rated load voltage		5V DC–24V DC	24V DC
Operating load voltage range		4.75–26.4V DC	21.6–26.4V DC
Max. load current		0.2A	
Outputs per common		C16: 8 C32, T32, F32: 16	
OFF state leakage current		≤1μA	
ON state voltage drop		≤0.2V DC	
Response time	FALSE → TRUE	≤20μs (Load current: ≥5mA) ≤0.1ms (Load current: ≥0.5mA)	
	TRUE → FALSE	≤40μs (Load current: ≥5mA) ≤0.2ms (Load current: ≥0.5mA)	
External power supply for internal circuit (+ and - terminals)	Voltage	21.6–26.4V DC	
	Current	C16: ≤30mA C32, T32, F32: ≤60mA	C16:≤35mA C32, T32, F32: ≤70mA
Surge absorber		Zener diode	
Operation indicator		LEDs	

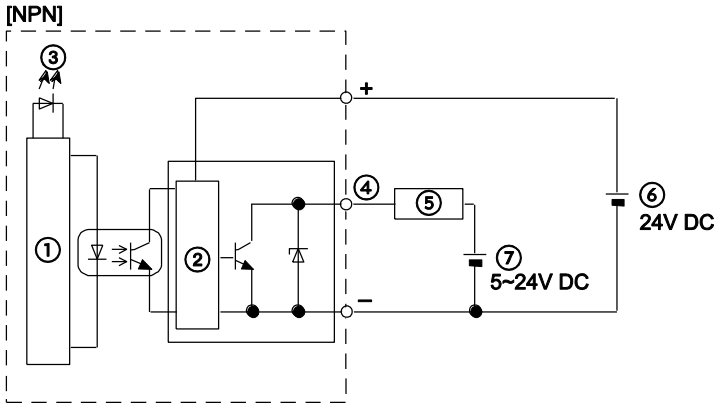
Outputs that are TRUE simultaneously

Keep the number of outputs per common which are simultaneously TRUE within the following range as determined by the ambient temperature.

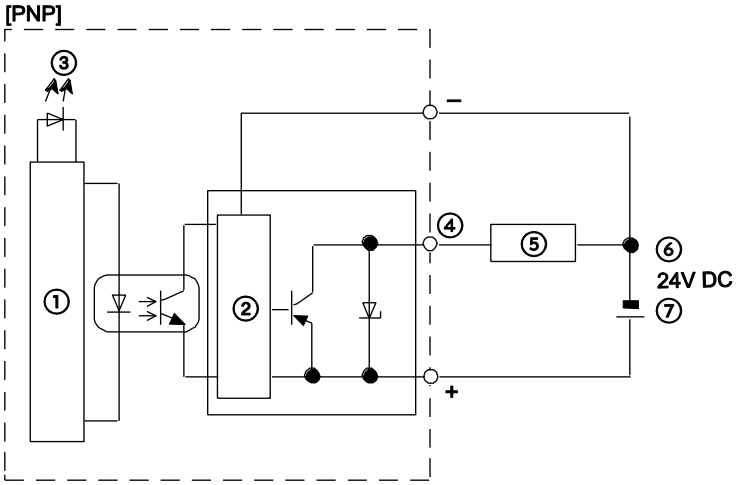


x	Ambient temperature [°C]
y	Number of outputs per common that are TRUE simultaneously
①	At 24V DC
②	At 26.4V DC

Internal circuit diagram



①	Internal circuit	⑤	Load
②	Output circuit	⑥	External power supply
③	Output status LED	⑦	Load power supply
④	Output		

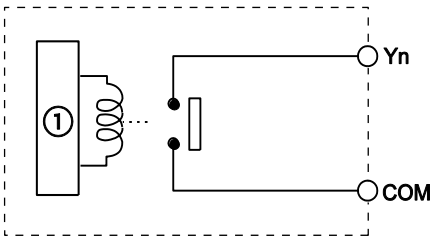


①	Internal circuit	⑤	Load
②	Output circuit	⑥	External power supply
③	Output status LED	⑦	Load power supply
④	Output		

Relay types (C10/C14)

Item		Description
Output type		1a output
Nominal switching capacity (resistive load)		2A 250V AC, 2A 30V DC (≤4.5A/common)
Outputs per common		C10: 2+1+1 C14: 4+1+1
Response time	FALSE → TRUE	≈10ms
	TRUE → FALSE	≈8ms
Mechanical lifetime		≥20 000 000 operations (switching frequency: 180 operations/min)
Electrical lifetime		≥100 000 operations (switching frequency at nominal switching capacity: 20 operations/min)
Surge absorber		–
Operation indicator		LEDs

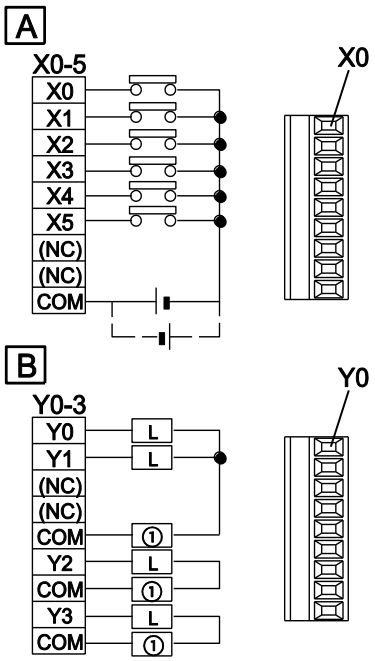
Internal circuit diagram



①	Internal circuit

2.4 Terminal layout

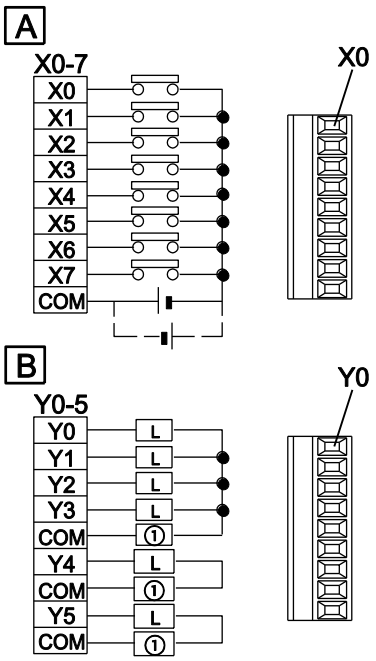
C10RS, C10CRS, C10RM, C10CRM



(The above illustration is the terminal block type.)

A	Input
B	Output
①	Power supply

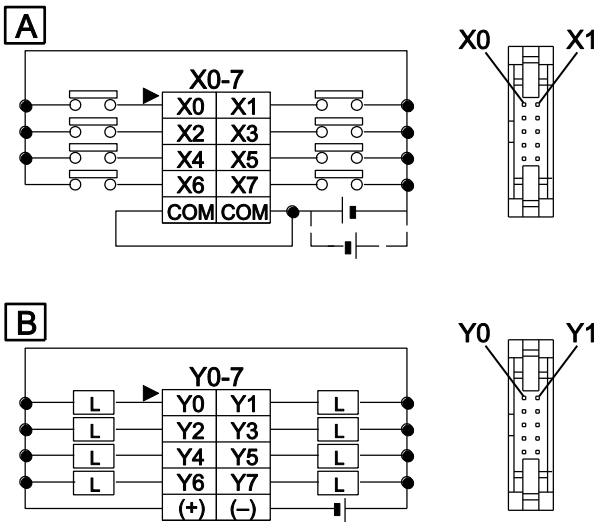
C14RS, C14CRS, C14RM, C14CRM



(The above illustration is the terminal block type.)

A	Input
B	Output
①	Power supply

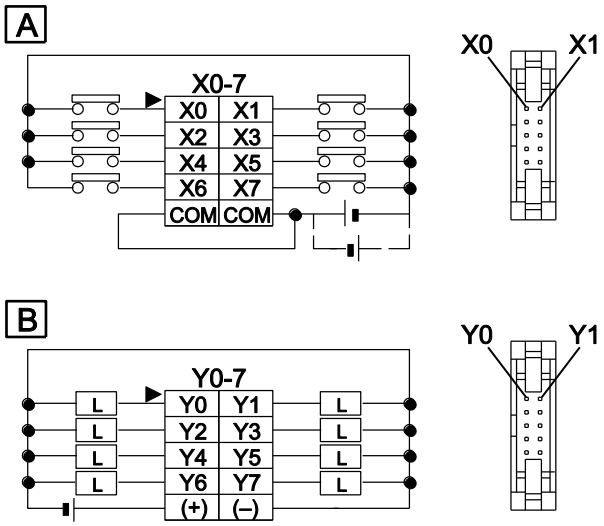
C16T, C16CT



The COM terminals of the input circuits are connected internally.

A	Input
B	Output

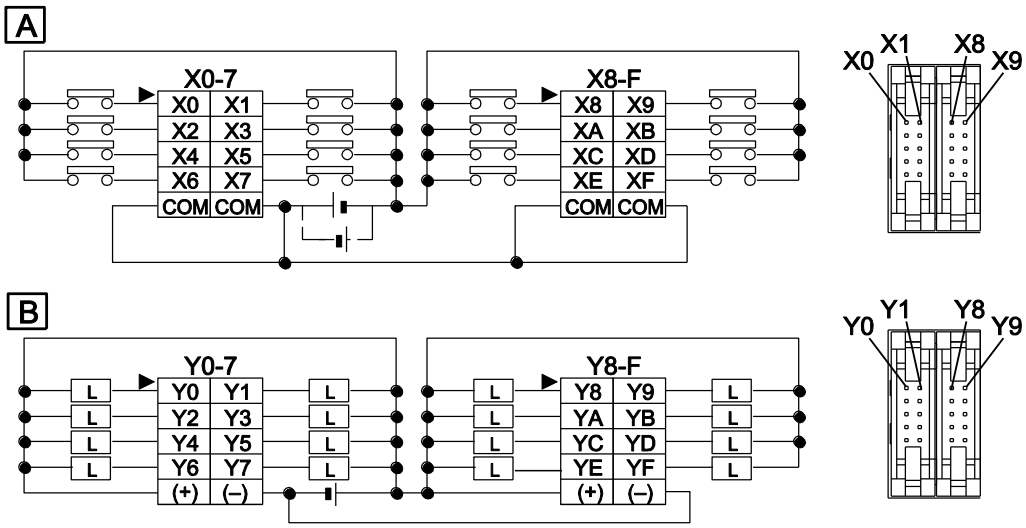
C16P, C16CP



The COM terminals of the input circuits are connected internally.

A	Input
B	Output

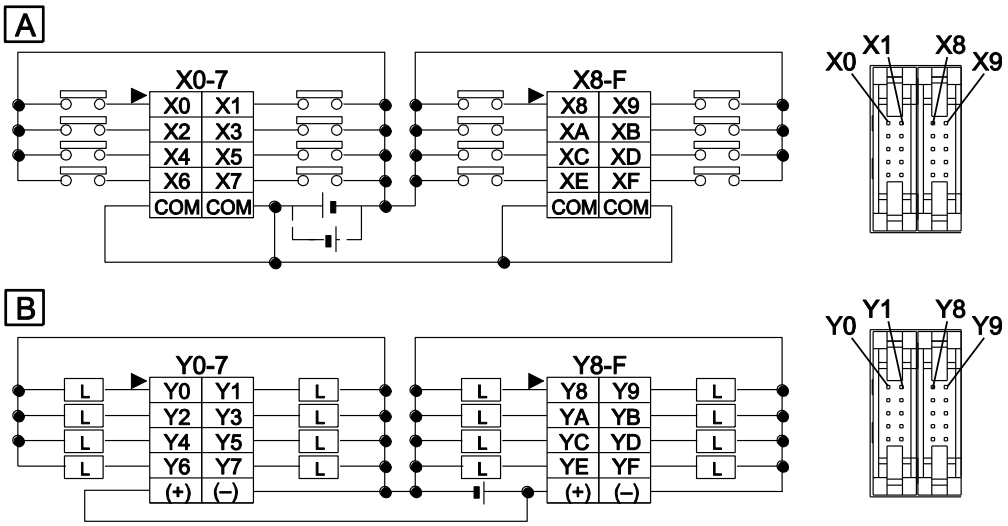
C32T, C32CT, T32CT, F32CT



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

A	Input
B	Output

C32P, C32CP, T32CP, F32CP



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

A	Input
B	Output

2.5 Backup and clock/calendar functions

The FP0R-T32 CPU is equipped with a secondary battery (charging type). This battery makes it possible to use:

- additional hold areas for data registers or other data
- the clock/calendar function

The FP0R-F32 CPU has a built-in FRAM, which allows saving all data without a backup battery. The FP0R-F32 type does not offer a clock/calendar function.

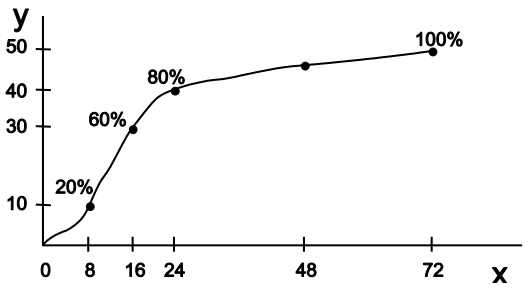
Charging the battery

The built-in backup battery is not charged when the unit is shipped. Charge the battery sufficiently before use.

Charging time for full charge: 72 hours (at an ambient temperature of 25°C)

The battery will be charged automatically when the DC power is supplied to the CPU.

The number of days the backup battery remains effective (backup time) depends on the charging time. If the battery has been fully charged (72 hours at an ambient temperature of 25°C), the battery will function for approx. 50 days.



x	Charging time (hours)
y	Backup time (days at 25°C)

The backup time will vary according to the ambient temperature when the battery is charged.

Ambient temperature when charged:	Backup time:
70°C	≈14 days
25°C	≈50 days
-20°C	≈25 days

Predicted life of built-in backup battery

The life of the built-in backup battery varies depending on the ambient temperature while the CPU is on (energized). The temperature when the CPU is off (not powered) has little influence on the battery life.

Ambient temperature	Lifetime of built-in backup battery
55°C	≈430 days (≈1 year)
45°C	≈1200 days (≈3 years)
40°C	≈2100 days (≈6 years)
35°C	≈3300 days (≈9 years)
≤34°C	≈10 years

The built-in backup battery cannot be replaced.

Precision of clock/calendar

Ambient temperature	Error
0°C	<104s/month
25°C	<51s/month
55°C	<155s/month

2.5.1 Backup function

Additional hold areas which will be saved with a backup battery (FP0R-T32) or with the built-in FRAM (FP0R-F32) can be specified for the following memory areas:

- Timers/Counter (T/C)
- Internal flags (R)
- Data registers (DT)
- Step ladders

Programs and system register settings will be held in the internal ROM regardless of the built-in backup battery.

Specifying hold areas

If no settings are made in system registers 6 to 14, the default address ranges will be saved when the PLC is turned off. To save additional hold areas, follow the procedure below.

Procedure

1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click "Hold on/off"

NOTICE

If the battery is empty the data values in the hold area will become indefinite during power-off. They are cleared to 0 the next time the power is turned on.
We recommend adding a program for checking if the data is set to 0 when the power is turned on the next time.

2.5.2 Clock/calendar function

As the initial clock/calendar values are indefinite, write the values using a programming tool.

2.5.2.1 Memory area for clock/calendar function

With the clock/calendar function, clock and calendar data stored in special data registers DT90053 to DT90057 can be read and used in sequence programs. To access special data registers and special internal flags, use the PLC-independent system variables.

Hour and minute data (DT90053) can only be read. All other data can be read and written.

Special data register	FPWIN Pro system variable	Upper byte	Lower byte
DT90053	sys_wClockCalendarHourMin	Hour data 16#00–16#23	Minute data 16#00–16#59
DT90054	sys_wClockCalendarMinSec	Minute data 16#00–16#59	Second data 16#00–16#59
DT90055	sys_wClockCalendarDayHour	Day data 16#01–16#31	Hour data 16#00–16#23
DT90056	sys_wClockCalendarYearMonth	Year data 16#00–16#99	Month data 16#01–16#12
DT90057	sys_wClockCalendarDayOfWeek	–	Day-of-the-week data 16#00–16#06
DT90058	sys_wClockCalendarSet	Bit 15=TRUE (16#8000): activates clock/calendar setting Bit 0=TRUE (16#0): sets seconds to 0	

2.5.2.2 Settings for clock/calendar function

The clock/calendar values are backed up using a battery.

This feature is available in the FP0R-T32 CPU only.

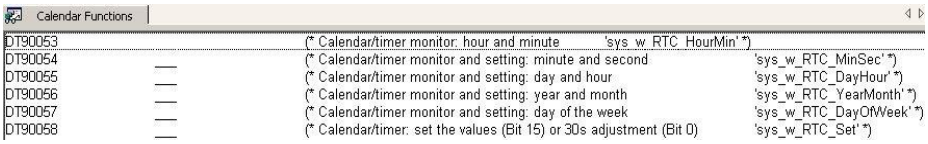
There are no default clock/calendar settings. There are two ways to set the clock/calendar function:

Procedure

Using the programming software

1. **Online** → **Online mode** or 
2. **Monitor** → **Display special registers** → **Calendar functions**
3. Enter the desired date and time values

   Confirm each value with [Enter].



Using a program

1. The date/time values are written to special data registers DT90054 to DT90057.
2. A value of 16#8000 is written to DT90058.

Note

- To access special data registers and special internal flags, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header.
- To set the clock/calendar, you can also use the instruction SET_RTC_DT or SET_RTC_INT.

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

2.5.2.3 Sample program for fixed schedule and automatic start

In this example, the clock/calendar function is used to output the Y0 signal for one second at 8:30 a.m. every day. Here, the hour/minute data stored in special data register DT90053 is used to output the signal at the appointed time. The value of DT90053 is written using the system variable sys_wClockCalendarHourMin.

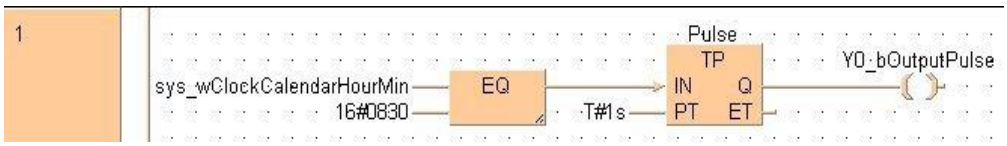
GVL

	Class	Identifier	FP Address	IEC Address	Type	Initial
0	VAR_GLOBAL	Y0_bOutputPulse	Y0	%QX0.0	BOOL	FALSE

POU Header

	Class	Identifier	Type	Initial
0	VAR	Pulse	TP	
1	VAR_EXTERNAL	Y0_bOutputPulse	BOOL	FALSE
2	VAR			

LD Body



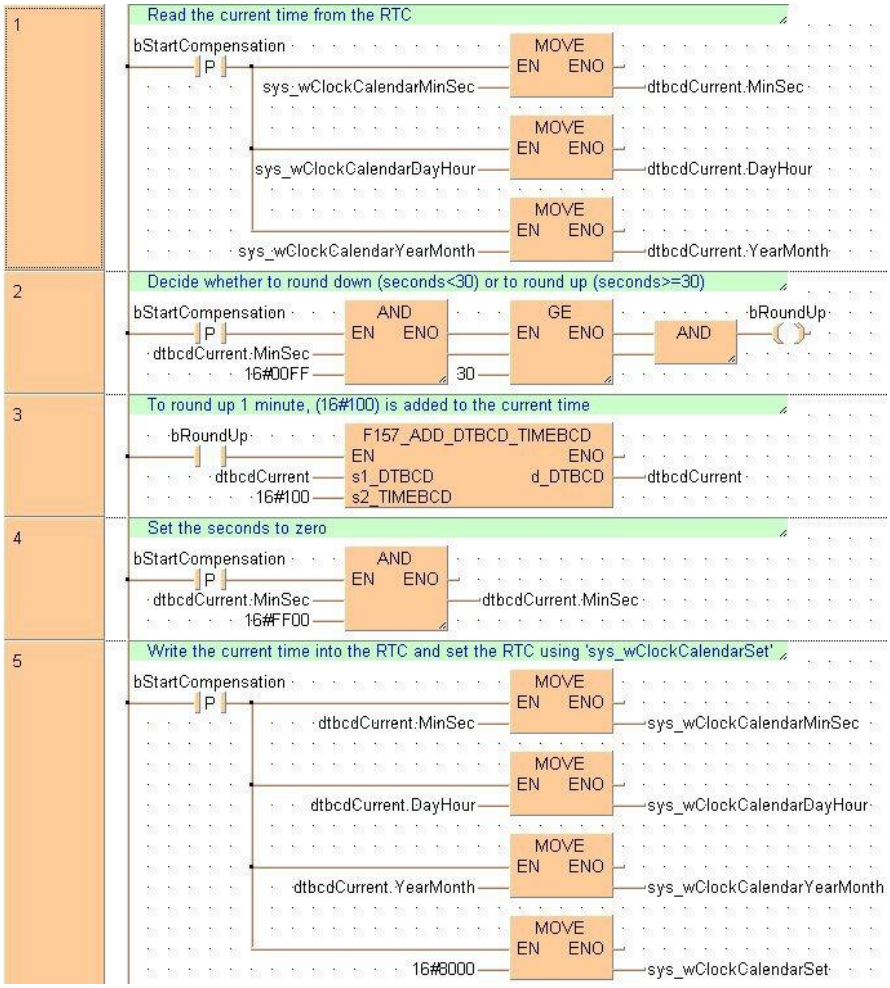
2.5.2.4 Sample program for 30-second compensation

This is a program to perform the compensation for 30 seconds when R0 turns to TRUE. If the 30-second compensation is required, use this program.

POU Header

	Class	Identifier	Type	Initial
0	VAR	bStartCompensation	BOOL	FALSE
1	VAR	bRoundUp	BOOL	FALSE
2	VAR	dtbcdCurrent	DTBCD	
3	VAR	wSec	WORD	0

LD Body



Chapter 3

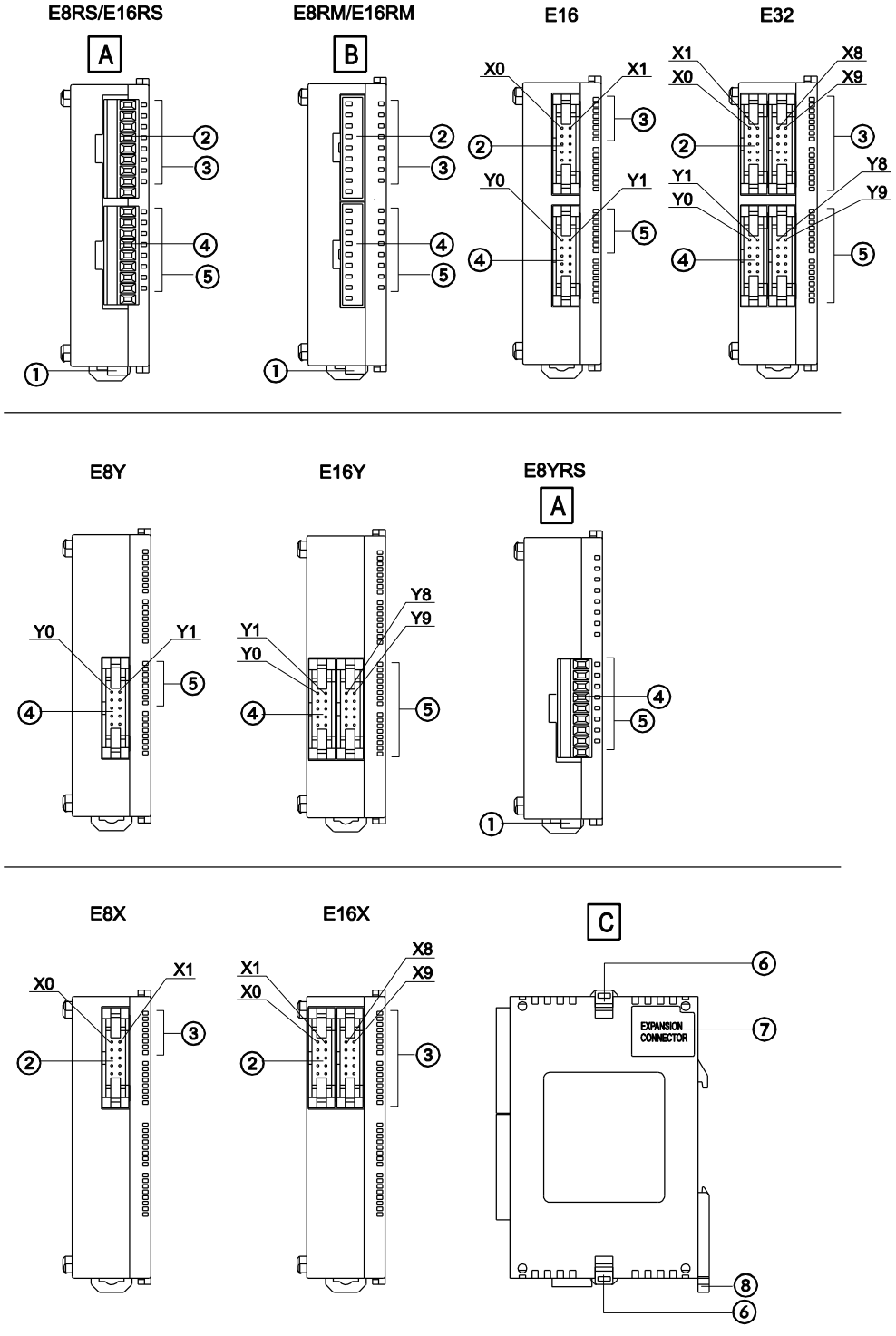
Expansion

3.1 Expansion method

You can expand the FP0R by adding FP0/FP0R I/O expansion units (see p. 19), FP0 intelligent units (see p. 20) and FP Series link units (see p. 20).

The expansion units are connected to the right side of the CPU. Use the expansion connectors and the expansion hooks on the side of each unit. See "Connecting FP0/FP0R expansion units" on p. 65.

3.2 Expansion unit parts and functions



A	Terminal block type
B	MIL connector type
C	Right side view (common to all expansion units)

① Power supply connector (24V DC)

Use the power supply cable provided. Product no.: AFP0581

② Input connector

③ Input status LEDs

④ Output connector

⑤ Output status LEDs

⑥ Expansion hook

Used to secure an expansion unit.

⑦ Connector for FP0/FP0R expansion units

Connects an FP0/FP0R expansion unit to the internal circuit. The connector is located under the seal.

⑧ DIN rail attachment lever

Used for easy attachment to a DIN rail. The lever is also used for installation on a slim type mounting plate. See "Slim type mounting plate" on p. 61.

3.3 Expansion input specifications

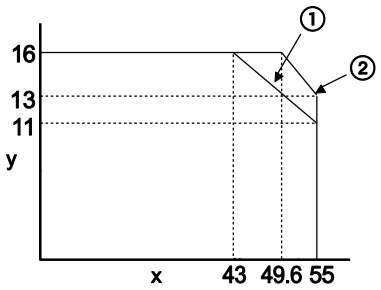
Item		Description
Insulation method		Optical coupler
Rated input voltage		24V DC
Rated input current		≈4.7mA (at 24V DC) (≈4.3mA for FP0 unit)[1)]
Input impedance		≈5.1kΩ (≈5.6kΩ for FP0 unit)[1)]
Operating voltage range		21.6–26.4V DC
Input points per common		E8X/E16P/E16T/E32RS: 8 E32T/E16X: 16 E8R: 4 (Either the positive or negative pole of the input power supply can be connected to the common terminal.)
Min. ON voltage/min. ON current		19.2V DC/3mA
Max. OFF voltage/max. OFF current		2.4V DC/1mA
Response time	FALSE → TRUE	≤2ms
	TRUE → FALSE	
Operation indicator		LEDs

1) All FP0 expansion units have been replaced by newer FP0R units with improved specifications.

Inputs that are TRUE simultaneously

Keep the number of inputs per common which are simultaneously TRUE within the following range as determined by the ambient temperature.

E32



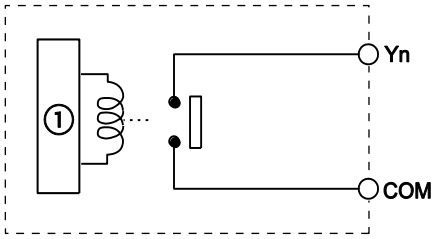
x	Ambient temperature [°C]
y	Number of inputs per common that are TRUE simultaneously
①	At 24V DC
②	At 26.4V DC

3.4 Expansion output specifications

Relay output specifications (E8RS/E8RM/E8YRS/E16RS/E16RM/E32RS)

Item		Description
Output type		1a
Nominal switching capacity (resistive load)		2A 250V AC, 2A 30V DC (≤4.5A/common)
Outputs per common		E8R: 4 E16R/E8YR/E32RS: 8
Response time	FALSE → TRUE	≈10ms
	TRUE → FALSE	≈8ms
Mechanical lifetime		≥20 000 000 operations (switching frequency: 180 operations/min)
Electrical lifetime		≥100 000 operations (switching frequency at nominal switching capacity: 20 operations/min)
Surge absorber		–
Operation indicator		LEDs

Internal circuit diagram



① Internal circuit

Transistor output specifications (NPN: E8YT/E16YT/E16T/E32T, PNP: E8YP/E16YP/E16P/E32P)

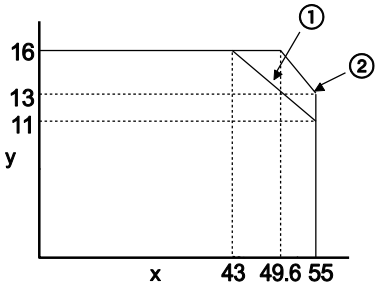
Item		Description	
		NPN	PNP
Insulation method		Optical coupler	
Output type		Open collector	
Rated load voltage		5V DC–24V DC	24V DC
Operating load voltage range		4.75–26.4V DC	21.6–26.4V DC
Max. load current		0.3A/point (max. 1A/common) (0.1A for FP0 unit)[1)]	
Max. surge current		0.3A	
Outputs per common		E16T/E8Y: 8 E32/E16Y: 16	
OFF state leakage current		≤100μA	
ON state voltage drop		≤1.5V	
Response time	FALSE → TRUE	≤1ms	
	TRUE → FALSE	≤1ms	
External power supply for internal circuit	Voltage	21.6–26.4V DC	
	Current	3mA/point	
Surge absorber		Zener diode	
Operation indicator		LEDs	

1) All FP0 expansion units have been replaced by newer FP0R units with improved specifications.

Outputs that are TRUE simultaneously

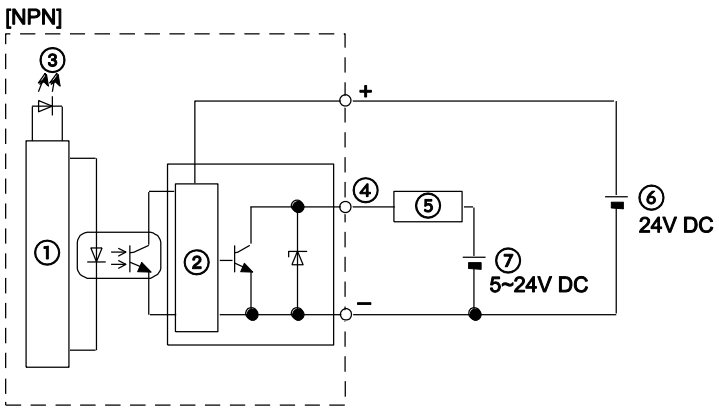
Keep the number of outputs per common which are simultaneously TRUE within the following range as determined by the ambient temperature.

E32

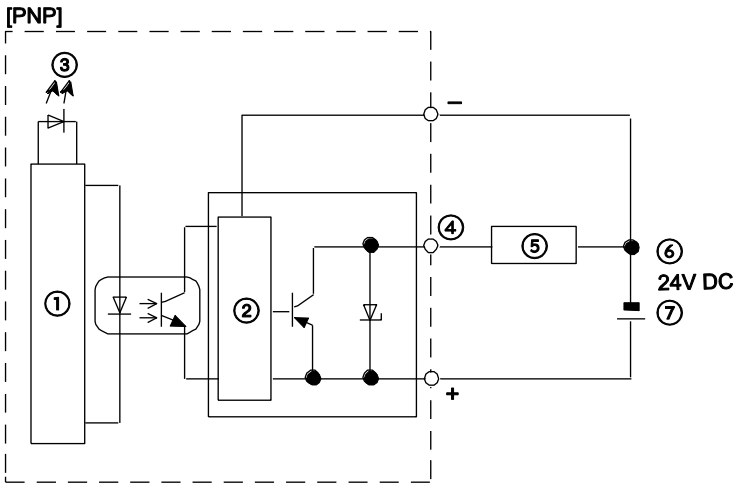


x	Ambient temperature [°C]
y	Number of outputs per common that are TRUE simultaneously
①	At 24V DC
②	At 26.4V DC

Internal circuit diagram



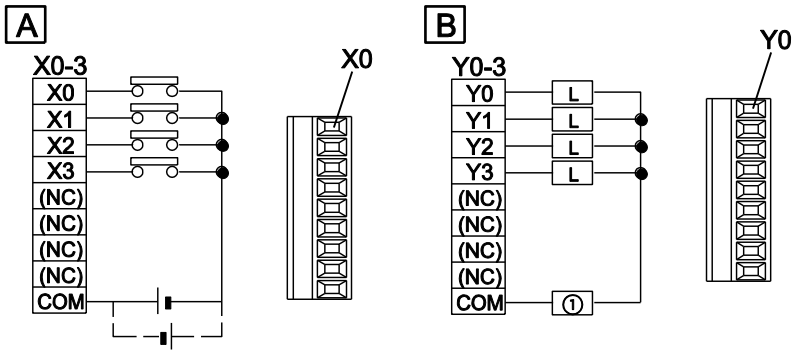
①	Internal circuit	⑤	Load
②	Output circuit	⑥	External power supply
③	Output status LED	⑦	Load power supply
④	Output		



①	Internal circuit	⑤	Load
②	Output circuit	⑥	External power supply
③	Output status LED	⑦	Load power supply
④	Output		

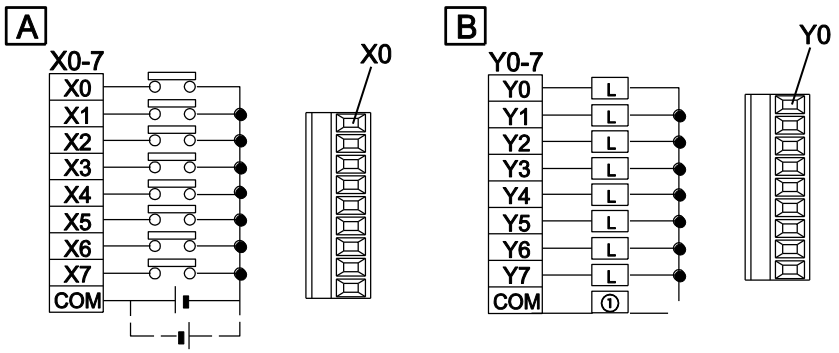
3.5 Terminal layout

E8RS, E8RM



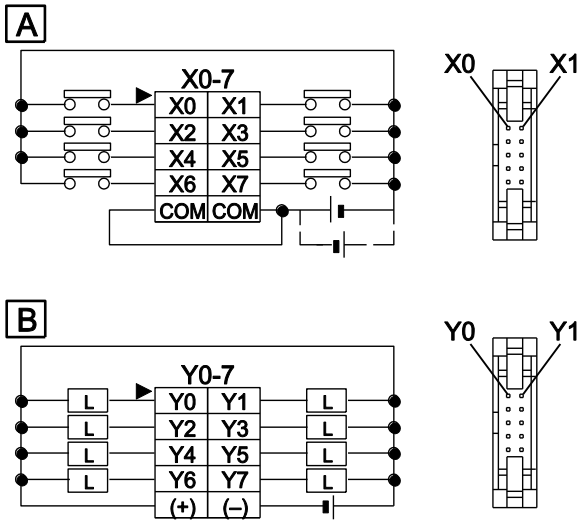
A	Input
B	Output
①	Power supply

E16R, E8YRS, E32RS



A	Input (no input for E8YRS)
B	Output
①	Power supply

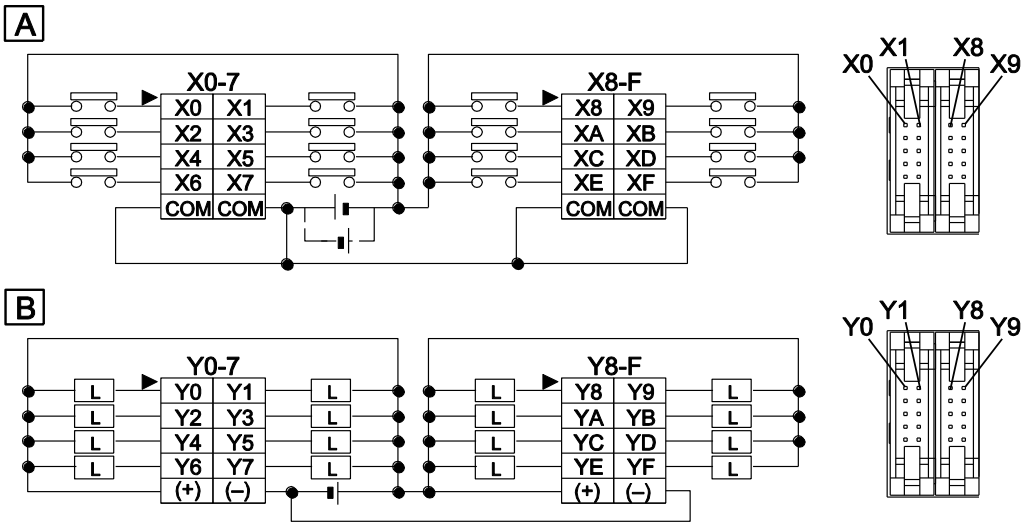
E8X, E16T, E8YT



The COM terminals of the input circuits are connected internally.

A	Input (no input for E8YT)
B	Output (no output for E8X)

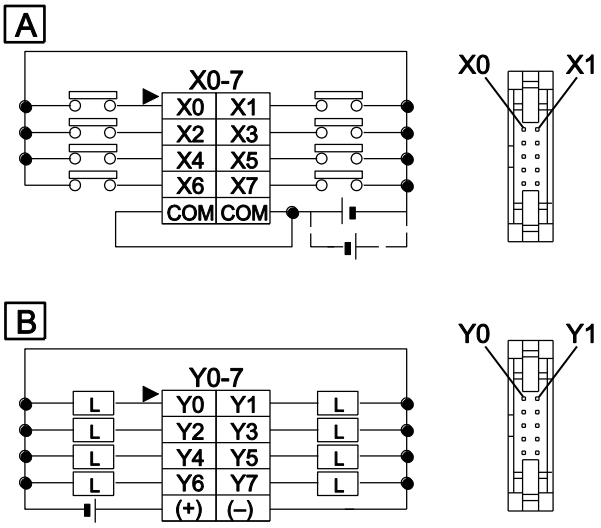
E16X, E32T, E16YT



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

A	Input (no input for E16YT)
B	Output (no output for E16X)

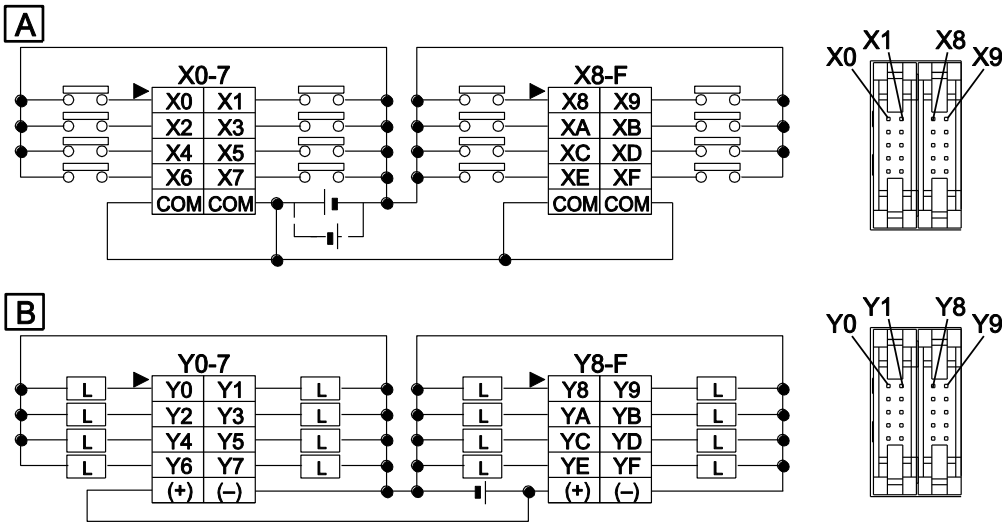
E16P, E8YP



The COM terminals of the input circuits are connected internally.

A	Input (no input for E8YT)
B	Output

E32P, E16YP



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

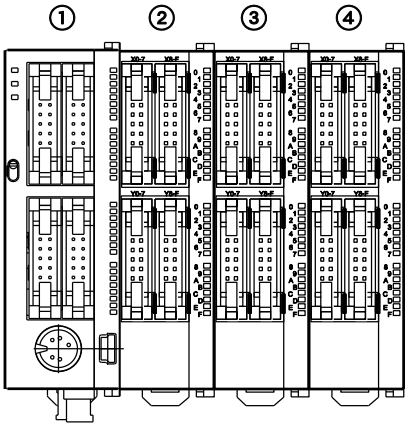
A	Input (no input for E16YP)
B	Output

Chapter 4

I/O allocation

4.1 General

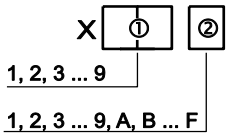
I/O allocation is performed automatically when an expansion unit is added and is determined by the installation location. The I/O allocation of the FP0R CPU is fixed.



Type of unit	Unit number		I/O addresses
FP0R CPU	①	–	X0–XF Y0–YF
FP0/FP0R I/O expansion unit	②	1	X20–X3F Y20–Y3F
	③	2	X40–X5F Y40–Y5F
	④	3	X60–X7F Y60–Y7F

Note

- The input X and output Y are expressed as a combination of decimal (①) and hexadecimal (②) numbers:



- On the FP0R and the FP0, the same numbers are used for inputs and outputs, e.g. X20, Y20.
- The usable I/O addresses depend on the unit type. See "FP0/FP0R expansion units" on p. 56.

4.2 CPU

The I/O allocation of the FP0R CPU is fixed.

CPU type		I/Os	I/O addresses
C10	Input	6	X0–X5
	Output	4	Y0–Y3
C14	Input	8	X0–X7
	Output	6	Y0–Y5
C16	Input	8	X0–X7
	Output	8	Y0–Y7
C32/T32/F32	Input	16	X0–XF
	Output	16	Y0–YF

4.3 FP0/FP0R expansion units

I/O allocation is performed automatically when an expansion unit is added and is determined by the installation location. Expansion units from the FP0/FP0R series are connected on the right side of the CPU. The I/O addresses are allocated from the unit nearest to the CPU in ascending order.

Type of unit		I/Os	Channel	Unit number (installation location)		
				1	2	3
FP0/FP0R I/O expansion unit						
FP0R-E8X	Input	8	–	X20–X27	X40–X47	X60–X67
FP0R-E8R	Input	4	–	X20–X23	X40–X43	X60–X63
	Output	4	–	Y20–Y23	Y40–Y43	Y60–Y63
FP0R-E8YR, E8YT, E8YP	Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
FP0R-E16X	Input	16	–	X20–X2F	X40–X4F	X60–X6F
FP0R-E16R, E16T, E16P	Input	8	–	X20–X27	X40–X47	X60–X67
	Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
FP0R-E16YT, E16YP	Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
FP0R-E32T, E32P, E32RS	Input	16	–	X20–X2F	X40–X4F	X60–X6F
	Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
FP0 analog I/O unit FP0-A21	Input	16	0	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
	Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)

Type of unit		I/Os	Channel	Unit number (installation location)		
				1	2	3
FP0 A/D conversion unit FP0-A80 and FP0 thermocouple unit FP0-TC4, FP0-TC8	Input	16	0, 2, 4, 6	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1, 3, 5, 7	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
FP0 D/A conversion unit FP0-A04V, FP0-A04I	Input	16	–	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Output	16	0, 2	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
	Output	16	1, 3	WY3 (Y30–Y3F)	WY5 (Y50–Y5F)	WY7 (Y70–Y7F)
FP0 RTD unit FP0-RTD6	Input	16	0, 2, 4	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1, 3, 5	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
	Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 I/O link unit FP0-IOL	Input	32	–	X20–X3F	X40–X5F	X60–X7F
	Output	32	–	Y20–Y3F	Y40–Y5F	Y60–Y7F

Note

The data for each channel of the A/D and D/A conversion units FP0-A80, FP0-TC4/TC8, FP0-A04V/I, and FP0-RTD6 is converted and loaded with a user program that includes a switching flag to convert the data in 16-bit words (see corresponding manuals).

Chapter 5

Installation and wiring

5.1 Installation

Please follow the installation instructions carefully to prevent failure or malfunctions.

5.1.1 Installation environment and space

Operating environment

After installing the unit, make sure to use it within the range of the general specifications:

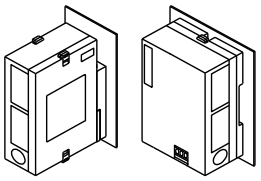
- Ambient temperature: 0–+55°C
- Ambient humidity: 10%–95% RH (at 25°C, non-condensing)
- Pollution degree: 2
- Do not use the unit in the following environments:
  - Direct sunlight
  - Sudden temperature changes causing condensation
  - Inflammable or corrosive gases
  - Excessive airborne dust, metal particles or salts
  - Benzine, paint thinner, alcohol or other organic solvents or strong alkaline solutions such as ammonia or caustic soda
  - Vibration, shock or direct drop of water
  - Influence from power transmission lines, high voltage equipment, power cables, power equipment, radio transmitters, or any other equipment that would generate high switching surges. Maintain at least 100mm of space between these devices and the unit.

Static electricity

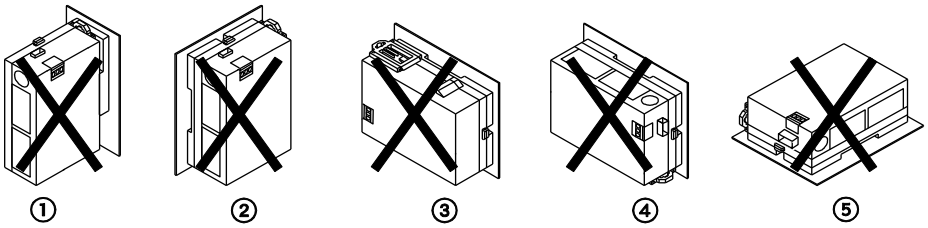
Before touching the unit or equipment, always touch some grounded metal to discharge any static electricity you may have generated (especially in dry locations). The discharge of static electricity can damage parts and equipment.

Measures regarding heat discharge

- Always install the CPU orientated with the TOOL port facing outward on the bottom in order to prevent the generation of heat.



Do **NOT** install the CPU as shown below.

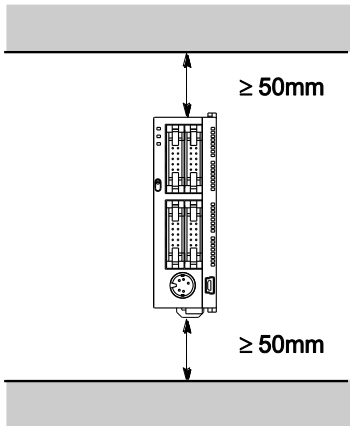


①	Upside-down
②	Upside-down
③	Input and output connectors face down
④	Input and output connectors on top
⑤	Horizontal installation of the unit

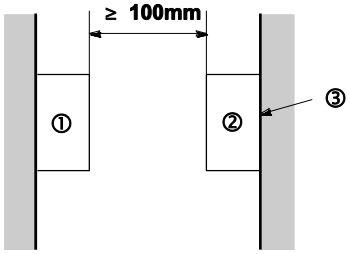
- Do not install the unit above devices which generate heat such as heaters, transformers or large-scale resistors.

Installation space

- Leave at least 50mm of space between the wiring ducts of the unit and other devices to allow heat radiation and unit replacement.



- Maintain a minimum of 100mm between devices to avoid adverse effects from noise and heat when installing a device or panel door to the front of the unit.



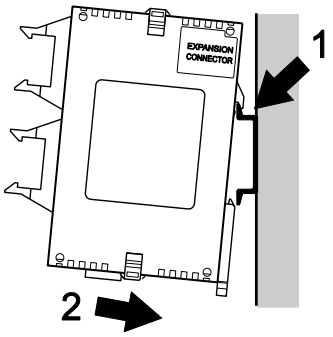
①	PLC
②	Other device
③	Panel door

- Leave at least 100 mm of space from the CPU's front surface to allow room for programming tool connections and wiring.

5.1.2 Using DIN rails

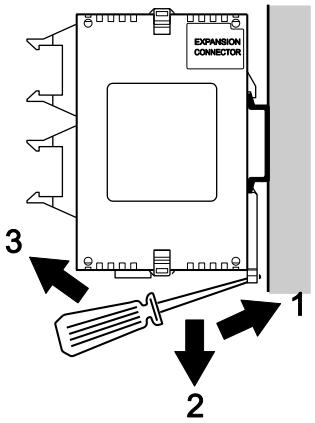
The CPU can be easily attached to DIN rails.

Procedure



1. Fit upper hook of unit onto DIN rail
2. Without moving upper hook, press on lower hook to fit unit into position

Removal is very simple, too:



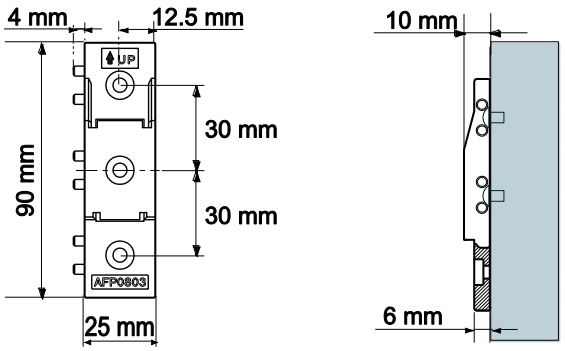
1. Insert slotted screwdriver into DIN rail attachment lever
2. Pull attachment lever downwards
3. Lift up unit and remove from rail

5.1.3 Using optional mounting plates

Use M4 size pan-head screws for attaching the mounting plate to the mounting panel. The diagrams below show the dimensions of the mounting plates.

5.1.3.1 Slim type mounting plate

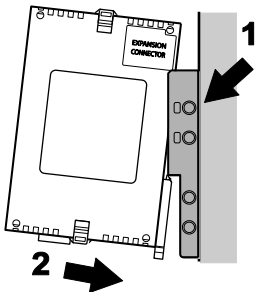
The mounting plate AFP0803 can be used alternatively to DIN-rail mounting.



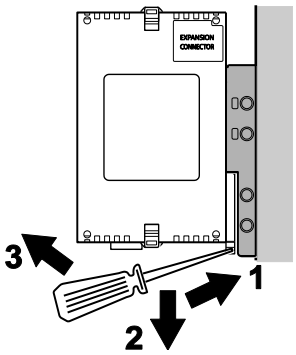
Installation and removal

Installation and removal of the unit is similar to the procedure using DIN rails:

Installation:



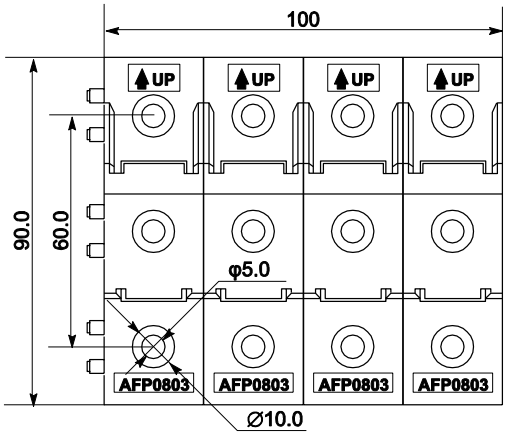
Removal:

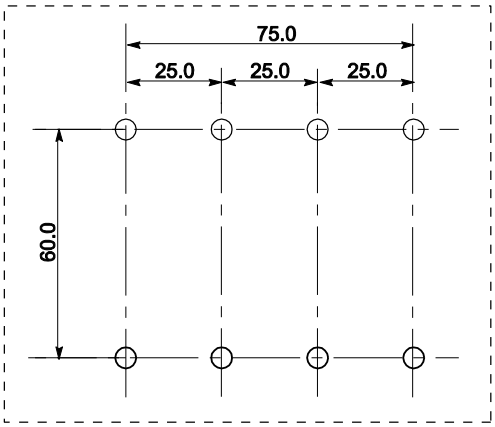


Combining mounting plates

When combining several mounting plates, tighten the screws after joining all of the mounting plates to be connected. Tighten all corner screws.

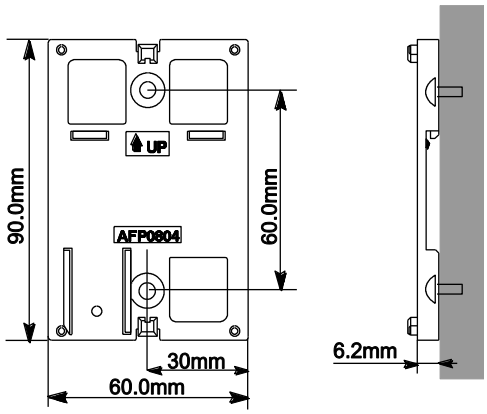
The following diagram shows the combination of the mounting plates AFP0803 when the maximum number of expansion units is used and the mounting hole dimensions:



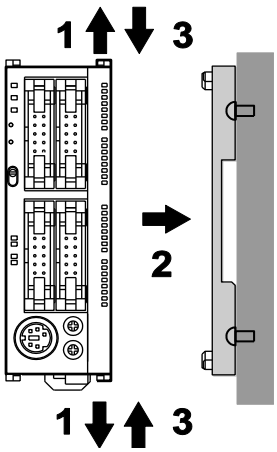


5.1.3.2 Flat type mounting plate

The flat type mounting plate (AFP0804) should only be used with a stand-alone CPU. It should not be used if an expansion unit is attached to the CPU.



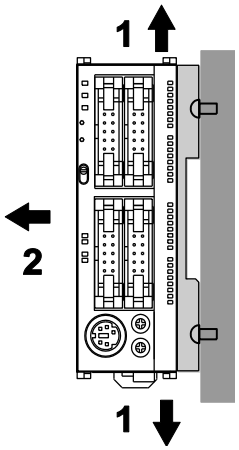
Installation



Procedure

1. Raise expansion hooks on top and bottom of the unit
2. Press unit on mounting plate and align expansion hooks with plate
3. Push expansion hooks back into place

Removal

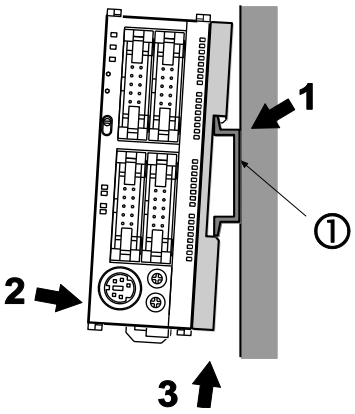


Procedure

1. Raise expansion hooks on top and bottom of the unit
2. Remove unit from mounting plate

Attachment to DIN rail

A unit with an attached flat type mounting plate can also be installed sideways on a DIN rail.



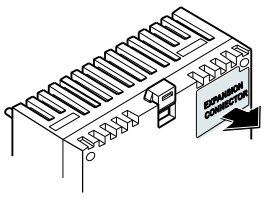
① DIN rail

5.2 Connecting FP0/FP0R expansion units

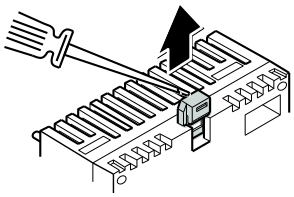
The expansion units are connected to the right side of the CPU. Use the expansion connectors and the expansion hooks on the side of each unit.

Procedure

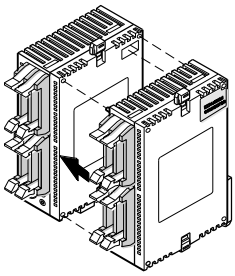
1. Peel seal on right side of the unit to expose expansion connector



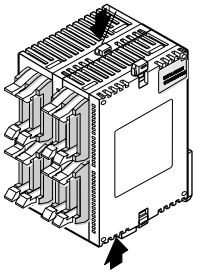
2. Raise expansion hooks on top and bottom of the unit



3. Align pins and holes in all four corners



4. Insert pins into holes so that there is no gap between units
5. Push expansion hooks back into place



You can now add up to two more units in the same manner.

5.3 Safety instructions for wiring

In certain applications, malfunction may occur for the following reasons:

- Power ON timing differences between the PLC system and input/output devices or mechanical power apparatus
- A response time lag when a momentary power drop occurs
- Abnormality in the PLC, external power supply circuit, or other devices

In order to prevent a malfunction that results in a system shutdown, choose the adequate safety measures listed below:

Interlock circuit

When a motor's clockwise/counter-clockwise operation is controlled, provide an interlock circuit that prevents clockwise and counter-clockwise signals from being input into the motor at the same time.

Emergency stop circuit

Provide an emergency stop circuit externally to turn off the power supply of controlled devices in order to prevent a system shutdown or an irreparable accident if a malfunction occurs.

Start-up sequence

The PLC should be operated only after all of the field devices are energized. To ensure this sequence, the following measures are recommended:

- Turn ON the PLC with the operation mode selector set to PROG mode, and then switch to RUN mode
- Program the PLC so as to disregard the inputs and outputs until the field devices are energized

Note

When stopping the operation of the PLC, have the input/output devices turned off after the PLC has stopped operating.

Grounding

When installing the PLC next to devices that generate high voltages from switching, such as inverters, do not ground them together. Use an exclusive ground for each device.

Momentary power failures

The FP0R continues to operate normally for a certain period of time in case of a momentary power failure. We call this the momentary power off time. However, if the power failure exceeds this period of time, operation depends on the combination of units, the power supply voltage, etc. In some cases, operation mirrors a power supply reset.

For the momentary power off time values, see "General specifications" on p. 231.

Protection of power supply

An insulated power supply with an internal protective circuit should be used (FP power supply). The power supply for the CPU is a non-insulated circuit, so if an incorrect voltage is directly applied, the internal circuit may be damaged or destroyed.

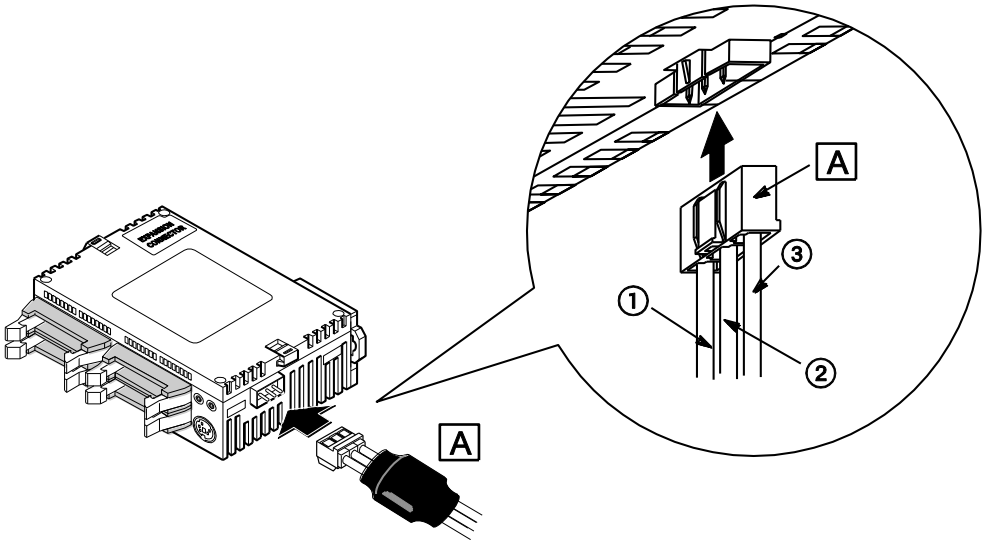
If using a power supply device without an internal protective circuit, always make sure power is supplied to the unit through a protective element such as a fuse.

Protection of output sections

If current exceeding the nominal switching capacity is being supplied in the form of a motor lock current or a coil shorting in an electromagnetic device, a protective element such as a fuse should be attached externally.

5.4 Wiring the power supply

Use the power supply cable provided. Attach as shown.



A	Power supply cable (AFPG805)
①	Brown: 24V DC
②	Blue: 0V
③	Green: function earth

Specifications

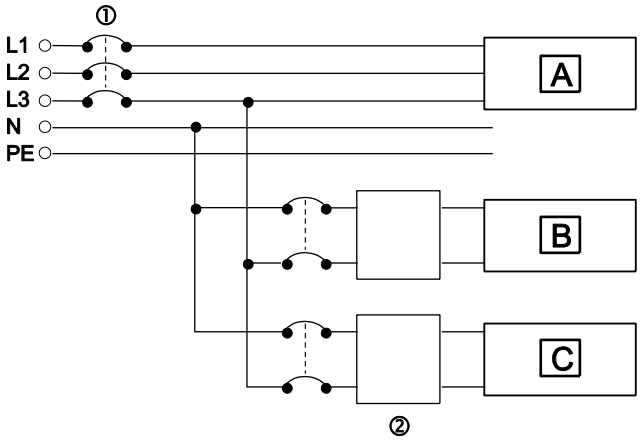
Rated voltage:	24V DC
Operating voltage range:	21.6–26.4V DC

Note

- To minimize adverse effects from noise, twist the brown and blue wires of the power supply cable.
- To protect the system against faulty voltages from the power supply line, use an insulated power supply with an internal protective circuit.
- The regulator on the unit is a non-insulated type.
- If using a power supply device without an internal protective circuit, always make sure power is supplied to the unit through a protective element such as a fuse.

Isolation of power supply systems

Insulate the wiring systems to the CPU, input/output devices, and mechanical power apparatus.



A	Mechanical power apparatus
B	Input/output devices
C	CPU
①	Circuit breaker
②	Insulated DC power supply

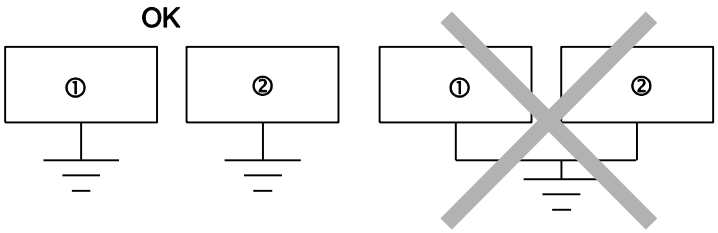
Power supply sequence

- Make sure the power supply of the CPU turns off before the power supply for input and output. If the power supply for input and output is turned off first, the CPU will detect the input fluctuations and may begin an unexpected operation.
- Be sure to supply power to a CPU and an expansion unit from the same power supply, and turn the power on and off simultaneously for both.

5.4.1 Grounding

If necessary, ground the instrument to increase the noise resistance.

- For grounding purposes, use wiring with a minimum of 2mm^2. The grounding connection should have a resistance of less than 100Ω.
- The point of grounding should be as close to the PLC as possible. The ground wire should be as short as possible.
- Always use an exclusive ground for PLCs and other devices. If two devices share a single ground point, it may produce an adverse effect.



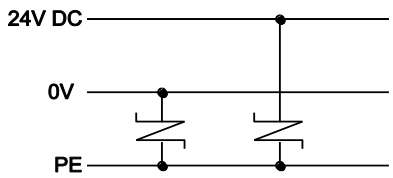
①	PLC
②	Other device (inverter etc.)

Risk of short circuits

Depending on the surroundings in which the equipment is used, grounding may cause problems.

Example 1:

Since the power supply line of the FP0/FP0R expansion unit (24V DC and 0V terminal) is connected to the function earth through a varistor, the varistor may be shorted if there is an irregular potential between the power supply line and function earth. (The power supply line of the FP0R is connected to function earth through a high-voltage capacitor. Therefore, there is no risk of a short-circuit.)

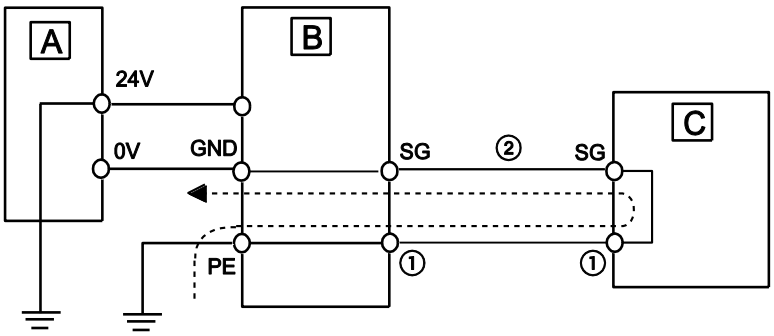


Power supply line of FP0R with built-in 39V varistor

Example 2:

Do not ground the function earth terminal of the FP0R when grounding a plus terminal (+) of the power supply.

In some computers, the SG terminal of the RS232C port is connected with the connector shielding. Also, the FP0R TOOL port shielding is connected with the function earth terminal (PE). Therefore, the GND and function earth terminals of the FP0R may be connected if a computer is connected. Especially when the FP0R is connected to a computer with a plus terminal (+) grounded, the minus terminal (-) of an FP0R is connected with the function earth terminal. A resulting short circuit may damage the FP0R and its neighboring parts.



A	Power supply	①	Shielding
B	CPU	②	Cable
C	Computer		

5.5 Input and output wiring

Note

- Separate the input/output wires from the power and high voltage wires by at least 100mm.
- Be sure to select the thickness (diameter) of the input and output wires while taking into consideration the required current capacity.
- Arrange the wiring so that the input and output wiring are separated and so that these wirings are separated from the power wiring as much as possible. Do not route them through the same duct or wrap them up together.

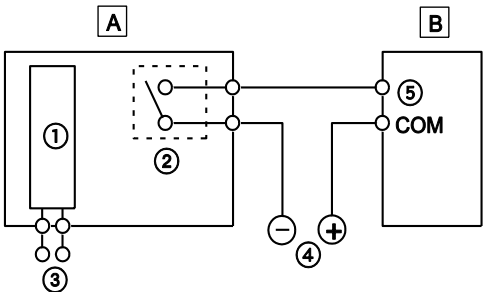
5.5.1 Input wiring

For connecting input devices see the diagrams and recommendations given below.

5.5.1.1 Photoelectric and proximity sensors

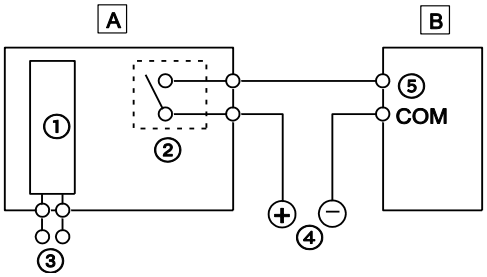
Relay output type

Sink input:



A	Sensor
B	FP0R
①	Internal circuit
②	Flag
③	Power supply for sensor
④	Power supply for input
⑤	Input terminal

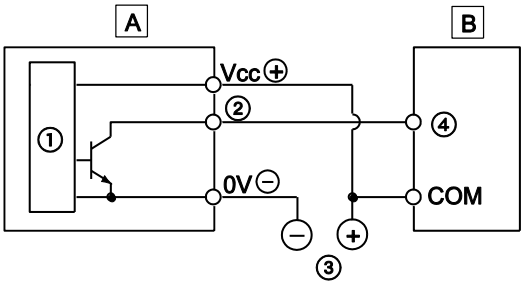
Source input:



A	Sensor
B	FP0R
①	Internal circuit
②	Flag
③	Power supply for sensor
④	Power supply for input
⑤	Input terminal

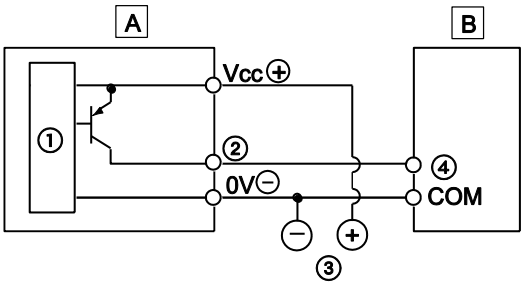
Open collector output type

Sink output (NPN):



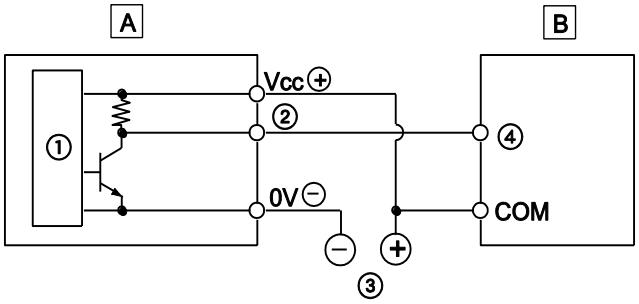
A	Sensor
B	FP0R
①	Internal circuit
②	Output
③	Power supply for input
④	Input terminal

Source output (PNP):



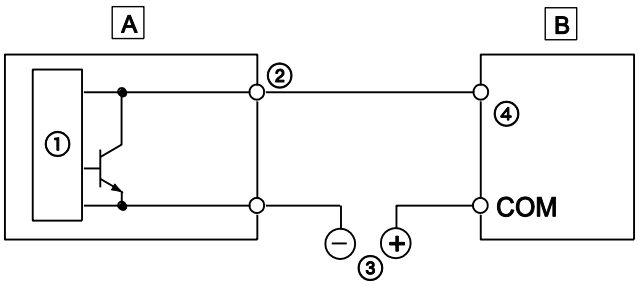
A	Sensor
B	FP0R
①	Internal circuit
②	Output
③	Power supply for input
④	Input terminal

Voltage output (universal output) type



A	Sensor
B	FP0R
①	Internal circuit
②	Output
③	Power supply for input
④	Input terminal

Two-wire output type

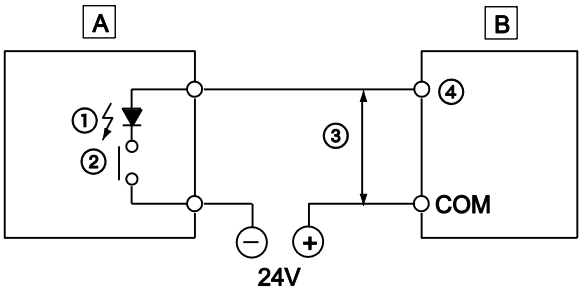


A	Sensor
B	FP0R
①	Internal circuit
②	Output
③	Power supply for input
④	Input terminal

5.5.1.2 Input wiring precautions

When using an LED-equipped Reed switch

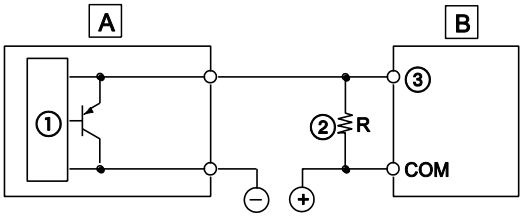
When an LED is connected in series to an input contact such as an LED-equipped Reed switch, make sure that the ON voltage applied to the PLC input terminal is greater than 21.6V DC. In particular, take care when connecting a number of switches in series.



A	LED-equipped Reed switch
B	FP0R
①	LED
②	Contact
③	≥21.6V
④	Input terminal

When using a two-wire type sensor

If the input of the PLC does not turn off because of leakage current from the two-wire type sensor (photoelectric sensor or proximity sensor), the use of a bleeder resistor is recommended, as shown below.



A	Two-wire type sensor
B	FP0R
①	Internal circuit
②	Bleeder resistor
③	Input terminal

The formula is based on an input impedance of 9.1kΩ. The input impedance varies depending on the input terminal number.

The off voltage of the input is 2.4V. Therefore, select a bleeder resistor value R so that the voltage between the COM terminal and the input terminal will be less than 2.4V.

$$I \times \frac{9.1R}{9.1 + R} \leq 2.4$$

Therefore:

$$R \leq \frac{21.84}{9.1I - 2.4} \ (k\Omega)$$

The power dissipation P of the resistor is:

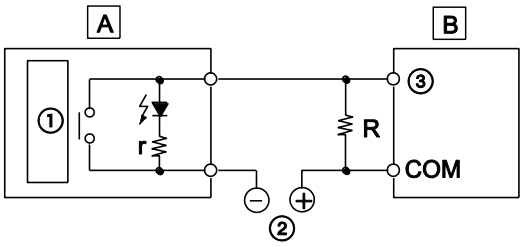
$$P = \frac{(V)^2}{R} \ [W]$$

V = Power supply voltage

In the actual selection, use a value that is 3 to 5 times the value of P.

When using an LED-equipped limit switch

If the input of the PLC does not turn off because of the leakage current from the LED-equipped limit switch, the use of a bleeder resistor is recommended, as shown below.



A	LED-equipped limit switch
B	FP0R
r	Internal resistor of limit switch (kΩ)
R	Bleeder resistor (kΩ)
①	Internal circuit
②	Power supply for input
③	Input terminal

The OFF voltage of the input is 2.4V. Therefore, when the power supply is 24V, select the bleeder resistor R so that the current will be greater than the result of this formula:

$$I = \frac{24 - 2.4}{r}$$

The resistance R of the bleeder resistor is:

$$R \leq \frac{21.84}{9.1I - 2.4} \ (k\Omega)$$

The power dissipation P of the resistor is:

$$P = \frac{(V)^2}{R} \ [W]$$

V = Power supply voltage

In the actual selection, use a value that is 3 to 5 times the value of P.

5.5.2 Output wiring

There is no fuse in the output circuit. It is recommended to install external fuses in every circuit to reduce the risk of burning out the output circuit when the output is shorted.

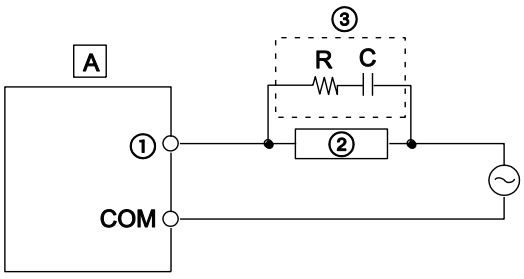
Do not connect a load that exceeds the maximum switching ability of the output terminal.

5.5.2.1 Protective circuit for inductive loads

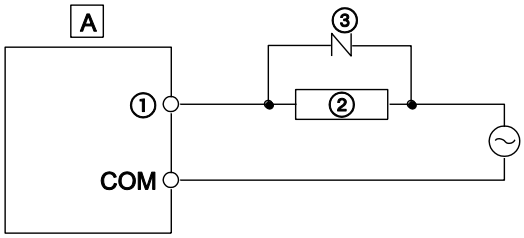
With an inductive load, a protective circuit should be installed in parallel with the load.

When switching DC inductive loads with the relay output type, be sure to connect a diode across the ends of the load.

Using an AC inductive load (relay output type)

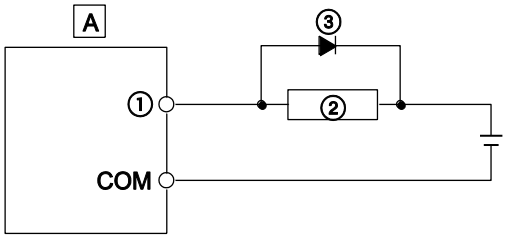


A	FP0R
①	Output terminal
②	Load
③	Surge absorber, e.g. resistance R: 50Ω, capacitance C: 0.47μF



A	FP0R
①	Output terminal
②	Load
③	Varistor

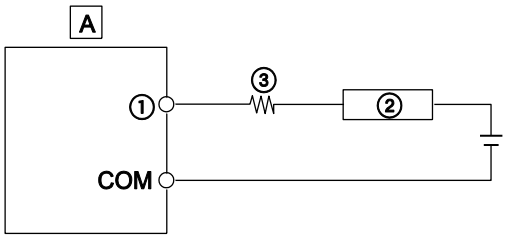
Using a DC inductive load



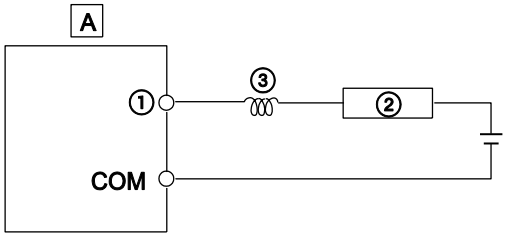
A	FP0R
①	Output terminal
②	Load
③	Diode

5.5.2.2 Protective circuit for capacitive loads

When connecting loads with large inrush currents, connect a protection circuit as shown below to minimize their effect.



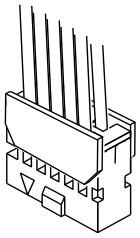
A	FP0R
①	Output terminal
②	Load
③	Resistor



A	FP0R
①	Output terminal
②	Load
③	Inductor

5.6 Wiring the MIL connector

The connector indicated below is supplied with transistor type CPUs and I/O expansion units. Use the wires indicated below. A pressure connection tool for connecting the wires is recommended.



This connector can be ordered as an accessory.

Ordering information

Product no.	Product name	Type	Packaging
AFP0807	Connector set	10-pin type	2 pieces
AXW61001	Semi-cover	10-pin type	2 pieces
AXW7221	Crimp contacts	For AWG22/24	5 pieces

Suitable wire

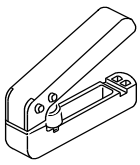
Size	Cross-sectional area [mm^2]	Insulation thickness [mm]	Rated current
AWG22	0.3	∅ 1.5–1.1mm	3A
AWG24	0.2		

Optional cables

Description	Product no.
I/O cable with 10-pin MIL connector, (2pcs: 1 × 10 blue, 1 × 10 white wires), 1m	AFP0521D
I/O cable with 10-pin MIL connector, (2pcs: 1 × 10 blue, 1 × 10 white wires), 3m	AFP0523D
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 blue wires), 1m	AFP0521BLUED
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 blue wires), 3m	AFP0523BLUED
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 colored wires), 1m	AFP0521COLD
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 colored wires), 3m	AFP0523COLD
I/O cable with 40-pin MIL connector, blue wires, 1m	AYT58403BLUED
I/O cable with 40-pin MIL connector, blue wires, 3m	AYT58406BLUED

I/O cable with 40-pin MIL connector, colored wires according to DIN 47100, 3m	AYT58406COLD

Pressure connection tool AXY5200FP

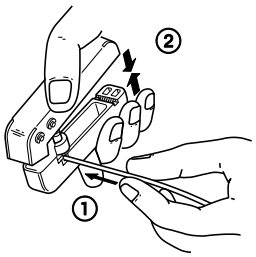


Wiring method

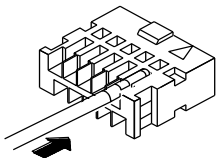
The wire end can be directly crimped without removing the wire's insulation, saving labor.

Procedure

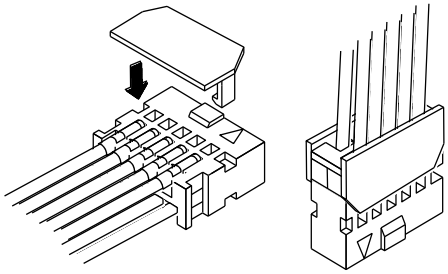
1. Insert wire without removing its insulation until it stops
2. Lightly grip tool



3. Insert press-fitted wire into connector housing

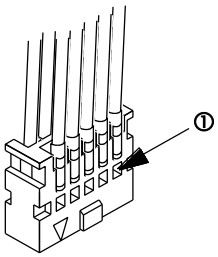


4. When all wires have been inserted, fit semi-cover into place



Note

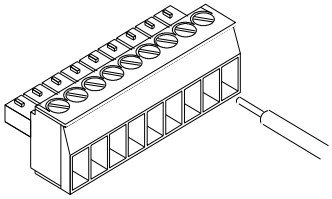
If there is a wiring mistake or the cable is incorrectly pressure-connected, the contact puller pin provided with the fitting can be used to remove the contact.



①	Press the housing against the pressure connection tool so that the contact puller pin comes in contact with this section.

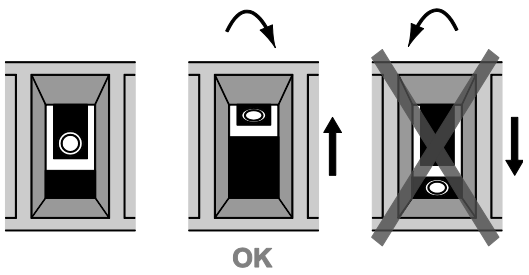
5.7 Wiring the terminal block

Screw-type terminal blocks are used. The suitable wires are given below.



Precautions

- When removing the wire's insulation, be careful not to scratch the core wire.
- Do not twist the wires to connect them.
- Do not solder the wires to connect them. The solder may break due to vibration.
- After wiring, make sure stress is not applied to the wire.
- If the socket in the terminal block closes upon counter-clockwise rotation, the connection is wrong. Disconnect the wire, check the terminal hole, and then re-connect the wire.



Terminal block

Item	Description
Number of pins	9
Manufacturer	Phoenix Contact Co.
Model	MC1,5/9-ST-3,5
Product no.	1840434

Suitable wire

Size	Cross-sectional area [mm^2]
AWG22	0.3
AWG24–16	0.2–1.25

Pole terminals with compatible insulation sleeve

For pole terminals, please consider the following specifications:

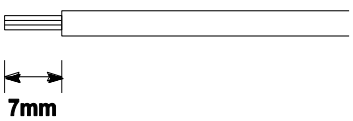
Cross-sectional area [mm^2]	Size
0.25	AWG24
0.50	AWG20
0.75	AWG18
1.00	AWG18
0.5 x 2	AWG20 (for 2 pieces)

The tightening torque should be 0.22–0.25Nm or less. Use a screwdriver with a blade size of 0.4 x 2.5.

Wiring method

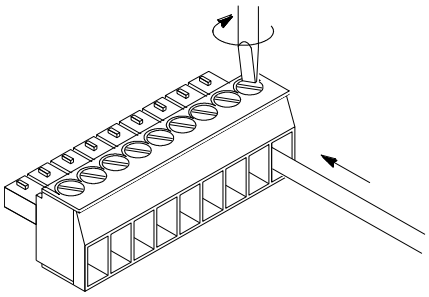
Procedure

1. Remove a portion of the wire's insulation



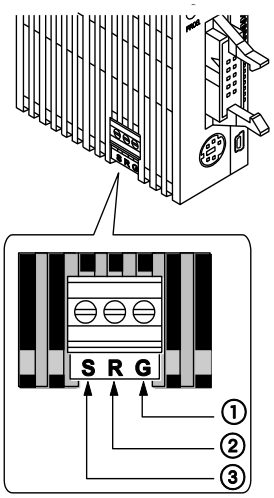
2. Insert the wire into the terminal block until it contacts the back of the socket

3. Turn the screw clockwise to fix the wire in place



5.8 Wiring the COM port

A screw-down connection type is used for the COM port. The suitable wires are given below.

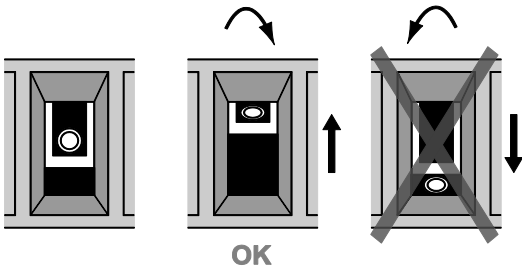


	Symbol	RS232C	RS485
①	G	Signal Ground	E terminal
②	R	Receive Data (Input)	Transmission line (-)
③	S	Send Data (Output)	Transmission line (+)

Precautions

- When removing the wire's insulation, be careful not to scratch the core wire.
- Do not twist the wires to connect them.
- Do not solder the wires to connect them. The solder may break due to vibration.
- After wiring, make sure stress is not applied to the wire.

- If the socket in the terminal block closes upon counter-clockwise rotation, the connection is wrong. Disconnect the wire, check the terminal hole, and then re-connect the wire.



Terminal block

The communication connector manufactured by Phoenix Contact is used.

Item	Description
Number of pins	3
Manufacturer	Phoenix Contact Co.
Model	MKDS1/3-3.5
Product no.	1751400

Suitable wire

Size	Cross-sectional area [mm^2]
AWG28–16	0.08–1.25

Only use shielded twisted pair cables.

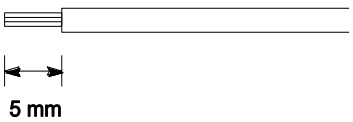
It is recommended to ground the shielded part.

When using a pole terminal, please refer to "Wiring the terminal block" on p. 81.

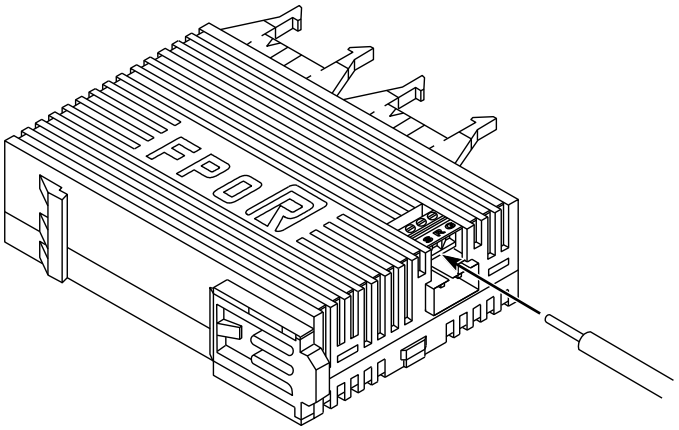
Wiring method

Procedure

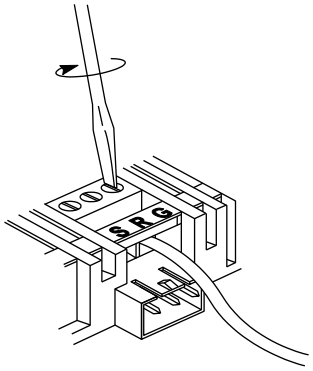
1. Remove a portion of the wire's insulation



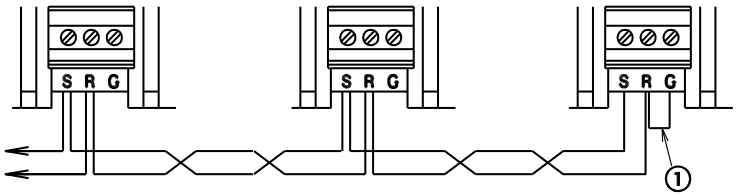
2. Insert the wire into the COM port until it contacts the back of the socket



3. Turn the screw clockwise to fix the wire in place



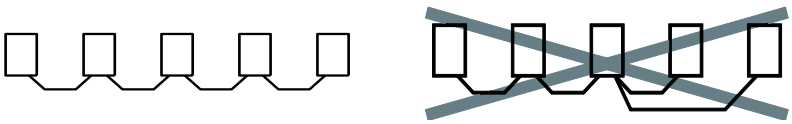
RS485 connection diagram



①	Bridge the E terminal and the free (-) terminal on the first and on the last station of the transmission line to terminate the data bus.

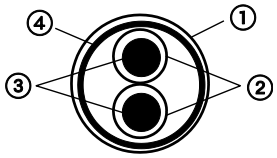
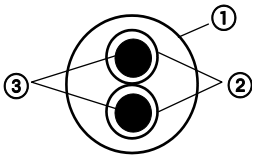
Note

Wiring should extend from one station to the next. Never run two wires from a single station to two other stations.



5.8.1 Transmission cables

Please use the following transmission cables.

Type	Conductor		Insulator		Cable diameter [mm]
	Size [mm^2]	Resistance (at 20°C) [Ω/km]	Material	Thickness [mm]	
Shielded twisted pair 	≥0.5 (AWG20)	≤33.4	Polyethylene	≤0.5	≈7.8
VCTF 	≥0.5 (AWG20)	≤37.8	Polychlorinated biphenyl	≤0.6	≈6.2

①	Cover
②	Insulator
③	Conductor
④	Shield

Note

- Only use shielded twisted pair cables.
- Only use one type of transmission cable. Do not mix more than one type.
- Ground one end of the shielded twisted pair cable.
- If two wires are connected to the plus and minus terminals of the RS485 port, use wires of the same cross-sectional area ($0.5mm^2$).

Chapter 6

Communication

6.1 Communication modes

The FP0R offers four different communication modes:

- MEWTOCOL-COM Master/Slave
- Program controlled
- PLC Link (MEWNET-W0)
- Modbus RTU Master/Slave

Communication ports

The FP0R is equipped with the following ports:

- TOOL port (RS232C interface)
- USB port (USB 2.0 Full Speed interface)
- COM port (RS232C or RS485 interface)

6.1.1 Terminology in FPWIN Pro and FPWIN GR

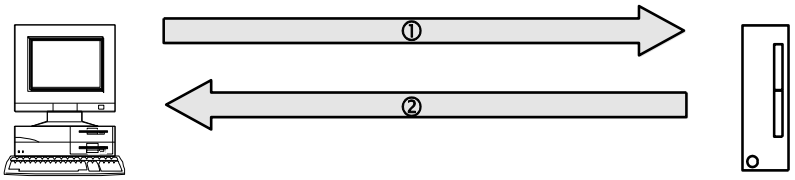
Although similar, FPWIN Pro and FPWIN GR use slightly different terminology to describe communication modes. The following table provides the terminological equivalents for FPWIN GR.

FPWIN Pro	FPWIN GR
MEWTOCOL-COM Master/Slave	Computer link
Program controlled	General-purpose serial communication
Modbus RTU Master/Slave	MODBUS RTU
PLC Link (MEWNET-W0)	PC (PLC) link

6.1.2 MEWTOCOL-COM master/slave

This communication mode uses the proprietary MEWTOCOL-COM protocol to exchange data between a master and one or more slaves. This is called 1:1 or 1:N communication. A 1:N network is also known as a C-NET.

MEWTOCOL-COM connection between a computer and the FP0R:



MEWTOCOL-COM connection between a computer and the FP0R

① Command message ② Response message

There is a master function and a slave function. The side that issues commands is called master. The slave receives the commands, executes the process and sends back responses. The slave answers automatically to the commands received from the master, so no program is necessary on the slave.

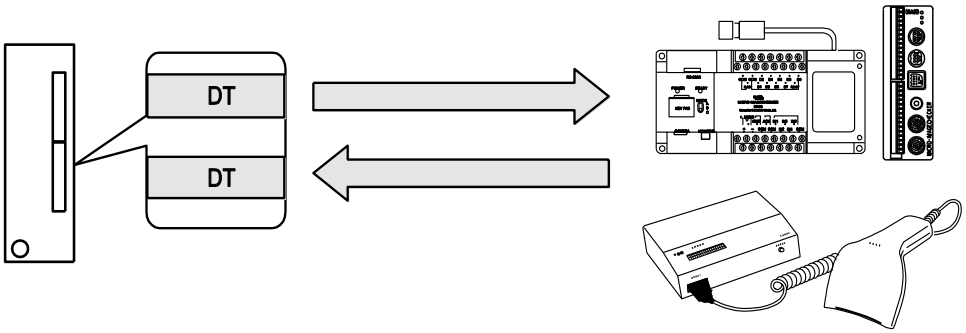
Reference

For detailed information on the MEWTOCOL-COM communication mode, see "MEWTOCOL-COM" on p. 105.

6.1.3 Program controlled communication

With program controlled communication, the user generates a program which governs the data transfer between a PLC and one or more external devices connected to the communication port, e.g. an image processing device or a bar code reader. Hence any desired protocol can be programmed to adapt to external devices.

Typically, such a user program consists of sending and receiving the data. The data to be sent and the data received are stored in data register areas (DT) defined as send and receive buffers.



Reference

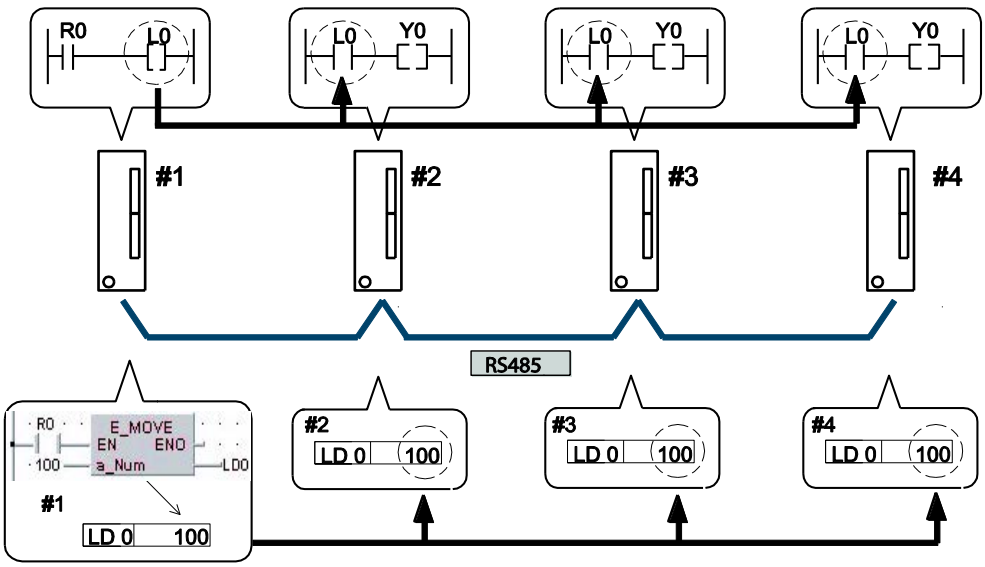
For detailed information on the program controlled communication mode, see "Program controlled communication" on p. 119.

6.1.4 PLC Link

PLC Link is an economic way of linking PLCs using a twisted-pair cable and the MEWNET protocol. Data is shared with all PLCs by means of dedicated internal flags called link flags (L) and data registers called link registers (LD). The statuses of the link flags and link registers of one PLC are automatically fed back to the other PLCs on the same network. The link flags and link registers of the PLCs contain areas for sending and areas for receiving data. Station numbers and link areas are allocated using the system registers.

Example

Link flag L0 for station #1 turns to TRUE. The status change is fed back to the programs of the other stations, and Y0 of the other stations is set to TRUE. A constant of 100 is written to link register LD0 of station #1. The contents of LD0 in the other stations are also changed to a constant of 100.



PLC Link connection between four FP0R units

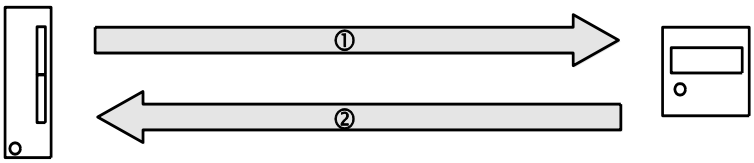
#	Station number of PLC	LD	Link register

Reference

For detailed information on the PLC Link communication mode, see "PLC Link" on p. 141.

6.1.5 Modbus RTU master/slave

This communication mode uses the Modbus RTU protocol to exchange data between a master and one or more slaves. This is called 1:1 or 1:N communication.



Modbus RTU connection between the FP0R and an external device

① Command message ② Response message

There is a master function and a slave function. The side that issues commands is called master. The slave receives the commands, executes the process and sends back responses. The slave answers automatically to the commands received from the master, so no program is necessary on the slave.

The Modbus protocol supports both ASCII mode and RTU binary mode. However, the PLCs of the FP Series only support the RTU binary mode.

Reference

For detailed information on the Modbus RTU communication mode, see "Modbus RTU communication" on p. 160.

6.2 Ports: names and principle applications

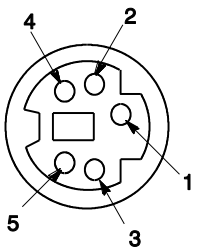
Port name	Connector	Communication mode
TOOL port	Mini DIN 5-pin connector	• MEWTOCOL-COM Slave • Program controlled (in RUN mode only)[1)]
USB port	USB miniB type	• MEWTOCOL-COM Slave
COM port	3-wire RS232C or 2-wire RS485 (screw type)	• MEWTOCOL-COM Master/Slave • Program controlled • Modbus RTU Master/Slave • PLC Link

1) In PROG mode, the TOOL port is automatically set to MEWTOCOL-COM mode even if program controlled mode has been selected. This way it is always possible to communicate in PROG mode with a programming software like Control FPWIN Pro.

6.2.1 TOOL port

The TOOL port can be used to connect a programming tool.

A commercial 5-pin mini DIN connector is used for the TOOL port on the CPU.



Pin no.	Signal name	Abbreviation	Signal direction
1	Signal Ground	SG	–
2	Send Data	SD	CPU → External device
3	Receive Data	RD	CPU ← External device
4	(Not used)	–	–
5	+5V	+5V	CPU → External device

The factory settings are shown below. They can be changed in the system registers.

Communication parameter	Factory settings
Baud rate	9600bit/s
Data length	8
Parity	Odd
Stop bit	1bit

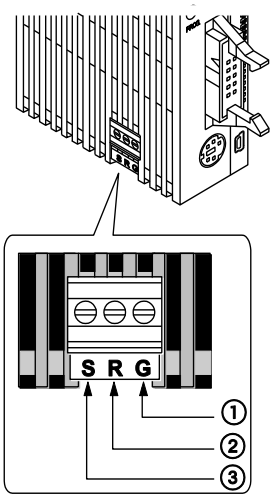
Set the station number for the TOOL port in the TOOL port setting area of the system registers.

6.2.2 COM port

This port is used to connect devices via RS232C or RS485 to enable data input/output.

CPU types with a COM port for RS232C communication: C10CR, C14CR, C16C, C32C, T32C, F32C

CPU types with a COM port for RS485 communication: C10MR, C14MR, C16M, C32M, T32M, F32M

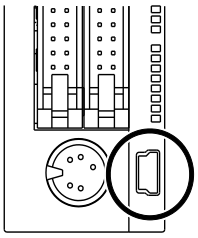


	Symbol	RS232C	RS485
①	G	Signal Ground	E terminal
②	R	Receive Data (Input)	Transmission line (-)
③	S	Send Data (Output)	Transmission line (+)

6.2.3 USB port

The USB port can be used to connect a programming tool.

Panasonic's USB cable CABMINIUSB5D or a commercial USB2.0 AB type cable can be used.



To use the USB port, you must install the USB driver.

Specifications

Item	Description
Connector	5-pin Mini-B type
Standard (baud rate)	USB 2.0 Full Speed
Communication mode	MEWTOCOL-COM Slave

NOTICE

Install the programming tool before connecting the FP0R with a PC.
If you connect the FP0R to a PC with the USB cable before the programming tool is installed or during installation, the USB driver will not be installed correctly.

USB port settings

The settings for the USB port are fixed and cannot be changed.

Connecting the PLCs with a personal computer using a USB cable enables communication with our programming software.

This communication method uses the USB as a virtual serial port, i.e. the FP0R connected via USB is treated by the PC as if connected via the COM port. The COM port number of the COM port allocated for the USB is fixed unless you change the number.

You need only perform the connection procedure the first time you establish the USB connection.

However, you must change the communication setting when switching between the USB and TOOL port connection.

System requirements

- Operating system on the PC:
  - Windows®2000
  - Windows®XP
  - Windows®Vista
  - Windows®7
- Control FPWIN Pro version 6.1 or later, or FPWIN GR version 2.80 or later
- USB cable (see p. 23)

Note

- A USB hub cannot be used.
- When multiple FP0R units are connected to one PC with the USB, they cannot communicate with the PC simultaneously. The PC can communicate with the FP0R that was connected first only, and it cannot communicate with the other FP0R.

6.2.3.1 Installing the USB driver

The following two USB drivers must be installed to recognize the USB:

- USB driver
- USB-COM conversion driver

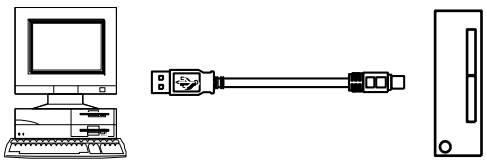
The installation procedure may differ depending on the PC's OS.

Note

For a PC with more than one connector, you may be requested to reinstall these two drivers if the USB connectors' positions have changed.

Procedure

1. Turn on the power supply of the FP0R
2. Connect the FP0R with a PC using a USB cable



3. The PC recognizes the USB driver automatically.
4. Follow the wizard's instructions

Confirming COM Ports

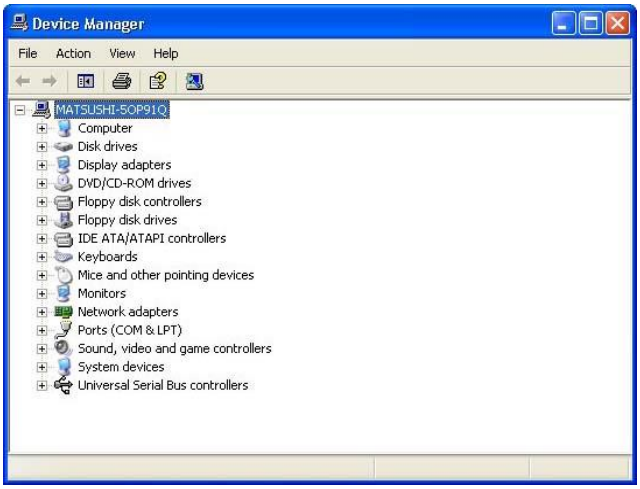
The FP0R connected to the PC via USB is treated as if connected via a COM port. It depends on your PC environment to which COM port the USB is allocated. Therefore, it is necessary to confirm the COM port number.

A COM port number is necessary for communication with the programming tool.

Procedure

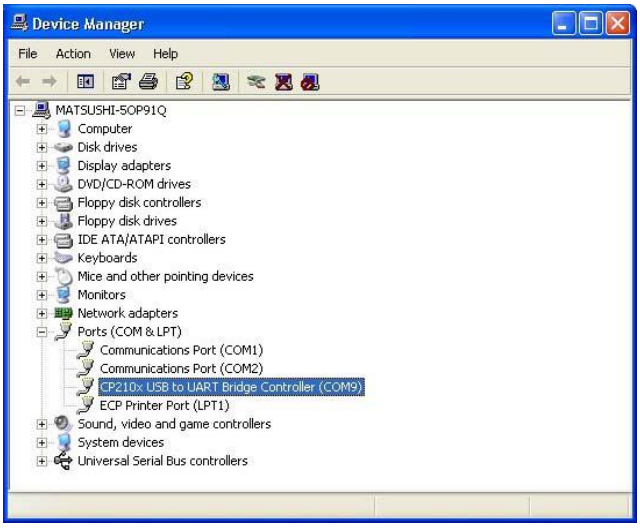
1. Display Device Manager

   For **Windows®7**: Control Panel → Device Manager.
   For **Windows®XP**: My computer → View system information → Hardware tab → Device Manager.



2. Double-click "Ports (COM & LPT)"
3. Confirm the COM port no.

   "CP210x USB to UART Bridge Controller (COM n)" is the COM port allocated. COM9 is allocated in the following display.



Note

If "? CP210x USB to UART Bridge Controller" appears in "Other devices" or "Unknown device" is indicated, the installation has failed. Reinstall the USB driver (see p. 98).

6.2.3.2 Communication with the programming tool

In Control FPWIN Pro, follow these steps:

Procedure

1. Online → **Communication parameters**
2. Make the following settings in the "Communication setting" dialog:

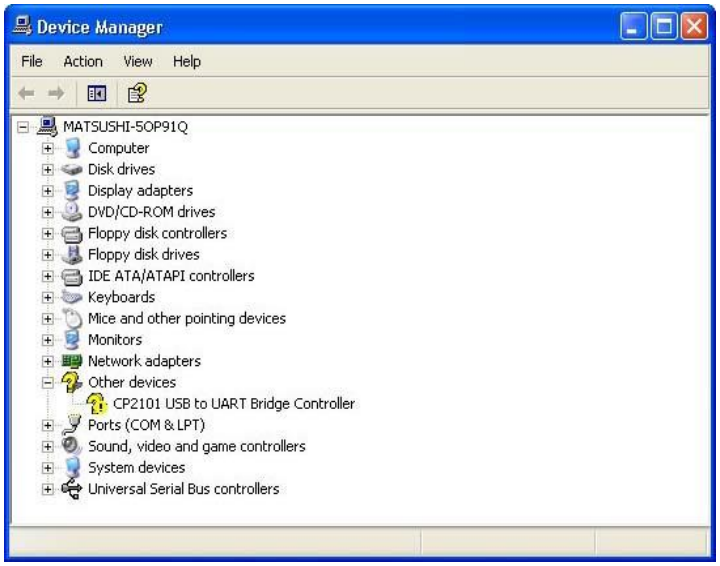
Parameter	Setting
Network type	C-NET (RS232C, USB)
COM port	COM port number allocated for the USB
Baud rate	115200bit/s (Communicates with 115200bit/s when the USB is connected)
Data length	8 bits
Stop bit	1 bit
Parity	Odd

Reference

Please refer to your programming tool's online help for information on the COM port settings.

6.2.3.3 Reinstalling the USB driver

The USB driver must be installed again if the installation failed. If "? CP210x USB to UART Bridge Controller" appears in "Other devices" or "Unknown device" is indicated, the installation has failed.



Also, reinstall the driver if the USB connection does not work well.

Reinstalling the USB driver

Procedure

1. Right-click "? CP210X USB to UART Bridge Controller"
2. Select "Delete"
3. Reinstall the USB driver (see p. 95)

6.3 Communication specifications

TOOL port

Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX

Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Slave Modem connection Program controlled (in RUN mode only)

USB port

Item	Description
Standard (baud rate)	USB 2.0 Full Speed
Communication mode	MEWTOCOL-COM Slave

COM port (RS232C)

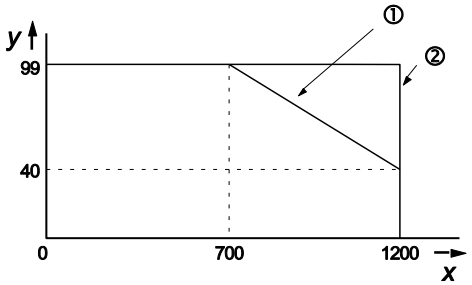
Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Master/Slave Modem connection Program controlled Modbus RTU Master/Slave PLC Link

COM port (RS485)

Item		Description
Interface		RS485
Connection mode		1:N
Transmission distance		1200m [1)2)]
Baud rate		19200, 115200bit/s [2)3)]
Communication method		2-wire, half-duplex
Synchronous method		Start stop synchronous system
Transmission line		Shielded twisted-pair cable or VCTF
Transmission code	MEWTOCOL-COM	ASCII
	Program controlled	ASCII, Binary
	Modbus RTU	Binary
Communication format (set in system registers) [4)]		Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
No. of connected stations [2) 5)]		≤99 (≤32 with C-NET adapter)
Communication mode		MEWTOCOL-COM Master/Slave Modem connection Program controlled Modbus RTU Master/Slave PLC Link

[1)] The number of stations, transmission distance, and baud rate may vary depending on the connected RS485 device.

2) The values for the transmission distance, baud rate and number of stations should be within the values noted in the following graph.



x Transmission distance [m]

y Number of stations

① For a baud rate of 115200bit/s

② For a baud rate of 19200bit/s

3) Set the baud rate in the system registers and set the DIP switch on the bottom of the unit to the same setting. When a C-NET adapter is connected to the RS485 interface, you can only specify a baud rate of 19200bit/s.

4) The start and end code can only be used in program controlled communication.

5) Station numbers should be registered via the system registers.

Note

If the potential difference between the power supplies of RS485 devices exceeds 4V, communication may fail because the RS485 port is non-isolated. The large potential difference will damage the connected devices.

Default settings

Port	Baud rate	Data length	Parity	Stop bit
TOOL port	9600bit/s	8 bits	Odd	1 bit
COM port (RS232C)	9600bit/s	8 bits	Odd	1 bit
COM port (RS485)	115200bit/s	8 bits	Odd	1 bit

6.4 Communication parameters

The communication parameters are set in the system registers of the PLC. Make settings for the communication mode, communication format, baud rate, station number, and receive buffer if necessary.

During PROG mode:

Use the programming tool to enter settings for the communication port in the system registers.

During RUN mode:

Use the SYS1 instruction to change the communication parameters. Please refer to the Control FPWIN Pro online help for detailed information.

The communication mode can be switched using F159_MRTN (see p. 105).

6.4.1 Setting system registers in PROG mode

Procedure

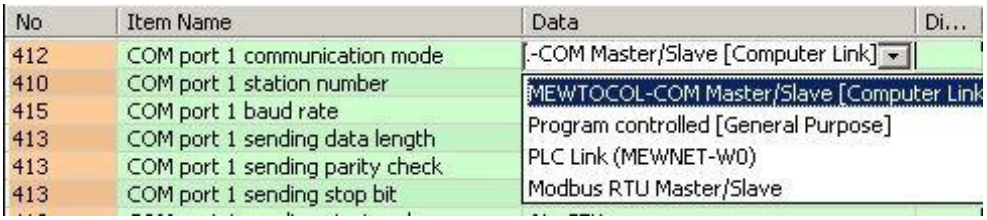
1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click "COM port"

   To make settings for the TOOL port, select "TOOL port" under "System registers".

The following communication parameters are set in the system registers:

Communication mode

Select a communication mode. The factory setting for the communication mode is "MEWTOCOL-COM Master/Slave".



Station number

The station number must be set for MEWTOCOL-COM Master/Slave, Modbus RTU, and for PLC Link.

MEWTOCOL-COM/ Modbus RTU:	The station number can be set within a range of 1 to 99. In the FP0 compatibility mode, the station number can be set within a range of 1 to 32.
PLC Link:	The station number can be set within a range of 1 to 16.

By default, the station number for each communication port is set to 1 in the system register settings. There is no need to change this for 1:1 communication, but if 1:N communication is used to connect multiple PLCs to the transmission line, the station number must be specified to identify the different PLCs.

The station number is specified either by using

- the SYS1 instruction
- system register settings in the programming tool

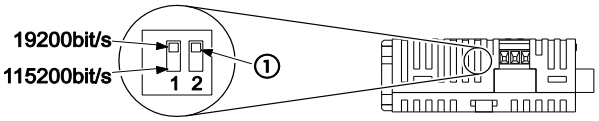
The priority for setting the station number is in the above order.

Reference

For details, please refer to the description of the SYS1 instruction in your Programming Manual or in the FPWIN Pro online help.

Baud rate

- The default baud rate for most ports is 9600bit/s. Select a value from 2400 to 115200bit/s.
- Lower baud rates of 300, 600, and 1200bit/s can be specified using the SYS1 instruction. However, this will not change the setting value of the system register.
- The setting must match the external device connected to the communication port.
- When using the RS485 port, a baud rate of 19200bit/s or 115200 bit/s is possible. Set the baud rate in the system registers and set the DIP switch on the bottom of the unit to the same setting. Confirm the baud rate setting before installation. The factory setting is 115200bit/s.



RS485 baud rate switch

①	Unused

- PLC Link: The baud rate is fixed at 115200bit/s.
- FP0 compatibility mode:

TOOL port:	9600 or 19200bit/s
COM port:	300, 600, 1200, 2400, 4800, 9600, or 19200bit/s

Communication format setting

Default settings:

Data length:	8 bits
Parity:	Odd
Stop bit:	1 bit
Start code:	No STX
End code:	CR, use SendCharactersAndClearString for end code suppression
Baud rate	115200bit/s

The setting must match the external device connected to the communication port.

MEWTOCOL-COM/ Modbus RTU:	The end code setting must always be "CR", and the start code setting must be "No STX".
PLC Link:	The communication format settings are fixed.

Receive buffer

For program controlled communication, a receive buffer must be specified in the system registers. Set a value for receive buffer starting address and receive buffer capacity. See "Setting communication parameters" on p. 142.

6.4.2 Changing communication mode in RUN mode

The communication mode of the CPU's communication ports can be changed during RUN mode. You can toggle between program controlled mode and MEWTOCOL-COM mode by executing F159_MTRN and setting the variable **n_Number** (the number of bytes to be sent) to 16#8000. Or use the SetCommunicationMode instruction to set the communication mode.

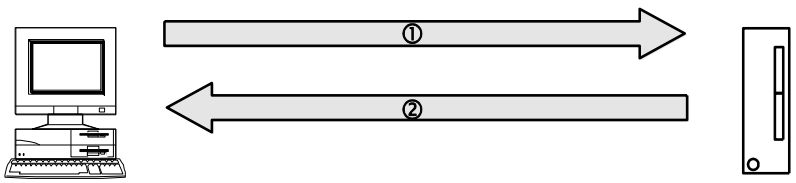
For programming examples, please refer to F159_MTRN and SetCommunicationMode in the FPWIN Pro online help.

Note

- When the power is turned on, the communication mode selected in the system registers is set.
- It is not possible to change to Modbus RTU mode or PLC Link mode during RUN mode.

6.5 MEWTOCOL-COM

This communication mode uses the proprietary MEWTOCOL-COM protocol to exchange data between a master and one or more slaves. This is called 1:1 or 1:N communication. A 1:N network is also known as a C-NET.



MEWTOCOL-COM connection between a computer and the FP0R

① Command message ② Response message

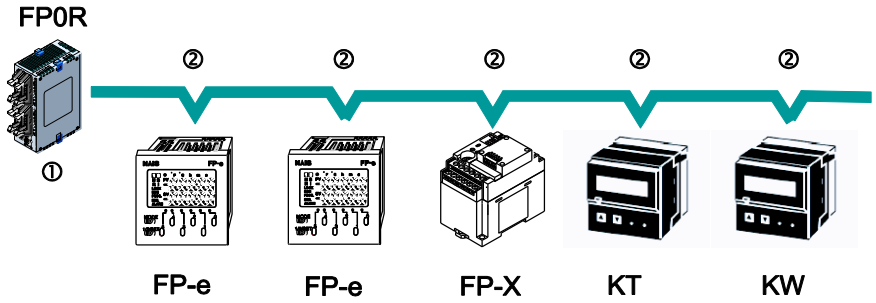
There is a master function and a slave function. The side that issues commands is called master. The slave receives the commands, executes the process and sends back responses. The slave answers automatically to the commands received from the master, so no program is necessary on the slave.

MEWTOCOL-COM master function

The master can be a PLC or any external device supporting the master function. To use the built-in master functionality of the PLC, select MEWTOCOL-COM Master/Slave in the system registers and implement a PLC program. The applicable instructions are F145_WRITE_DATA and F146_READ_DATA.

MEWTOCOL-COM Master/Slave mode is recommended over program controlled mode since programming is easier.

The master function can be used for communication with all Panasonic devices equipped with a MEWTOCOL-COM slave function, for example PLCs, Imagecheckers, temperature controllers, or eco-power meters.



① Master ② Slave

Note

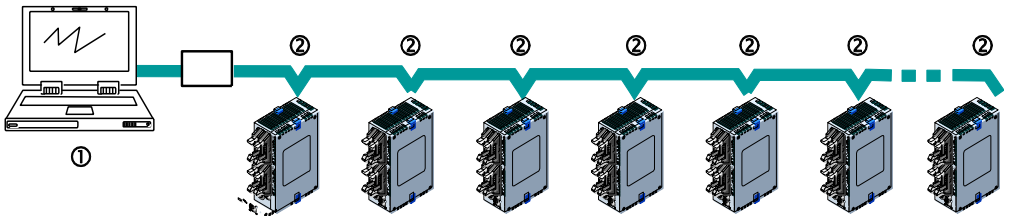
The master function is only available via the COM port.

Do not execute the F145_WRITE_DATA and F146_READ_DATA instructions when the unit is used as a slave unit.

MEWTOCOL-COM slave function

The slave can be a PLC or any external device which supports the MEWTOCOL-COM protocol. The slave automatically receives a command, processes it and sends back a response. To use the built-in slave functionality of the PLC, select "MEWTOCOL-COM Master/Slave" in the system registers. For 1:N communication in a C-NET, the station number must be specified in the system registers of the slave. No program is necessary on the slaves.

The program for the master side must send and receive commands according to the MEWTOCOL-COM protocol. MEWTOCOL-COM contains the commands used to control and monitor the slave operation.



① Master ② Slave

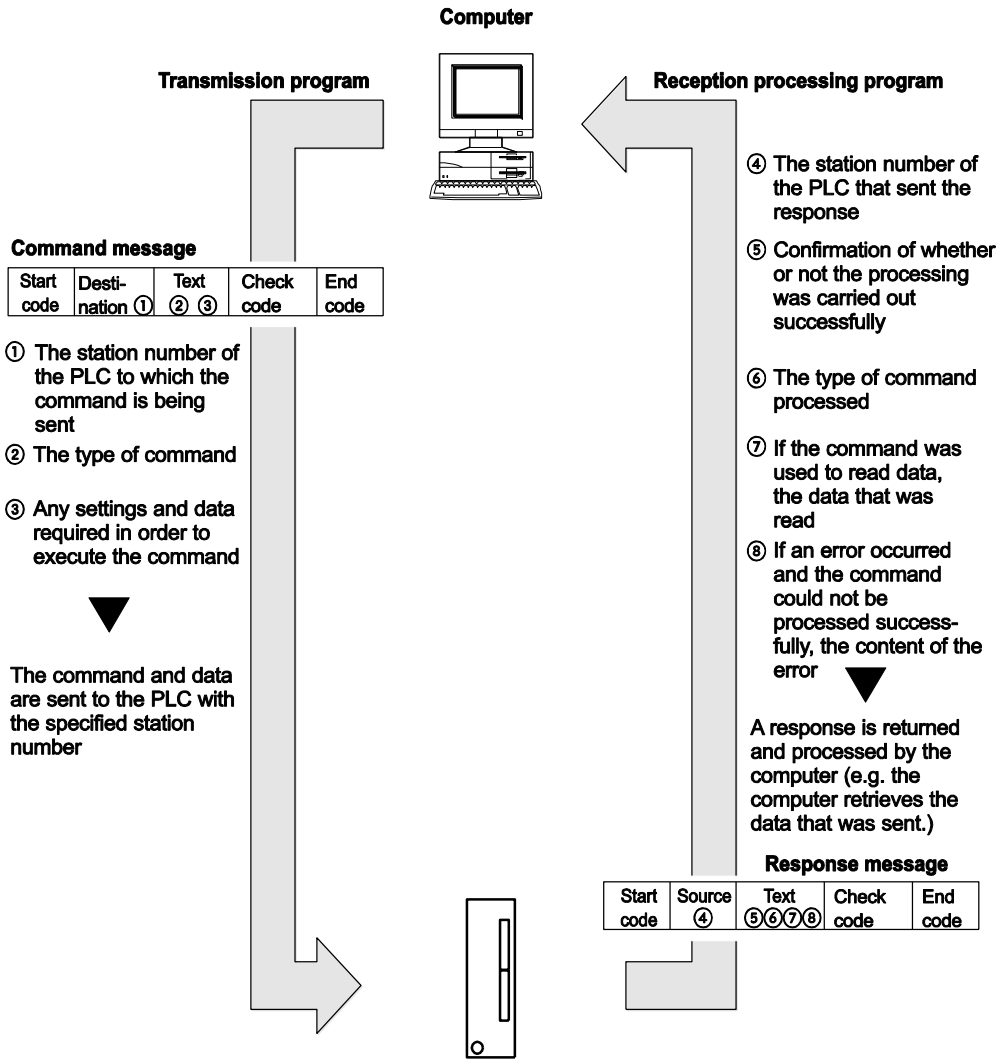
Note

Panasonic offers software tools with implemented MEWTOCOL-COM Master functionality:

- Control FP Connect – connects your Visual Basic application to Panasonic PLCs
- PCWAY – displays PLC data in Excel

6.5.1 Operation outline for MEWTOCOL-COM slave

Instructions issued by the computer to the PLC are called commands. Messages sent back to the computer from the PLC are called responses. When the PLC receives a command, it processes the command regardless of the sequence program, and sends a response back to the computer. Communication is carried out in a conversational format, based on the MEWTOCOL-COM communication procedures. Data is sent in ASCII format. The computer has the first right of transmission. The right of transmission shifts back and forth between the computer and the PLC each time a message is sent.

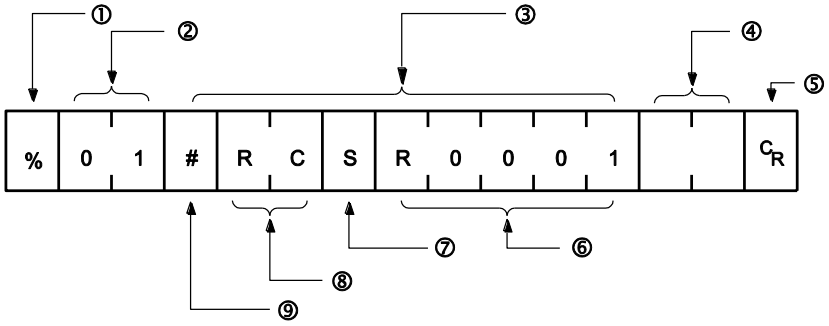


MEWTOCOL-COM communication between the FP0R and a computer

6.5.2 Command and response format

Command message

All command-related items should be noted in the text segment. The station number must be specified before sending the command.



①	**Start code**
	Commands must always have a "%" (ASCII code: 16#25) or a "<" (ASCII code: 16#3C) at the beginning of a message. The FP0R supports an expansion start code ("<") to send single frames of up to 2048 characters. Using the start code "%", a maximum of 118 characters can be sent in one frame.
②	**Station number**
	The station number of the slave to which you want to send the command must be specified. The range is 01 to 99 (decimal). In 1:1 communication, the station number "01" (ASCII code: 16#3031) should be specified.
③	**Text**
	The content differs depending on the command. The content should be noted in all upper-case characters, following the fixed formula for the particular command. The method for writing text segments in the message varies depending on the type of command.
④	**Check code**
	Hexadecimal block check code (BCC) for error detection using horizontal parity. The BCC should be created so that it targets all of the text data from the header to the last text character. The BCC starts from the header and checks each character in sequence, using the exclusive OR operation, and replaces the final result with character text. It is normally part of the calculation program and is created automatically. The parity check can be skipped by entering "* *" (ASCII code: 16#2A2A) instead of the BCC.
⑤	**End code**
	Messages must always end with a "$_{R}^{C}$" (ASCII code: 16#0D).
⑥	**Target address**
	Address of the target area to be read or written (e.g. internal flag R1)
⑦	**Data area**
	Specification of the number of points to be read or written (S = 1 point)

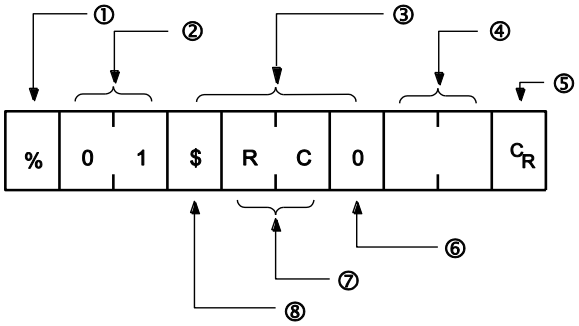
⑧	**Command name**
	e.g. RC, read contact area
⑨	**Command code**
	# (16#23) indicates that this is a command

Note

If there are large numbers of characters to be written, they may be divided and sent as several commands. If there is a large number of characters in the value that was loaded, they may be divided and several responses sent.

Response message

The slave that received the command in the example above sends the processing results to the computer.

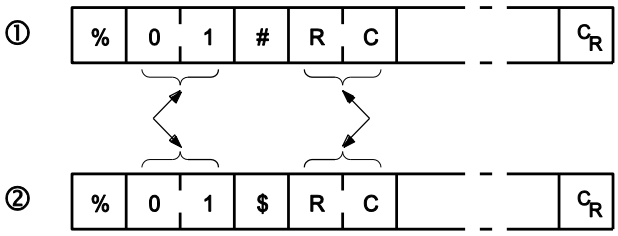


①	**Start code**
	A "%" (ASCII code: 16#25) or "<" (ASCII code: 16#3C) must be at the beginning of a message. The response must start with the same start code as the command.
②	**Station number**
	The station number of the slave that processed the command is stored here.
③	**Text**
	The content of this varies depending on the type of command. The value should be read based on the content. If the processing is not completed successfully, an error code will be stored here, so that the content of the error can be checked.
④	**Check code**
	Hexadecimal block check code (BCC) for error detection using horizontal parity. The BCC starts from the header and checks each character in sequence, using the exclusive OR operation, and replaces the final result with character text.
⑤	**End code**
	There is always a "$_{CR}$" (ASCII code: 16#0D) at the end of the message.
⑥	**Data**
	For a read command, the data read is stored here.

⑦	**Command name/error code**
	Normal processing: The command name is stored here. Error condition: The error code is stored here.
⑧	**Response code**
	Normal processing: "$" (ASCII code: 16#24) Error condition: ! (ASCII code: 16#21) If the response contains an "!" instead of a "$", check the meaning of the error code.

Note

- If no response is returned, the command may not have arrived at the slave, or the slave may not be functioning. Check to make sure all of the communication specifications (e.g. baud rate, data length, and parity) match between the master and the slave.
- Station number and command name are always identical in a command and its corresponding response (see below). This makes the correspondence between a command and a response clear.



① Command message ② Response message

6.5.3 Commands

Command name	Code	Description
Read contact area	RC (RCS) (RCP) (RCC)	Reads the on and off status of contacts. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Write contact area	WC (WCS) (WCP) (WCC)	Turns contacts on and off. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Read data area	RD	Reads the contents of a data area.
Write data area	WD	Writes data to a data area.
Read timer/counter set value area	RS	Reads the value set for a timer/counter.
Write timer/counter set value area	WS	Writes a timer/counter setting value.
Read timer/counter elapsed value area	RK	Reads the timer/counter elapsed value.
Write timer/counter elapsed value area	WK	Writes the timer/counter elapsed value.

Command name	Code	Description
Register or Reset contacts monitored	MC	Registers the contact to be monitored.
Register or Reset data monitored	MD	Registers the data to be monitored.
Monitoring start	MG	Monitors a registered contact or data using MD and MC.
Preset contact area (fill command)	SC	Embeds the area of a specified range in a 16-point on and off pattern.
Preset data area (fill command)	SD	Writes the same contents to the data area of a specified range.
Read system register	RR	Reads the contents of a system register.
Write system register	WR	Specifies the contents of a system register.
Read the status of PLC	RT	Reads the specifications of the PLC and error codes if an error occurs.
Remote control	RM	Switches the operation mode of the PLC.
Abort	AB	Aborts communication.

6.5.4 Setting communication parameters

Make the following settings for the communication port:

- communication mode
- station number
- baud rate
- communication format

For details on setting the communication parameters, see "Setting system registers in PROG mode" on p. 102.

Note

- The end code setting must always be "CR", and the start code setting must be "No STX".
- The station number can be set within a range of 1 to 99.
- With a C-NET adapter, a maximum of 32 stations can be specified.
- The master function is only available via the COM port.

6.5.4.1 FP0 compatibility mode

Make sure that the PLC type selected in FPWIN Pro is "FP0".

All ports can be used in FP0 compatibility mode. For the USB port, the settings are fixed.

Make the following settings for the communication port:

TOOL port

- station number
- modem connection (disable/enable)
- communication format (sending data length)
- baud rate

COM port

- communication mode
- station number
- baud rate
- communication format
- modem connection (disable/enable)

For details on setting the communication parameters, see p. 102.

Note

The end code setting must always be "CR", and the start code setting must be "No STX".

6.5.5 1:1 slave communication

System register settings

For 1:1 MEWTOCOL-COM communication, the system registers should be set as shown below.

Name	Name	Set value
410	COM port 1 - station number	1
412	COM port 1 - communication mode	MEWTOCOL-COM Master/Slave
413	COM port 1 - communication format	Data length: 8 bits Parity: Odd Stop bit: 1 bit End code: CR Start code: No STX
415	COM port 1 - baud rate	2400–115200bit/s

Note

The communication format and baud rate of the PLC should be set to match the connected device.

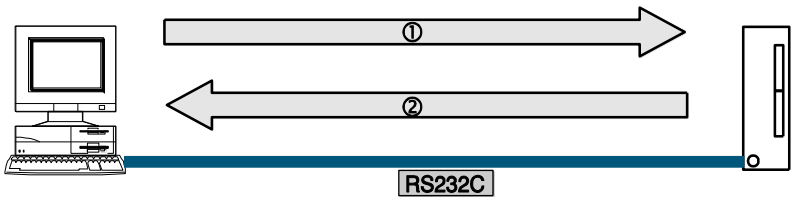
Programming

For MEWTOCOL-COM communication, a program must be created that allows command messages to be sent and response messages to be received on the computer side. There is no programming required on the slave. Only the station number and the communication parameters must be set in the system registers. The program for the master side must send and receive commands according to the MEWTOCOL-COM protocol. MEWTOCOL-COM contains the commands used to control and monitor the slave operation.

If a software program such as PCWAY is used on the computer side, PLC data can easily be read and written without having to think about the MEWTOCOL-COM protocol.

6.5.5.1 1:1 communication with a computer

For a 1:1 MEWTOCOL-COM connection between the FP0R and a computer, an RS232C cable is needed. Communication is performed via commands from the computer and responses from the PLC.

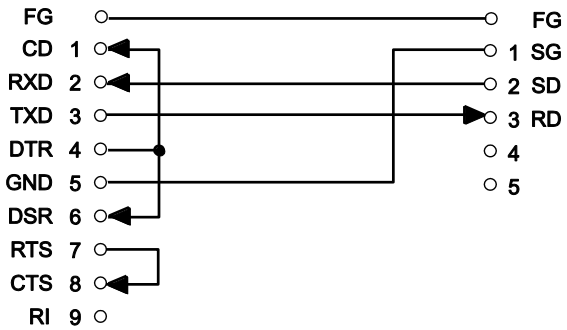


① Command message ② Response message

1:1 MEWTOCOL-COM connection between a computer and the FP0R

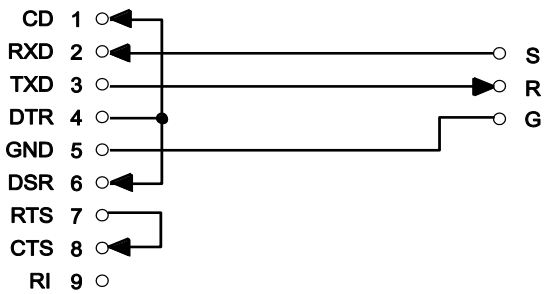
It is recommended to connect the computer to the TOOL port of the FP0R. A connection cable (order no. AFC8513D) with a 5-pin mini-DIN connector and a 9-pin Sub-D connector is available. A communication cable with a 9-pin sub-D connector on one end and open wires on the other end (AIGNCAB232D5) is available for connection to the COM port.

- Using the TOOL port



Left: computer, right: FP0R

- Using the COM port (RS232C)



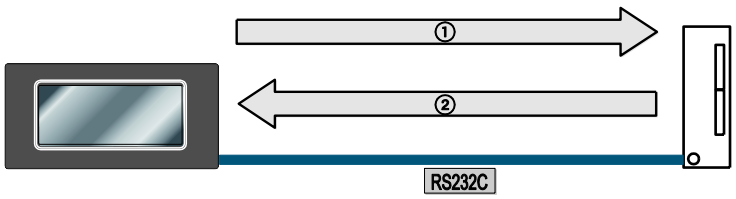
Left: computer, right: FP0R

6.5.5.2 1:1 communication with GT touch panels

For a 1:1 MEWTOCOL-COM connection between the FP0R and a touch panel of the GT series, an RS232C cable is needed. Communication is performed via commands from the touch panel and responses from the PLC.

No program is required for communication. Simply set the mutual communication settings to operate the PLC via the touch panel.

It is recommended to connect the computer to the TOOL port of the FP0R. A connection cable (order no. AFC8513D) with a 5-pin mini-DIN connector and a 9-pin Sub-D connector is available.



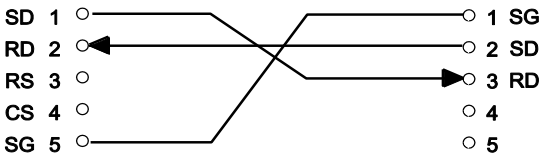
MEWTOCOL-COM connection between a touch panel of the GT series and the FP0R

①	Command message	②	Response message

Note

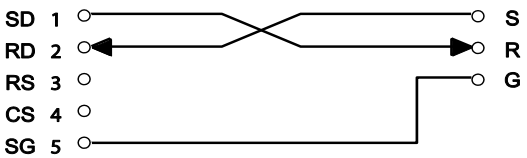
A USB cable cannot be used.

- Using the TOOL port



Left: GT terminal, right: FP0R

- Using the COM port (RS232C)



Left: GT terminal, right: FP0R

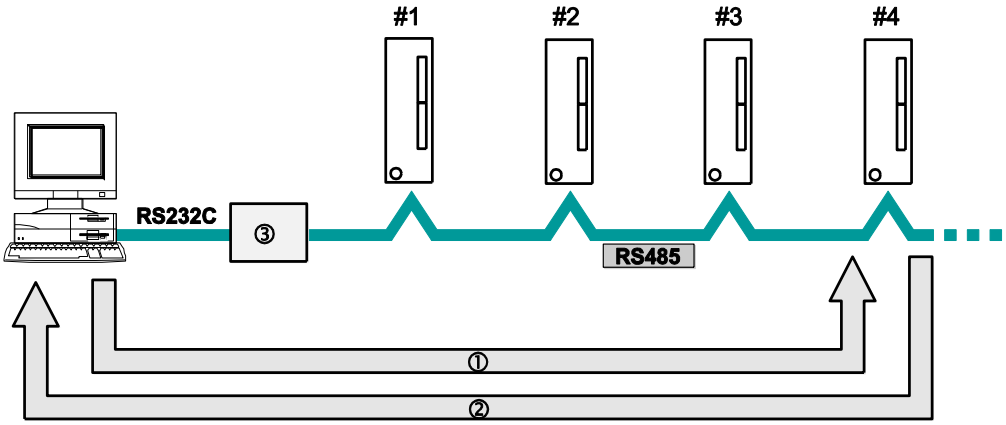
Reference

Please refer to the "GT Series Technical Manual" for more information.

6.5.6 1:N slave communication

For a 1:N MEWTOCOL-COM connection between a computer and several PLCs, the computer and the first PLC are connected through a commercially available RS232C-RS485 converter. The other PLCs are connected using twisted pair cables.

The computer and the PLCs communicate via commands and responses: The computer sends a command specifying the station number, and the PLC with that station number sends a response back to the computer.



1:N communication between a computer and several PLCs

①	The station number of the PLC to which the command is being sent is included in the command message.
②	The station number of the PLC sending a response is included in the response message.
③	Commercially available converter (also required for PLCs using the RS232C port)
#	Station number of PLC

System register settings

For 1:N MEWTOCOL-COM communication, the system registers for COM port 1 should be set as shown below.

Name	Name	Set value
410	COM port 1 - station number	1 to 99 (with C-NET adapter, a maximum of 32 stations is possible)
412	COM port 1 - communication mode	MEWTOCOL-COM Master/Slave
413	COM port 1 - communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR Start code: No STX
415	COM port 1 - baud rate	2400–115200bit/s

Note

The communication format and baud rate of the PLC should be set to match the connected device.

Lower baud rates of 300, 600, and 1200bit/s can be specified using the SYS1 instruction. However, this will not change the setting value of the system register.

When using the RS485 port, a baud rate of 19200bit/s or 115200 bit/s is possible. Set the baud rate in the system registers and set the DIP switch on the bottom of the unit to the same setting.

Programming

There is no programming required on the slave. Only the station number and the communication parameters must be set in the system registers. The program for the master side must send and receive commands according to the MEWTOCOL-COM protocol. MEWTOCOL-COM contains the commands used to control and monitor the slave operation.

If a software program such as PCWAY is used on the computer side, PLC data can easily be read and written without having to think about the MEWTOCOL-COM protocol.

6.5.7 Sample program for master communication

Use the F145_WRITE and F146_READ instructions for the MEWTOCOL-COM master function. Be sure to set the COM port used in the program to "MEWTOCOL-COM Master/Slave" in the system registers. The master function is only available via the COM port.

GVL

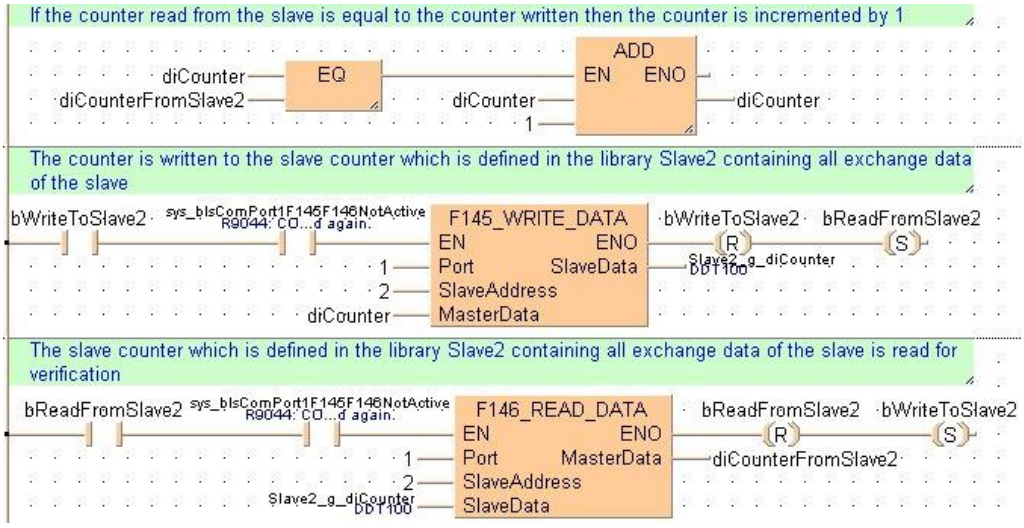
	Class	Identifier	FP Address	IEC Address	Type
0	VAR_GLOBAL	Slave2_g_diCounter	DDT100	%MD5.100	DINT

POU Header

	Class	Identifier	Type	Initial
0	VAR_EXTERNAL	Slave2_g_diCounter	DINT	0
1	VAR	diCounter	DINT	0
2	VAR	diCounterFromSlave2	DINT	-1
3	VAR	bWriteToSlave2	BOOL	TRUE
4	VAR	bReadFromSlave2	BOOL	FALSE

In order to have consistent data in the master project and in the slave project, the common data should be kept in the GVL of a common library.

LD Body



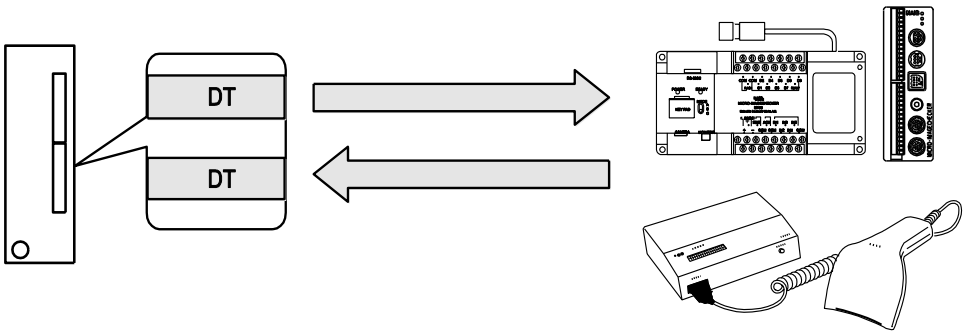
Reference

Please refer to the Control FPWIN Pro online help for detailed information.

6.6 Program controlled communication

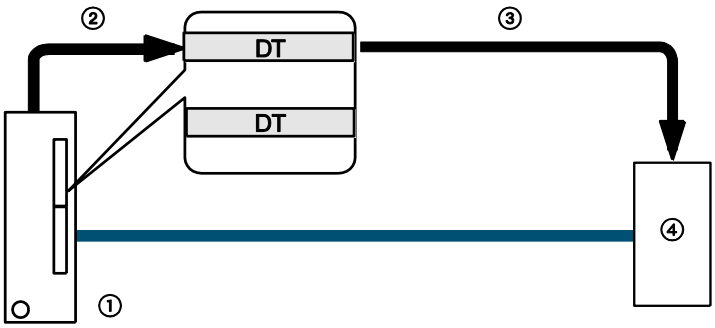
With program controlled communication, the user generates a program which governs the data transfer between a PLC and one or more external devices connected to the communication port, e.g. an image processing device or a bar code reader. Hence any desired protocol can be programmed to adapt to external devices.

Typically, such a user program consists of sending and receiving the data. The data to be sent and the data received are stored in data register areas (DT) defined as send and receive buffers.



Sending data

Sending includes generating the data for the send buffer and sending it using the instructions SendCharacters, SendCharactersAndClearString, or F159_MTRN. SendCharacters and SendCharactersAndClearString implicitly use F159_MTRN. (See also "Sending data" on p. 124.) Sending data can be controlled by the "transmission done" flag. (See also "Flag operation" on p. 133.)



①	PLC
②	Writing data into send buffer
③	Sending data using a send instruction
④	Device with RS232C port

The start and end codes specified in the system registers are automatically added to the data sent. The maximum volume of data that can be sent is 2048 bytes.

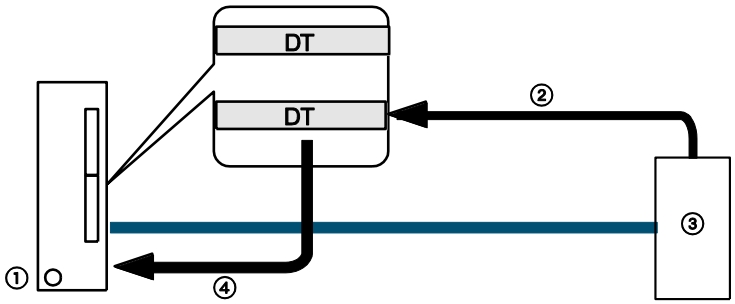
Receiving data

Data is automatically received in the receive buffer (see p. 128). The receive buffer must be defined in the system registers. After the end of re-

ception has been verified, data can be copied into a specified target area of the CPU. Receiving includes processing the data in the receive buffer and preparing the system to receive further data. (See also "Receiving data" on p. 126.)

The end of reception can be verified by:

- evaluating the "reception done" flag or by executing IsReceptionDone
- executing IsReceptionDoneByTimeOut
- directly evaluating the receive buffer. (See also "Flag operation" on p. 133.)



①	PLC
②	Receiving data in receive buffer
③	Device with RS232C port
④	"Reception done" flag turns to TRUE

No end code is included in the data stored. The maximum volume of data that can be received is 4094 bytes.

Note

In the FP0 compatibility mode, F159_MTRN is automatically translated into F144_TRNS.

6.6.1 Setting communication parameters

Make the following settings for the communication port:

- communication mode (Program controlled)
- baud rate
- communication format
- receive buffer

For details on setting the communication parameters, see "Setting system registers in PROG mode" on p. 102.

Note

Program controlled mode is available via COM port and TOOL port.

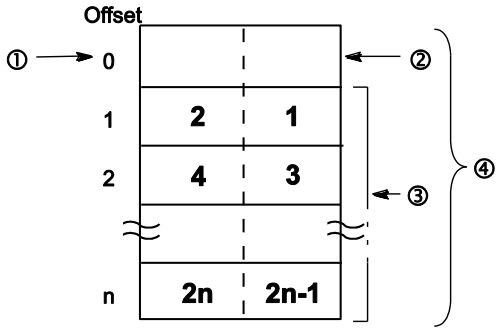
Specifying a receive buffer

For program controlled communication, a receive buffer must be specified in the DT memory area. The maximum area is 2048 words.

Specify the following items:

1. Starting address
2. The capacity of the receive buffer (number of words)

Receive buffer layout

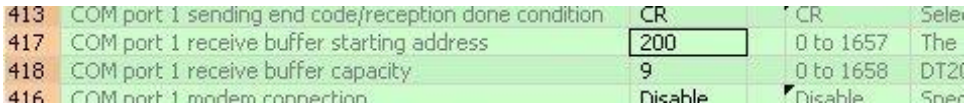


Bold numbers indicate the order of reception.

①	Starting address
②	Storage area for the number of bytes received
③	Storage area for the data received
④	Capacity

Incoming data is stored in the receive buffer. Start and end codes are not stored in the receive buffer. The storage area for the data received starts with the second word of the receive buffer (offset 1). Offset 0 contains the number of bytes received. The initial value of offset 0 is 0.

The receive buffer is specified in the system registers (see p. 102):



Note

FPWIN Pro: In order to use the data in the receive buffer, define a global variable having the same starting address and capacity.

The setting range for the receive buffer starting address is different for the 16k and the 32k type.

6.6.1.1 FP0 compatibility mode

Make sure that the PLC type selected in FPWIN Pro is "FP0".

In the FP0 compatibility mode, only the COM port can be used.

Make the following settings for the communication port:

- communication mode
- station number
- baud rate
- communication format
- receive buffer starting address
- receive buffer capacity

Please be aware that the setting ranges of the FP0 apply if the FP0R is used in FP0 compatibility mode.

For details on setting the communication parameters, see p. 102.

Note

The end code setting must always be "CR", and the start code setting must be "No STX".

6.6.2 Sending data

Sending includes generating the data for the send buffer and sending it using the instructions SendCharacters, SendCharactersAndClearString, or F159_MTRN. SendCharacters and SendCharactersAndClearString implicitly use F159_MTRN. The start and end codes specified in the system registers are automatically added to the data sent. The maximum volume of data that can be sent is 2048 bytes.

Procedure for sending data to external devices:

- **Step 1:** Set communication parameters (see p. 121)

  Required settings: communication mode (program controlled), baud rate, communication format

- **Step 2:** Write to send buffer (see p. 126)

  Not necessary when using SendCharacters or SendCharactersAndClearString.

- **Step 3:** Execute send command

  Use one of the following instructions:

Instruction	Comment
SendCharacters	Easy to use, fits most applications, may require more data memory
SendCharactersAndClearString	Like SendCharacters but works without send buffer, may require less data memory
F159_MTRN	Original F instruction with complete set of parameters, additional transfer instruction required to write data to send buffer

- **Step 4 (optional):** Evaluate "transmission done" flag

  Use one of the following methods:

Method	Comment
IsTransmissionDone	Returns the value of the "transmission done" flag. It turns to TRUE when the specified number of bytes has been sent.
sys_bIsComPort1TransmissionDone sys_bIsComPort2TransmissionDone sys_bIsToolPortTransmissionDone	These system variables turn to TRUE when the specified number of bytes has been sent.

Note

- When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE. Evaluation of the "transmission done" flag may be useful in cases where no response can be expected, e.g. with broadcast messages.
- Data cannot be sent unless the pin CS (Clear to Send) is on. When connecting to a three-wire port, short-circuit the RS and CS pins.

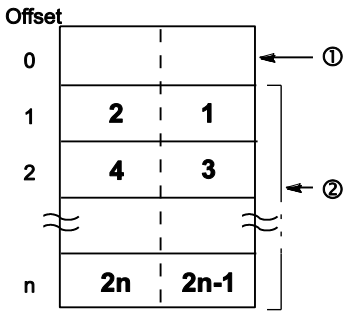
Reference

For details on the operation of the "reception done" flag, the "transmission done" flag, and the communication error flag see p. 133.

Writing to send buffer

The instructions SendCharacters and SendCharactersAndClearString automatically generate the data in the send buffer.

Send buffer layout



①	Storage area for the number of bytes to be sent
②	Storage area for the data to be sent

Bold numbers indicate the order of transmission. The storage area for the data to be sent starts with the second word of the send buffer (offset 1). Offset 0 contains the number of bytes to be sent. The maximum volume of data that can be sent is 2048 bytes.

If F159_MTRN is used for transmission, the data must be copied to the send buffer using a transfer instruction, e.g. F10_BKMV.

6.6.3 Receiving data

Data can be received from an external device if the "reception done" flag is FALSE. (The "reception done" flag turns to FALSE after switching to RUN mode.) Data is automatically received in the receive buffer (see p. 128). The receive buffer must be defined in the system registers. After the end of reception has been verified, data can be copied into a specified target area of the CPU.

When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. The maximum volume of data that can be received is 4094 bytes. No end code is included in the data stored.

Procedure for receiving data from external devices:

- **Step 1:** Set communication parameters (see p. 121) and receive buffer (see p. 128)

  Required settings: communication mode (program controlled), baud rate, communication format, receive buffer

- **Step 2:** Receive data

  Data is automatically received in the receive buffer.

- **Step 3:** Verify end of reception

  Use one of the following methods:

Method	Comment
IsReceptionDone	Returns the value of the "reception done" flag. It is TRUE if the end code has been received.
IsReceptionDoneByTimeOut	Used to verify the end of reception by time-out, e.g. with binary data when no end code is expected.
sys_bIsComPort1ReceptionDone sys_bIsComPort2ReceptionDone sys_bIsToolPortReceptionDone	These system variables turn to TRUE if the end code has been received.
Direct evaluation of the receive buffer.	

- **Step 4:** Process data in receive buffer

  Use one of the following instructions:

Instruction	Comment
ReceiveData	Automatically copies data received by the CPU into the specified variable.
ReceiveCharacters	Automatically copies characters received by the CPU into a string variable.
F10_BKMV	Transfers data from the receive buffer to a target area. Not required with ReceiveData or ReceiveCharacters.

- **Step 5:** Prepare CPU to receive next data

  Use one of the following instructions:

Instruction	Comment
ClearReceiveBuffer	The receive buffer is automatically reset when sending the next data. To reset the receive buffer without sending any data use one of these instructions.
F159_MTRN (n_Number=0)	

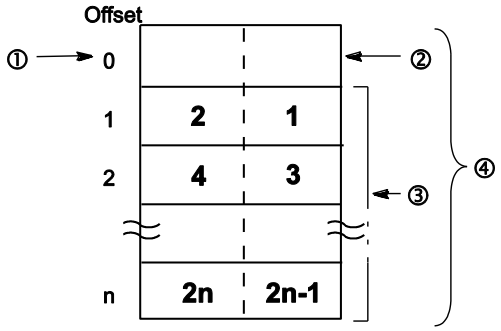
6.6.3.1 Setting receive buffer for CPU

For program controlled communication, a receive buffer must be specified in the DT memory area. The maximum area is 2048 words.

Specify the following items:

1. Starting address
2. The capacity of the receive buffer (number of words)

Receive buffer layout



Bold numbers indicate the order of reception.

①	Starting address
②	Storage area for the number of bytes received
③	Storage area for the data received
④	Capacity

Incoming data is stored in the receive buffer. Start and end codes are not stored in the receive buffer. The storage area for the data received starts with the second word of the receive buffer (offset 1). Offset 0 contains the number of bytes received. The initial value of offset 0 is 0.

Procedure

1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click "COM port"

   The communication ports occupy different bit positions of the same system register, so individual settings for each communication port are possible. To make settings for the TOOL port, select "TOOL port" under "System registers".
   The number of the system register for the respective settings may vary according to the PLC type used.

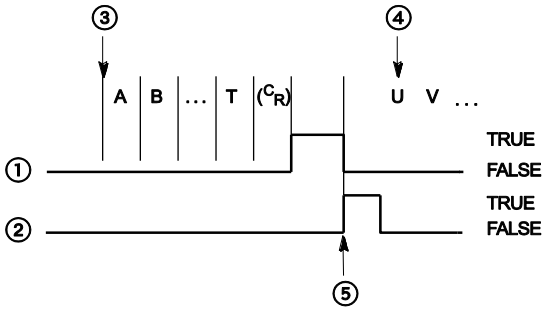
Note

In order to use the data in the receive buffer, define a global variable having the same starting address and capacity.

Processing data in receive buffer and preparing CPU to receive further data

Example

Receive a string of 8 bytes containing the characters "ABCDEFGH" via COM port 1. The characters are stored in ASCII HEX code without start and end codes.



①	"Reception done" flag
②	Execution condition
③	Reception begins
④	Reception continues
⑤	Execution of F159_MTRN (n_Number=0)

Receive buffer layout:

Offset		
0	8	
1	16#42(B)	16#41(A)
2	16#44(D)	16#43(C)
3	16#46(F)	16#45(E)
4	16#48(H)	16#47(G)

When reception begins, the value in offset 0 is 0. At the end of reception, the value in offset 0 is 8. The data in offset 1 to offset 4 is received in order from the low order byte.

System register settings

No	Item Name	Data	Dim...
412	COM port 1 communication mode	Program controlled...	
410	COM port 1 station number	1	
415	COM port 1 baud rate	9600	baud
413	COM port 1 sending data length	8 bits	
413	COM port 1 sending parity check	With-Odd	
413	COM port 1 sending stop bit	1 bit	
413	COM port 1 sending start code	No-STX	
413	COM port 1 sending end code/reception done condition	CR	
416	COM port 1 receive buffer starting address	200	
417	COM port 1 receive buffer capacity	5	
412	COM port 1 modem connection	Disable	

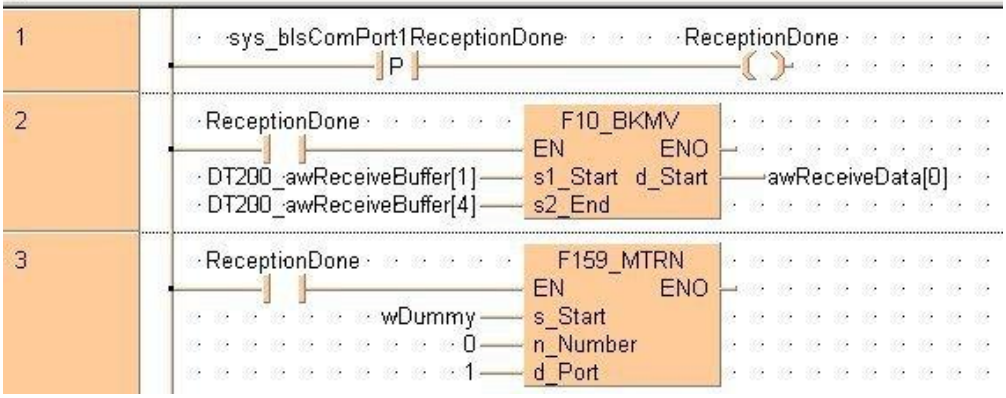
In order to use the data in the receive buffer, define a global variable having the same starting address and capacity. In this example, the starting address is DT200 (VAR_GLOBAL DT200_awReceivedBuffer) and the receive buffer capacity is 5 (ARRAY [0..4] OF WORD).

GVL

	Class	Identifier	FP A...	IEC Addr...	Type	Initial
0	VAR_GLOBAL	DT200_awReceiveBuffer	DT200	%MW5.200	ARRAY [0..4] OF WORD	[5(0)]

POU header and LD body

	Class	Identifier	Type	Initial	Comment
0	VAR	wDummy	WORD	0	
1	VAR	ReceptionDone	BOOL	FALSE	
2	VAR_EXTERNAL	DT200_awReceiveBuffer	ARRAY [0..4] OF WORD	[5(0)]	
3	VAR	awReceiveData	ARRAY[0..3] OF WORD	[4(0)]	



Data can be received from an external device if the "reception done" flag is FALSE. The "reception done" flag is evaluated by the system variable sys_bIsComPort1ReceptionDone. When the reception of the data is complete (the end code has been received), the "reception done" flag turns to TRUE, and subsequently, receiving data is prohibited. To prepare the system to receive the next data without immediately sending further data, the receive buffer is reset by executing F159_MTRN with n_Number = 0.

Note

- The status of the "reception done" flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal flag should be copied to a variable at the beginning of the program.
- The start code "STX" resets the receive buffer. Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. The next data will be stored starting at offset 1 and overwriting the existing data.

6.6.4 Format of send and receive data

Remember the following when accessing data in the send and receive buffers:

- The format of the data in the send buffer depends on the data type of the transmission data (e.g. STRING) and on the conversion function used in the PLC program (e.g. F95_ASC). There is no conversion when data in the send buffer is sent.
- The start and end codes specified in the system registers are automatically added to the data sent. The start code is added at the beginning, the end code at the end of the send string. Do not include start or end codes in the send string.
- The format of the data in the receive buffer depends on the data format used by the external device. Use a conversion function to convert the data into the desired format, e.g. F27_AHEX.
- Start and end codes in the data received are recognized if the corresponding start and end codes have been specified in the system registers. Start and end codes are not stored in the receive buffer. The end code serves as a reception done condition, i.e., the "reception done" flag turns to TRUE when the end code is received. The start code resets the receive buffer.
- If "None" is selected for the start code, a start code is not added to the data sent and is not recognized in the data received. Without start code, the receive buffer can only be reset by executing ClearReceiveBuffer or F159_MTRN.
- If "None" is selected for the end code, an end code is not added to the data sent and is not recognized in the data received. Without end code, the "reception done" flag does not turn to TRUE. The end of reception

can only be determined by a time-out using the IsReceptionDone-ByTimeOut function or by evaluating the data in the receive buffer (see p. 128).

Different end code settings for sending and receiving

Sometimes you do not want to send an end code, but need an end code in the data received to set the "reception done" flag to TRUE. In this case, select the desired end code in the system registers and execute F159_MTRN specifying a negative number for **n_Number**.

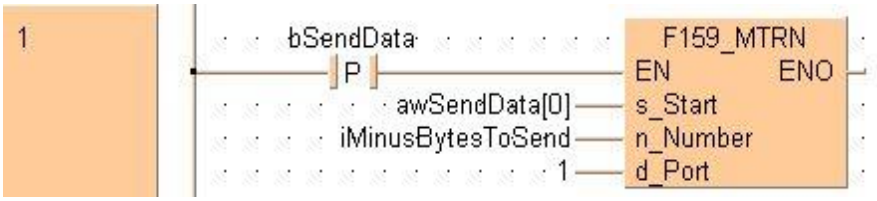
Example

Send 4 bytes of data without adding an end code:

POU Header

	Class	Identifier	Type	Initial	Comment
0	VAR	bSendData	BOOL	FALSE	
1	VAR_CONST...	iMinusBytesToSend	INT	-6	Negative number: No terminator added!
2	VAR	awSendData	ARRAY [0..3] OF WORD	[4(0)]	First word: Number of bytes sent. Words 1 to 3: 6 data bytes to send!
3	VAR				

LD Body




```
if (DF(bSendData)) then
    F159_MTRN(s_Start := awSendData[0], n_Number := iMinusBytesToSend, d_Port := 1);

end_if;
```

6.6.5 Flag operation

Program controlled communication provides for half duplex communication, i.e. communication is possible in both directions, but not simultaneously. Sending data can be controlled by the "transmission done" flag. The end of reception can be verified by:

- evaluating the "reception done" flag or by executing IsReceptionDone
- executing IsReceptionDoneByTimeOut
- directly evaluating the receive buffer.

The flags are special internal flags which turn to TRUE or to FALSE under specific conditions. They can be evaluated using special functions or system variables.

"Reception done" flag

When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. F159_MTRN turns the "reception done" flag to FALSE.

The "reception done" flag can be evaluated using the IsReceptionDone function. Or use the system variable sys_bIsComPort1ReceptionDone or sys_bIsToolPortReceptionDone, depending on the port. The end of reception can also be determined by time-out using the IsReceptionDone-ByTimeOut function or by checking the contents of the receive buffer.

The status of the "reception done" flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal flag should be copied to a variable at the beginning of the program.

Port name	**TOOL**	**COM1**
Port number	0	1
Special internal flag	R903E	R9038
Function name	IsReceptionDone	
System variable name	sys_bIsToolPortReceptionDone	sys_bIsComPort1ReceptionDone
Bit status	TRUE	

"Transmission done" flag

When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE. New data may be sent or received. Any send instruction turns the "transmission done" flag to FALSE and no data can be received.

The "transmission done" flag can be evaluated using the IsTransmissionDone function. Or use the system variable sys_bIsComPort1TransmissionDone or sys_bIsToolPortTransmissionDone, depending on the port.

Port name	TOOL	COM1
Port number°	0	1
Special internal flag	R903F	R9039
Function name	IsTransmissionDone	
System variable name	sys_bIsToolPortTransmission-Done	sys_bIsComPort1Transmission-Done
Bit status	TRUE	

Communication error flag

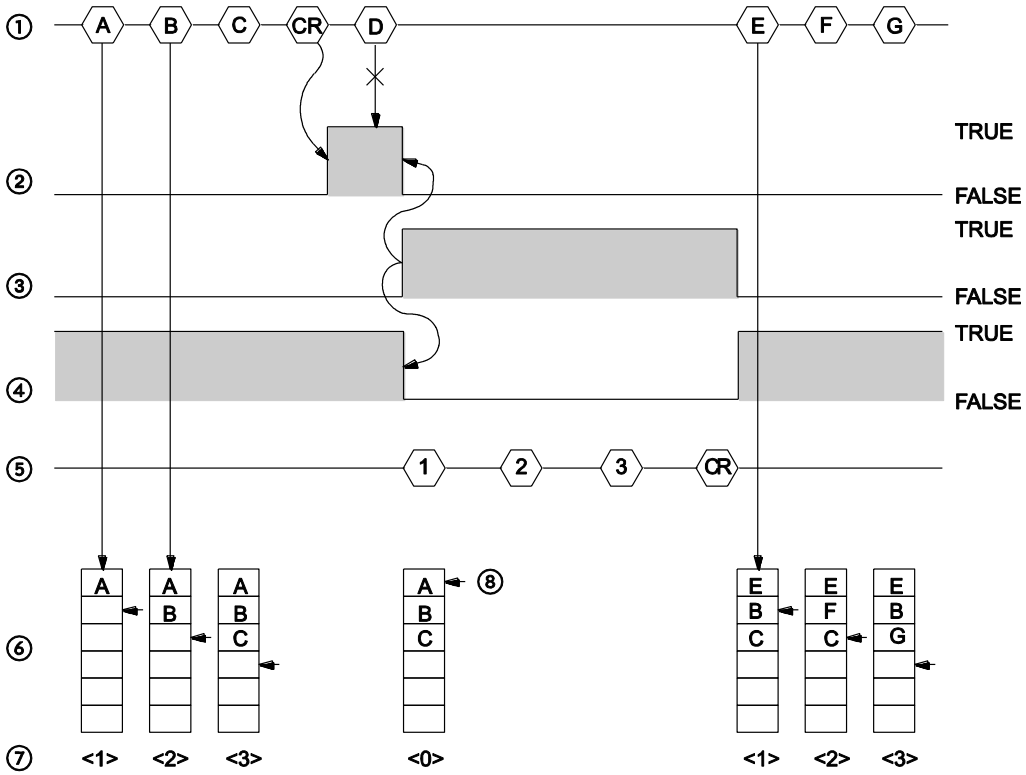
If the communication error flag turns to TRUE during reception, reception continues. Execute a send instruction to turn the error flag to FALSE and to move the write pointer back to offset 1.

The communication error flag can be evaluated using the IsCommunicationError function. Or use the system variable sys_bIsComPort1CommunicationError or sys_bIsToolPortCommunicationError, depending on the port.

Port name	Port number	Special internal flag	Function name	System variable name	Bit status
TOOL	0	R900E	IsCommunicationError	sys_bIsToolPort-CommunicationError	TRUE
COM1	1	R9037		sys_bIsComPort1-CommunicationError	

6.6.5.1 Start code: no-STX, end code: CR

Receiving and sending data:



①	Data received from external device	⑤	Data sent to external device
②	"Reception done" flag	⑥	Receive buffer
③	F159_MTRN execution	⑦	Number of bytes received
④	"Transmission done" flag	⑧	Write pointer

When receiving data, operation is as follows:

1. Characters A, B, and C received from the external device are stored in the receive buffer.
2. When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. (Character D is not stored.)
3. F159_MTRN is executed to send response data to the external device. When F159_MTRN is executed:
   - The receive buffer is reset.
   - The "reception done" flag turns to FALSE.
   - The "transmission done" flag turns to FALSE.
   - The communication error flag turns to FALSE.
   - Characters 1, 2, and 3 are sent to the external device.
   - The end code is automatically added to the data sent.

– While F159_MTRN is being executed, no data can be received. (The "transmission done" flag is FALSE.)

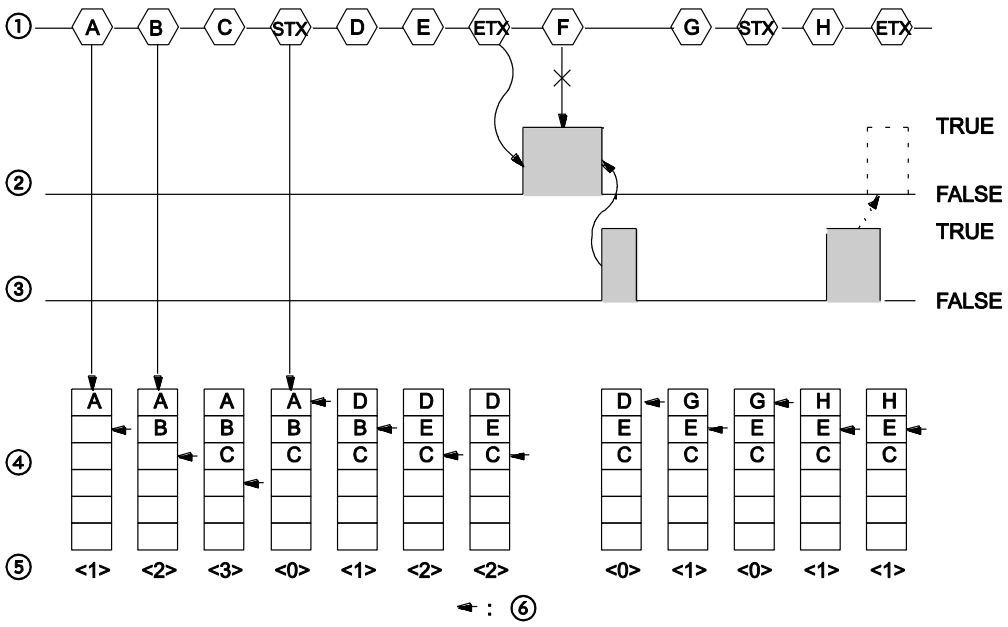
4. When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE.
5. Characters E, F, and G received from the external device are stored in the receive buffer.

Note

Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. The next data will be stored starting at offset 1 and overwriting the existing data.

6.6.5.2 Start code: STX, end code: ETX

Receiving data:



①	Data received from external device	④	Receive buffer
②	"Reception done" flag	⑤	Number of bytes received
③	F159_MTRN execution	⑥	Write pointer

When receiving data, operation is as follows:

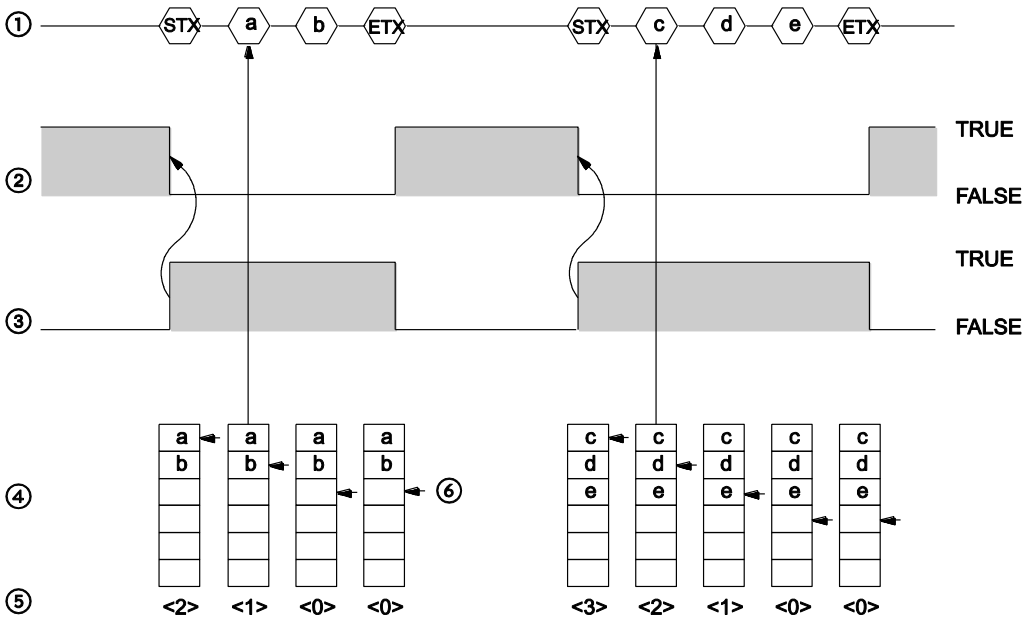
1. Characters A, B, and C received from the external device are stored in the receive buffer.
2. The start code "STX" resets the receive buffer.

3. Characters D and E received from the external device are stored in the receive buffer.
4. When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. (Character F is not stored.)
5. When F159_MTRN is executed:
   - The number of bytes received is set to 0 in offset 0 of the receive buffer.
   - The "reception done" flag turns to FALSE.
6. Character G is stored. (The number of bytes received is set to 1 in offset 0 of the receive buffer.)
7. The start code "STX" resets the receive buffer.
8. Character H is stored.
9. F159_MTRN is executed at the same time the end code is received from the external device. F159_MTRN turns the "reception done" flag to FALSE. Therefore, this flag will not be detected.

Note

- Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. The next data will be stored starting at offset 1 and overwriting the existing data.
- If two start codes are received from the external device, data following the second start code overwrites the data in the receive buffer.

Sending data:



①	Data to be sent	④	Send buffer
②	"Transmission done" flag	⑤	Number of bytes to be sent
③	F159_MTRN execution	⑥	Write pointer

When sending data, operation is as follows:

F159_MTRN is executed to send data to the external device. When F159_MTRN is executed:

1. The "transmission done" flag turns to FALSE.
2. The start code is sent automatically.
3. The number of bytes to be sent is set in offset 0 of the send buffer.
4. The characters a and b are sent to the external device.
   - The end code is automatically added to the data sent.
   - While F159_MTRN is being executed, no data can be received. (The "transmission done" flag is FALSE.)
5. When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE.
6. Now, F159_MTRN can be executed again. When F159_MTRN is executed: Steps 1 to 5 are repeated. This time, the characters c, d, and e are sent.

6.6.6 1:1 communication

System register settings

By default, the COM port is set to MEWTOCOL-COM mode. For 1:1 program controlled communication, the system registers should be set as shown below.

Settings for COM port 1 (or TOOL port):

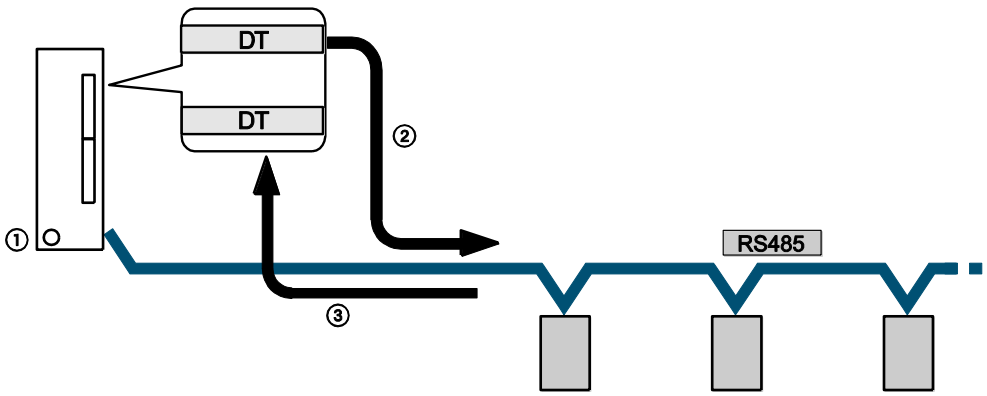
Name	Name	Set value
412	COM port 1 - communication mode	Program controlled
413	COM port 1 - communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
415	COM port 1 - baud rate	2400–115200bit/s
416 (420)	COM port 1 - receive buffer starting address	0–32764 (factory setting: 0) (see note)
417 (421)	COM port 1 - receive buffer capacity	0–2048 words (factory setting: 2048 words)

Note

When using C10, C14, or C16, the range is 0–12312.

6.6.7 1:N communication

The FP0R and the external devices are connected using an RS485 cable. A protocol that matches the external devices and the instruction F159_MTRN (or any instruction that uses F159_MTRN implicitly) are used to send and receive data.



①	PLC
②	Sending data using a send instruction
③	Receiving data in receive buffer

System register settings

By default, the COM port is set to MEWTOCOL-COM mode. For 1:N program controlled communication, the system registers should be set as shown below.

Settings for COM port 1 (or TOOL port):

Name	Name	Set value
412	COM port 1 - communication mode	Program controlled
413	COM port 1 - communication format [1)]	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
415	COM port 1 - baud rate [1)]	2400–115200bit/s
416 (420)	COM port 1 - receive buffer starting address	0–32762 (factory setting: 0)
417 (421)	COM port 1 - receive buffer capacity	0–2048 words (factory setting: 2048 words)

[1)] The setting must match the external device connected to the communication port.

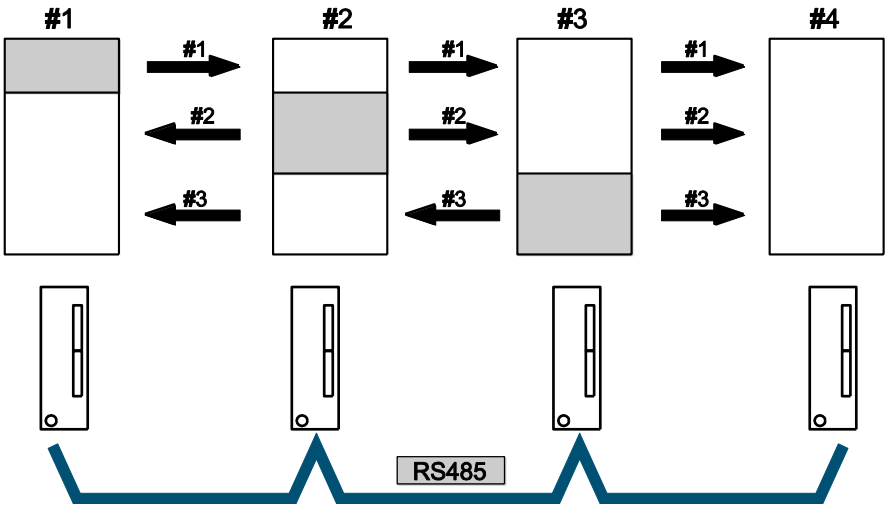
6.6.8 Programming in FP0 compatibility mode

Make sure that the PLC type selected in Control FPWIN Pro is "FP0".

In the FP0 compatibility mode, the instruction F144_TRNS is used instead of F159_MTRN.

6.7 PLC Link

PLC Link is an economic way of linking PLCs using a twisted-pair cable and the MEWNET protocol. Data is shared with all PLCs by means of dedicated internal flags called link flags (L) and data registers called link registers (LD). The statuses of the link flags and link registers of one PLC are automatically fed back to the other PLCs on the same network. The link flags and link registers of the PLCs contain areas for sending and areas for receiving data. Station numbers and link areas are allocated using the system registers.

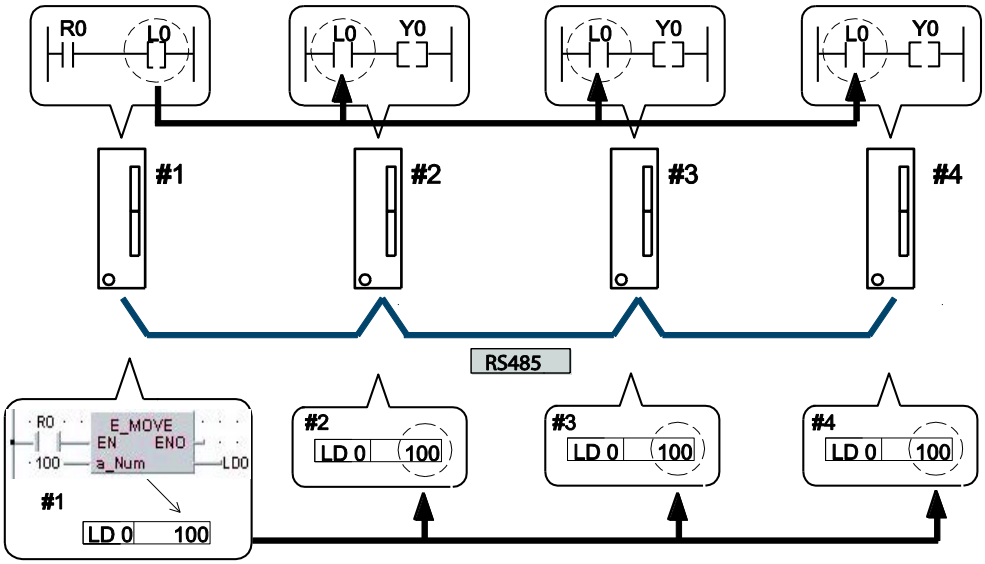


Sharing of data in a PLC link using dedicated send and receive areas

	Send area		Receive area	#	Station number of PLC

Example

Link flag L0 for station #1 turns to TRUE. The status change is fed back to the programs of the other stations, and Y0 of the other stations is set to TRUE. A constant of 100 is written to link register LD0 of station #1. The contents of LD0 in the other stations are also changed to a constant of 100.



PLC Link connection between four FP0R units

#	Station number of PLC	LD	Link register

Panasonic PLCs available for PLC Link

- FP0R (RS485 type)
- FP7 (using RS485 type communication cassette)
- FPΣ (using RS485 type communication cassette)
- FP-X (using RS485 type communication cassette)
- FP2-MCU (using RS485 type communication cassette)

6.7.1 Setting communication parameters

Make the following settings for the communication port:

- communication mode (PLC Link)
- station number
- link area

For details on setting the communication parameters, see "Setting system registers in PROG mode" on p. 102. For details on setting the link area, see "Link area allocation" on p. 144.

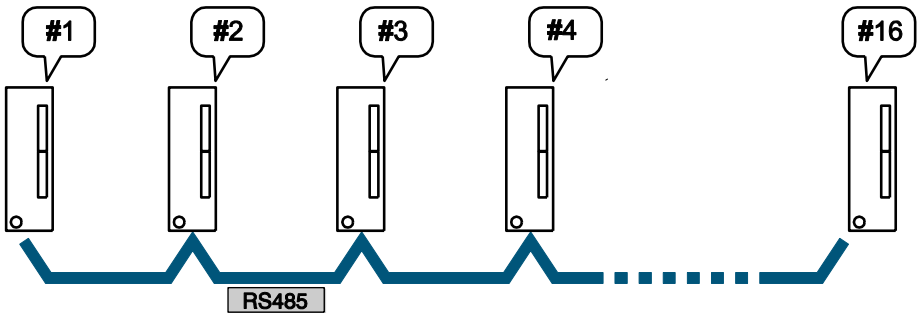
Note

- PLC Link is only available via the COM port.
- For RS232C connections, the maximum number of stations is 2.
- For PLC Link, the communication format and baud rate settings are fixed:

Data length:	8 bits
Parity:	Odd
Stop bit:	1 bit
Start code:	No STX
End code:	CR, use SendCharactersAndClearString for end code suppression
Baud rate	115200bit/s

Station number setting for a PLC link

The station number can be set within a range of 1 to 16. For details on setting station numbers, see p. 102.



A maximum of 16 stations can be connected in a PLC link

#	Station number of PLC

Note

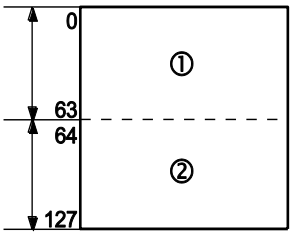
- Make sure the same station number is not used for more than one of the PLCs connected through the PLC Link function.
- If there are fewer than 16 stations linked, set the highest station number to reduce the link transmission cycle time. Station numbers should be set sequentially and consecutively, starting from 1, with no breaks between them. See "Setting the highest station number for a PLC link" on p. 151.

6.7.2 Link area allocation

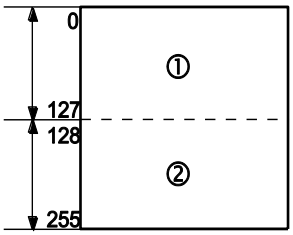
To use the PLC link function, link areas need to be allocated. Set the allocations for both the link flags and link registers using the system registers of the CPU.

Link areas consist of link flags and link registers and are divided into areas for PLC link 0 and PLC link 1. A maximum of 1024 link flags (bits) and 128 link registers (words) can be used in the PLC link areas.

Link flags



Link registers



Unit: words

①	For PLC link 0: 1024 bits (1st half)	①	For PLC link 0: 128 words (1st half)
②	For PLC link 1: 1024 bits (2nd half)	②	For PLC link 1: 128 words (2nd half)

System registers

No.		Name	Default value	Set values
	46	PLC link 0 and 1 allocation setting	Use PLC link 0	Use PLC link 0 Use PLC link 1
PLC link 0	40	Link flags - Send/receive area - Number of words shared by all linked PLCs	0	0-64 words
	41	Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0-128 words
	42	Link flags - Send area - Start sending from this word address	0	0-63
	43	Link flags - Send area - Number of words to send	0	0-64 words
	44	Link registers - Send area - Start sending from this word address	0	0-127
	45	Link registers - Send area - Number of words to send	0	0-128 words
	47[1)]	Highest station number in network	16	1-16

No.		Name	Default value	Set values
PLC link 1	50	Link flags - Send/receive area - Number of words shared by all linked PLCs	0	0-64 words
	51	Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0-128 words
	52	Link flags - Send area - Start sending from this word address	64	64-127
	53	Link flags - Send area - Number of words to send	0	0-64 words
	54	Link registers - Send area - Start sending from this word address	128	128-255
	55	Link registers - Send area - Number of words to send	0	0-128 words
	57[1)]	Highest station number in network	0	0-16

[1)] Set the same value for all PLCs in the link.

Note

Use the SYS2 instruction to set the link area in RUN mode. Please refer to the Control FPWIN Pro online help for detailed information.

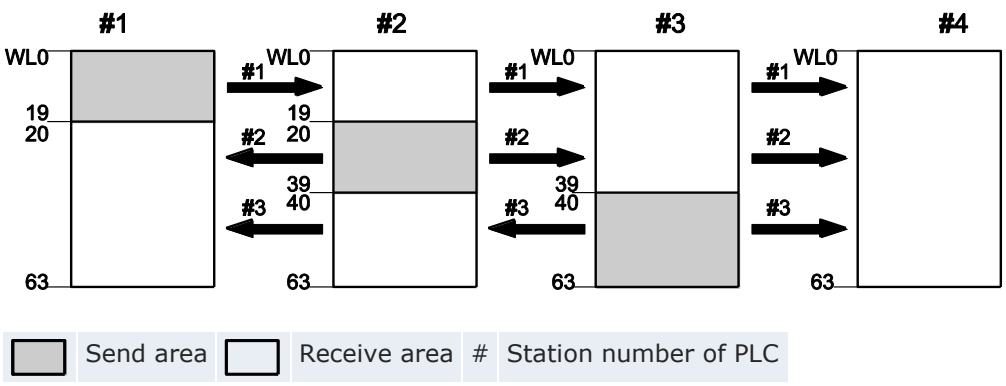
Using PLC link 1

You can either use PLC link 0 or PLC link 1. Set system register 46 to "Reverse" to use PLC link 1. See "PLC link 0 and 1 allocation setting" on p. 152.

6.7.2.1 Example for PLC link 0

The PLC link areas are divided into send and receive areas. The link flags and link registers are transmitted from the send area to the receive area of a different PLC. The link flags and registers in the receive area on the receiving side must be within the same area as on the sending side.

Link flag allocation

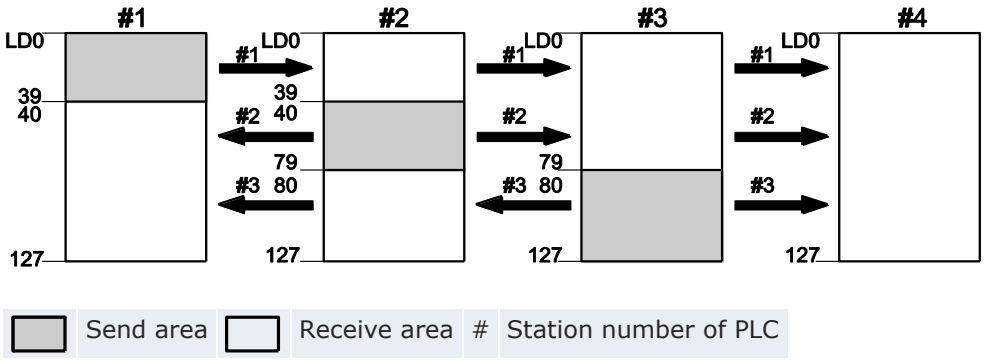


System register settings

No.	Name	Station settings			
		#1	#2	#3	#4
40[1)]	Link flags - Send/receive area - Number of words shared by all linked PLCs	64	64	64	64
42	Link flags - Send area - Start sending from this word address	0	20	40	0
43	Link flags - Send area - Number of words to send	20	20	24	0

1) The value of this system register must be identical for all stations.

Link register allocation



System register settings

No.	Name	Station settings			
		#1	#2	#3	#4
41[1)]	Link registers - Send/receive area - Number of words shared by all linked PLCs	128	128	128	128
44	Link registers - Send area - Start sending from this word address	0	40	80	0
45	Link registers - Send area - Number of words to send	40	40	48	0

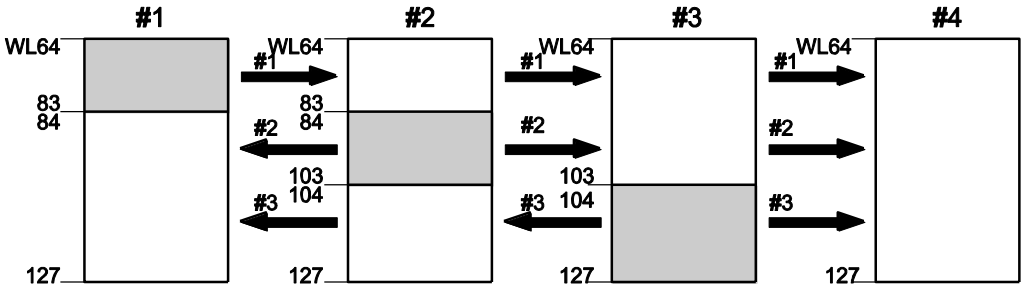
1) The value of this system register must be identical for all stations.

When link areas are allocated as shown above, the send area of station no. 1 can be transmitted to the receive areas of stations no. 2, 3, and 4. Also, the receive area of station no. 1 can receive data from the send areas of stations no. 2 and 3. Station no. 4 is allocated as a receive area only and can receive data from stations no. 1, 2, and 3, but cannot send data to other stations.

6.7.2.2 Example for PLC link 1

Set system register 46 to "Reverse" to use PLC link 1. See "PLC link 0 and 1 allocation setting" on p. 152.

Link flag allocation

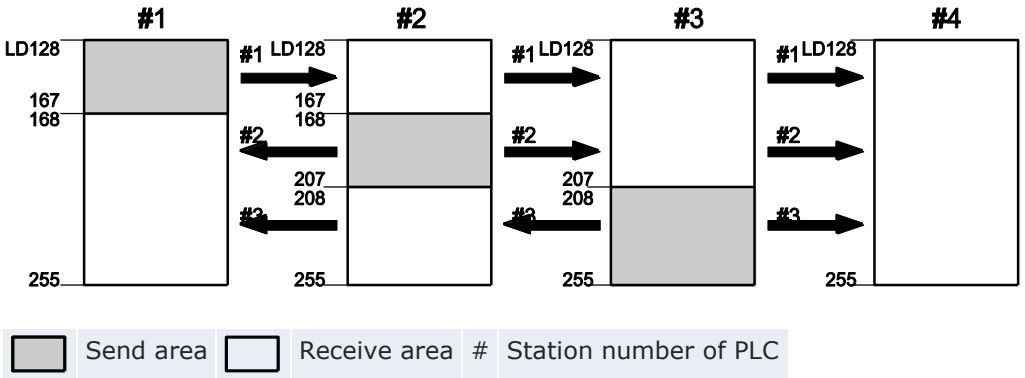


System register settings

Name	Name	Station settings			
		#1	#2	#3	#4
50[1)]	Link flags - Send/receive area - Number of words shared by all linked PLCs	64	64	64	64
52	Link flags - Send area - Start sending from this word address	64	84	104	64
53	Link flags - Send area - Number of words to send	20	20	24	0

1) The value of this system register must be identical for all stations.

Link register allocation



System register settings

Name	Name	Station settings			
		#1	#2	#3	#4
51[1)]	Link registers - Send/receive area - Number of words shared by all linked PLCs	128	128	128	128
54	Link registers - Send area - Start sending from this word address	128	168	208	128
55	Link registers - Send area - Number of words to send	40	40	48	0

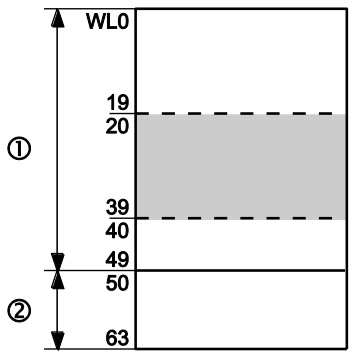
1) The value of this system register must be identical for all stations.

When link areas are allocated as shown above, the send area of station no. 1 can be transmitted to the receive areas of stations no. 2, 3, and 4. Also, the receive area of station no. 1 can receive data from the send areas of stations no. 2 and 3. Station no. 4 is allocated as a receive area only and can receive data from stations no. 1, 2, and 3, but cannot send data to other stations.

6.7.2.3 Partial use of link areas

In the link areas available for PLC link, link flags with a total of 1024 points (64 words) and link registers with a total of 128 words can be used. This does not mean, however, that it is necessary to reserve the entire area. Parts of the area which have not been reserved can be used as internal flags and internal registers.

Link flag allocation



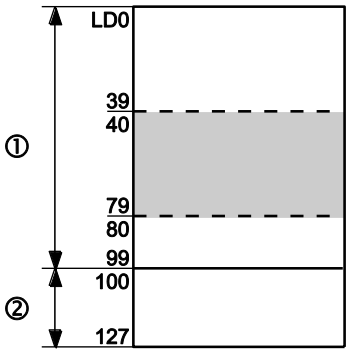
	Send area
	Receive area
	Area for internal flags
①	Used for link flags
②	Not used for link flags

System register settings

Name	Name	#1
40	Link flags - Send/receive area - Number of words shared by all linked PLCs	50
42	Link flags - Send area - Start sending from this word address	20
43	Link flags - Send area - Number of words to send	20

With the above settings for station number 1, the 14 words (224 points) consisting of WL50 to WL63 can be used as internal flags.

Link register allocation



	Send area
	Receive area
	Area for internal registers
①	Used for link registers
②	Not used for link registers

System register settings

Name	Name	#1
41	Link registers - Send/receive area - Number of words shared by all linked PLCs	100
44	Link registers - Send area - Start sending from this word address	40
45	Link registers - Send area - Number of words to send	40

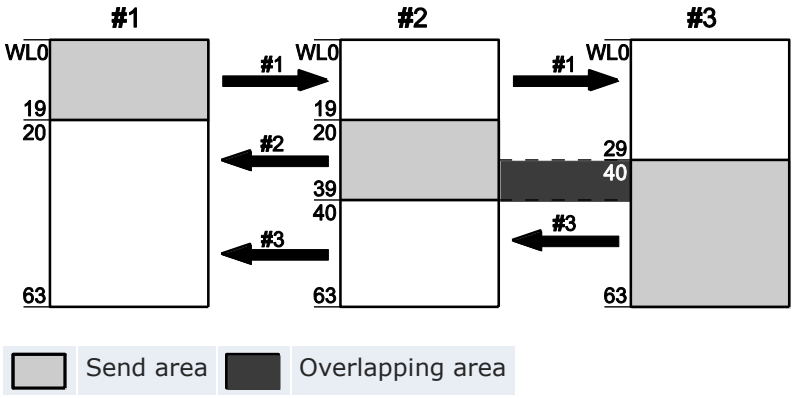
With the above settings for station number 1, the 28 words consisting of LD100 to LD127 can be used as internal registers.

6.7.2.4 Precautions for allocating link areas

A mistake in the link area allocation will cause an error, and communication will be disabled.

Avoid overlapping send areas

When sending data from the send area to the receive area of another PLC, send and receive areas must match. In the example shown below, there is an overlapping area between units no. 2 and 3, and this will cause an error, so that communication cannot be carried out.



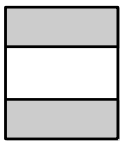
System register settings

No.	Name	Station settings		
		#1	#2	#3
40	Link flags - Send/receive area - Number of words shared by all linked PLCs	64	64	64
42	Link flags - Send area - Start sending from this word address	0	20	30
43	Link flags - Send area - Number of words to send	20	20	34

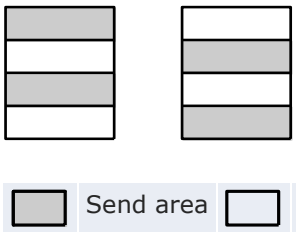
Invalid allocations

The following allocations are not possible, neither for link flags nor for link registers:

- Send area is split



- Send and receive areas are split into multiple segments



6.7.3 Setting the highest station number for a PLC link

Station numbers should be set sequentially and consecutively, starting from 1, with no breaks between them. If there is a missing station number, or if there is a station for which the power supply has not been turned on, the response time for the PLC link (the link transmission cycle time) will be longer (see p. 155).

If there are fewer than 16 stations linked, set the highest station number to reduce the link transmission cycle time. (The default value is 16.) Set the same value for all PLCs in the link.

The highest station number is set using system register no. 47 for PLC link 0 or system register no. 57 for PLC link 1.

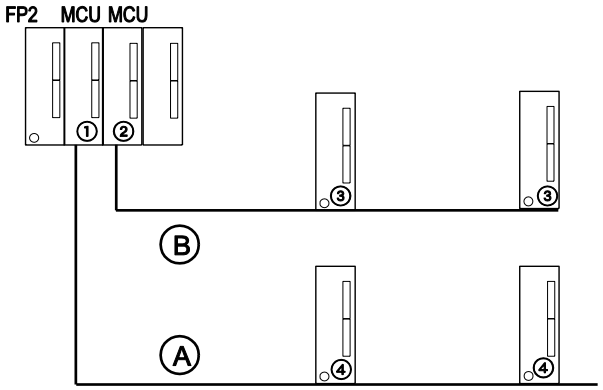
Sample settings

Total number of stations:	2		4				n
Station number:	1	2	1	2	3	4	n
Highest station number [1)]:	2	2	4	4	4	4	N

[1)] Same setting for each station

6.7.4 PLC link 0 and 1 allocation setting

For PLCs supporting two PLC links, the default setting of the system register "PLC link 0 and 1 allocation setting" is "Use PLC link 0". This means that the unit which is closest to the CPU uses PLC link 0 and the unit which is further away uses PLC link 1. To reverse this behavior, select "Use PLC link 1".



①	In the default setting ("Use PLC link 0"), the first half of the link relays and link registers is used (WL0-WL63, LD0-LD127).
②	In the default setting ("Use PLC link 0"), the second half of the link relays and link registers is used (WL64-WL127, LD 128-LD225).
③	Set "Use PLC link 0" in the system registers.
④	Set "Use PLC link 1" in the system registers.
Ⓐ	PLC link 0
Ⓑ	PLC link 1

6.7.5 Monitoring

When using a PLC link, the operation status of the links can be monitored using the flags below. In FPWIN Pro, choose **Monitor** → **Special flags and registers** → **PLC link status** to view the status of each flag.

To monitor other PLC link status items, such as the transmission cycle time and the number of times that errors have occurred, choose **Monitor** → **PLC link status** in FPWIN Pro.

Remote programming of other linked PLCs is not possible.

Note

To access special data registers and special internal flags, use the PLC-independent system variables.

Transmission assurance flags

- For PLC link 0: R9060 to R906F (correspond to station no. 1 to 16)
- For PLC link 1: R9080 to R908F (correspond to station no. 1 to 16)

Before using the data from a different station in the network, check to make sure the transmission assurance flag for this station is TRUE.

Flag no.	Station no.	System variable name	Conditions for TRUE/FALSE
R9060	1	sys_bIsPlcLink0Station1Active	TRUE: • if the PLC link is normal FALSE: • if transmission has been stopped, or • if a problem has occurred, or • if a PLC link is not being used
R9061	2	sys_bIsPlcLink0Station2Active	
R9062	3	sys_bIsPlcLink0Station3Active	
R9063	4	sys_bIsPlcLink0Station4Active	
R9064	5	sys_bIsPlcLink0Station5Active	
R9065	6	sys_bIsPlcLink0Station6Active	
R9066	7	sys_bIsPlcLink0Station7Active	
R9067	8	sys_bIsPlcLink0Station8Active	
R9068	9	sys_bIsPlcLink0Station9Active	
R9069	10	sys_bIsPlcLink0Station10Active	
R906A	11	sys_bIsPlcLink0Statio11Active	
R906B	12	sys_bIsPlcLink0Station12Active	
R906C	13	sys_bIsPlcLink0Station13Active	
R906D	14	sys_bIsPlcLink0Station14Active	
R906E	15	sys_bIsPlcLink0Station15Active	
R906F	16	sys_bIsPlcLink0Station16Active	

Operation mode flags

- For PLC link 0: R9070 to R907F (correspond to station nos. 1 to 16)
- For PLC link 1: R9090 to R909F (correspond to station nos. 1 to 16)

The operation modes (RUN/PROG) can be checked for any given PLC.

Flag no.	Station no.	System variable name	Conditions for TRUE/FALSE
R9070	1	sys_bIsPlcLink0Station1InRunMode	TRUE: • if the unit is in RUN mode FALSE: • if the unit is in PROG mode
R9071	2	sys_bIsPlcLink0Station2InRunMode	
R9072	3	sys_bIsPlcLink0Station3InRunMode	
R9073	4	sys_bIsPlcLink0Station4InRunMode	
R9074	5	sys_bIsPlcLink0Station5InRunMode	
R9075	6	sys_bIsPlcLink0Station6InRunMode	
R9076	7	sys_bIsPlcLink0Station7InRunMode	
R9077	8	sys_bIsPlcLink0Station8InRunMode	
R9078	9	sys_bIsPlcLink0Station9InRunMode	
R9079	10	sys_bIsPlcLink0Station10InRunMode	
R907A	11	sys_bIsPlcLink0Station11InRunMode	
R907B	12	sys_bIsPlcLink0Station12InRunMode	
R907C	13	sys_bIsPlcLink0Station13InRunMode	
R907D	14	sys_bIsPlcLink0Station14InRunMode	
R907E	15	sys_bIsPlcLink0Station15InRunMode	
R907F	16	sys_bIsPlcLink0Station16InRunMode	

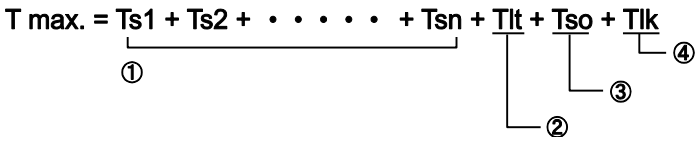
PLC link transmission error flag R9050

This flag turns to TRUE if a problem is detected during transmission.

Flag no.	Station no.	System variable name	Conditions for TRUE/FALSE
R9050	1–16	sys_bIsPlcLink0-TransmissionError	TRUE: • if a transmission error has occurred in the PLC link, or • if an error has occurred in the setting of the PLC link area FALSE: • if there are no transmission errors

6.7.6 PLC link response time

The maximum value for the transmission time (T) of one cycle can be calculated using the following formula.



① Ts (transmission time per station) = scan time + Tpc

Tpc = Ttx × Pcm

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Pcm = 23 + (number of relay words + number of register words) × 4

Tpc = PLC link sending time
Ttx = sending time per byte
Pcm = PLC link sending size

② Tlt (link table sending time) = Ttx × Ltm

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Ltm = 13 + 2 × n

Ttx = sending time per byte
Ltm = link table sending size
n = number of stations being added

③ Tso (master station scan time)

The master station scan time should be confirmed using the programming tool.

④ Tlk (link addition processing time) = Tlc + Twt + Tls+ Tso

If no stations are being added, Tlk = 0.

Tlc = 10 × Ttx

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Twt = Initial value 400ms (can be changed using SYS1 instruction)

Tls = 7 × Ttx

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Tlc = link addition command sending time
Twt = addition waiting time
Ttx = sending time per byte
Tls = link error stop command sending time
Tso = master station scan time

Ttx = sending time per byte
Tso = master station scan time

Calculation example 1

Conditions: All stations have been added to a 16-unit link. Highest station number = 16. Flags and registers have been evenly allocated. Scan time for each PLC: 1ms.

Ttx = 0.096

Pcm (per station) = 23 + (4 + 8) × 4 = 71
Tpc = Ttx × Pcm = 0.096 × 71 ≈ 6.82ms
Ts (per station) = 1 + 6.82 = 7.82ms
Tlt = 0.096 × (13 + 2 × 16) = 4.32ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be: T max. = 7.82 × 16 + 4.32 + 1 = 130.44ms

Calculation example 2

Conditions: All stations have been added to a 16-unit link. Highest station number = 16. Flags and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Pcm (per station) = 23 + (4 + 8) × 4 = 71
Tpc = Ttx × Pcm = 0.096 × 71 ≈ 6.82ms
Ts (per station) = 5 + 6.82 = 11.82ms
Tlt = 0.096 × (13 + 2 × 16) = 4.32ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be: T max. = 11.82 × 16 + 4.32 + 5 = 198.44ms

Calculation example 3

Conditions: All but one station have been added to a 16-unit link. Highest station number = 16. Flags and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Ts (per station) = 5 + 6.82 = 11.82ms
Tlt = 0.096 × (13 + 2 × 15) = 4.31ms
Tlk = 0.96 + 400 + 0.67 + 5 ≈ 407ms

Note: The default value for the addition waiting time is 400ms.

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be: T max. = 11.82 × 15 + 4.13 + 5 + 407 = 593.43ms

Calculation example 4

Conditions: All stations have been added to an 8-unit link. Highest station number = 8. Flags and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Pcm (per station) = 23 + (8 + 16) × 4 = 119
Tpc = Ttx × Pcm = 0.096 × 119 ≈ 11.43ms
Ts (per station) = 5 + 11.43ms = 16.43ms
Tlt = 0.096 × (13 + 2 × 8) = 2.79ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be: T max. = 16.43 × 8 + 2.79 + 5 = 139.23ms

Calculation example 5

Conditions: All stations have been added to a 2-unit link. Highest station number = 2. Flags and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Pcm (per station) = 23 + (32 + 64) × 4 = 407
Tpc = Ttx × Pcm = 0.096 × 407 ≈ 39.072ms
Ts (per station) = 5 + 39.072 = 44.072ms
Tlt = 0.096 × (13 + 2 × 2) ≈ 1.632ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be: T max. = 44.072 × 2 + 1.632 + 5 = 94.776ms

Calculation example 6

Conditions: All stations have been added to a 2-unit link. Highest station number = 2. 32 relays and 2 register words have been evenly allocated. Scan time for each PLC: 1ms.

Ttx = 0.096

Pcm (per station) = 23 + (1 + 1) × 4 = 31
Tpc = Ttx × Pcm = 0.096 × 31 ≈ 2.976ms
Ts (per station) = 1 + 2.976 = 3.976ms
Tlt = 0.096 × (13 + 2 × 2) ≈ 1.632ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be: T max. = 3.976 × 2 + 1.632 + 1 = 10.584ms

Note

- In the description, "stations that have been added" refers to stations which are connected between station no. 1 and the highest station number and for which the power supply has been turned on.
- Comparing examples 2 and 3, the transmission cycle time is longer if there is one station that has not been added to the link. As a result the PLC link response time is longer.
- The SYS1 instruction can be used to minimize the transmission cycle time even if there are one or more stations that have not been added to the link.

6.7.6.1 Reducing transmission cycle times

If there are stations that have not been added to the link, the link addition processing time (Tlk) and with this the transmission cycle time will be longer.

$$T\ max. = Ts1 + Ts2 + \cdots\cdots + Tsn + Tlt + Tso + Tlk$$

$$Tlk = Tlc + Twt + Tls + Tso$$

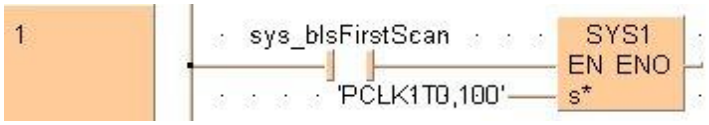
Tlk= link addition processing time
Tlc = link addition command sending time
Twt = addition waiting time
Tls = link error stop command sending time
Tso = master station scan time

With the SYS1 instruction, the link addition waiting time (Twt) in the above formula can be reduced. Thus, SYS1 can be used to minimize the increase in the transmission cycle time.

Example

Set SYS1 to change the waiting time for a link to be added to the PLC link from the default value of 400ms to 100ms.

LD Body



Note

- If there are any stations that have not been added to the link, the setting should not be changed as long as a longer link transmission cycle time does not cause any problems.
- The SYS1 instruction should be executed at the beginning of the program, at the rise of R9014. The same waiting time should be set for all linked PLCs.
- The waiting time should be set to a value of at least twice the maximum scan time for any of the PLCs connected to the link.
- If a short waiting time has been set, there may be PLCs that cannot be added to the link even if their power supply is on. (The shortest time that can be set is 10ms.)

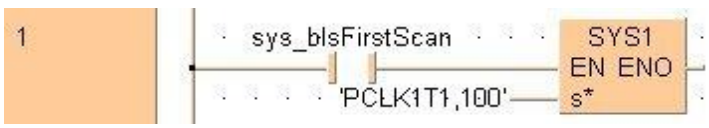
6.7.6.2 Error detection time for transmission assurance flags

If the power supply of any given PLC fails or is turned off, it takes (as a default value) 6.4 seconds for the transmission assurance flag of that PLC to be turned off at the other stations. This time period can be shortened using the SYS1 instruction.

Example

Set SYS1 to change the time that the PLC link transmission assurance is off from the default value of 6.4s to 100ms.

LD Body

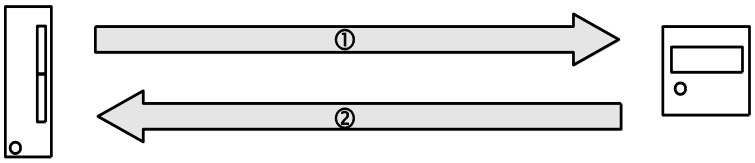


Note

- The setting should not be changed as long as a longer transmission assurance flag detection time does not cause any problems.
- The SYS1 instruction should be executed at the beginning of the program, at the rise of R9014. The same waiting time should be set for all linked PLCs.
- The time should be set to a value of at least twice the maximum transmission cycle time when all of the PLCs are connected to the link.
- If a short time has been set, the transmission assurance flag may not function properly. (The shortest time that can be set is 100ms.)

6.8 Modbus RTU communication

The Modbus RTU protocol enables the communication between the FP0R and other devices (including the Panasonic FP-e PLCs, touch panels of the GT series and KT temperature controllers as well as Modbus devices by other manufacturers). The master station sends instructions (command messages) to the slave stations and the slave stations respond (send response messages) based on the instructions received. The master station has read and write access to a maximum number of 99 slave stations.



Modbus RTU connection between the FP0R and an external device

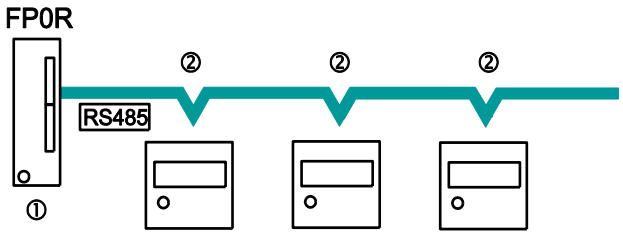
① Command message ② Response message

Note

The Modbus protocol supports both ASCII mode and RTU binary mode. However, the PLCs of the FP Series only support the RTU binary mode.

Modbus RTU master function

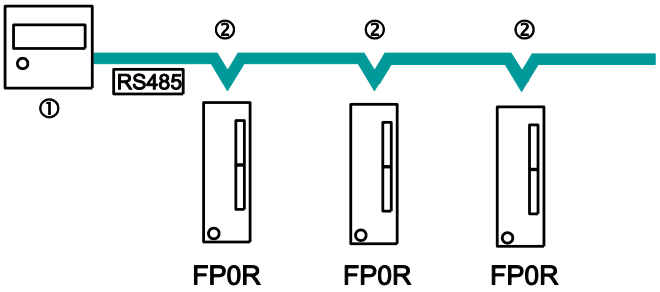
Write and read access to various slaves is possible using the F145 and F146 instructions. Individual access to each slave as well as global transmission is possible.



① Master ② Slave

Modbus RTU slave function

After having received a command message from the master station, the slave stations send back the response message based on the instructions received. Do not execute the F145_WRITE and F146_READ instructions on slave stations.



① Master ② Slave

Modbus RTU command message frame

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
Transmission time for 3.5 characters	8 bits	8 bits	n × 8 bits	16 bits	Transmission time for 3.5 characters

ADDRESS (station no.)	8 bits, 0–99 (decimal)[1)] 0 = broadcast address
FUNCTION	8 bits
DATA	Varies depending on the commands.
CRC	16 bits
END	Transmission time for 3.5 characters (differs depending on baud rate). Please refer to "Reception done judgment time".

[1)] Control FPWIN Pro does not support the address range from 0–247 of the Modbus RTU protocol.

Response in normal status

The same message as a command is returned for a single write command. A part of a command message (6 bytes from the beginning) is returned for a multiple write command.

Response in abnormal status

If a parameter which is to be processed but is disabled is found in a command (except for a transmission error):

ADDRESS	FUNCTION + 80H	ERROR CODE	CRC

ERROR CODE:
1: Illegal function
2: Illegal data address (no word address)
3: Illegal data value (not a multiple of 16)

Reception done judgment time

The process for receiving a message is complete after all data has been received and the time given in this table has been reached.

Baud rate	Reception done judgment time
2400	≈13.3ms
4800	≈6.7ms
9600	≈3.3ms
19200	≈1.7ms
38400	≈0.8ms
57600	≈0.6ms
115200	≈0.3ms

Supported commands

Executable instructions for master	Code (decimal)	Name (Modbus original)	Name for FP0R	Modbus Reference
F146_READ	01	Read Coil Status	Read Y and R Coils	0X
F146_READ	02	Read Input Status	Read X Input	1X
F146_READ	03	Read Holding Registers	Read DT	4X
F146_READ	04	Read Input Registers	Read WL and LD	3X
F145_WRITE	05	Force Single Coil	Write Single Y and R	0X
F145_WRITE	06	Preset Single Register	Write DT 1 Word	4X
Cannot be issued	08	Diagnostics	Loopback Test	–
F145_WRITE	15	Force Multiple Coils	Write Multiple Ys and Rs	0X
F145_WRITE	16	Preset Multiple Registers	Write DT Multiple Words	4X
Cannot be issued	22	Mask Write 4X Register	Write DT Mask	4X
Cannot be issued	23	Read/Write 4X Registers	Read/Write DT	4X

Modbus references and FP0R addresses

Modbus reference				PLC address
Name		Decimal address[1]	Hexadecimal address[2]	
Coil		000001–001760	0000–06DF	Y0–Y109F
		002049–006144	0800–17FF	R0–R255F
Input		100001–001760	0000–06DF	X0–X109F
Holding register	C10, C14, C16	400001–412315	0000–301B	DT0–DT12314
	C32, T32, F32	40001–432765	0000–7FFC	DT0–DT32764
Input register		300001–300128	0000–007F	WL0–WL127
		302001–302256	07D0–08CF	LD0–LD255

[1] Beginning with 0

[2] Beginning with 1

Reference

For details on Modbus settings and communication, please refer to the F145_WRITE_DATA and F146_READ_DATA commands in the FPWIN Pro online help.

6.8.1 Setting communication parameters

Make the following settings for the communication port:

- communication mode (Modbus RTU)
- station number
- baud rate
- communication format

For details on setting the communication parameters, see "Setting system registers in PROG mode" on p. 102.

Note

- The station number can be set within a range of 1 to 99.
- With a C-NET adapter, a maximum of 32 stations can be specified.

6.8.2 Sample program for master communication

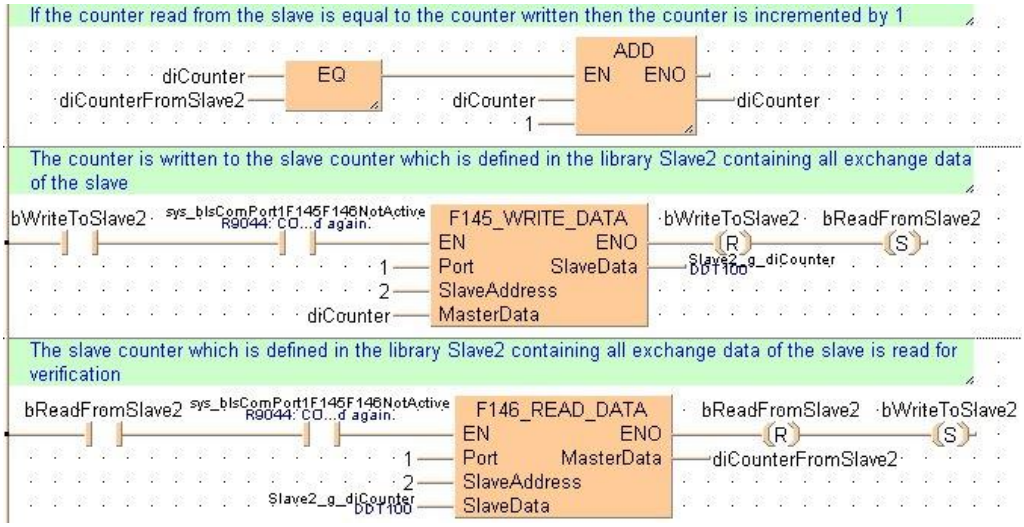
Use the F145_WRITE and F146_READ instructions for the Modbus master function. Be sure to select "Modbus RTU Master/Slave" for the COM port in system register 412.

POU Header

	Class	Identifier	Type	Initial
0	VAR_EXTERNAL	Slave2_g_diCounter	DINT	0
1	VAR	diCounter	DINT	0
2	VAR	diCounterFromSlave2	DINT	-1
3	VAR	bWriteToSlave2	BOOL	TRUE
4	VAR	bReadFromSlave2	BOOL	FALSE

In order to have consistent data in the master project and in the slave project, the common data should be kept in the GVL of a common library.

LD Body



Reference

For details on Modbus settings and communication, please refer to the F145_WRITE_DATA and F146_READ_DATA commands in the FPWIN Pro online help.

Chapter 7

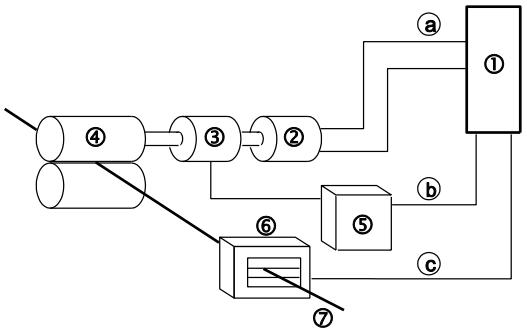
High-speed counter and pulse output

7.1 Overview

Three built-in hardware extensions allow the FP0R to be used for positioning control and measurement: high-speed counting, pulse output, and PWM (pulse-width modulation) output.

High-speed counter function

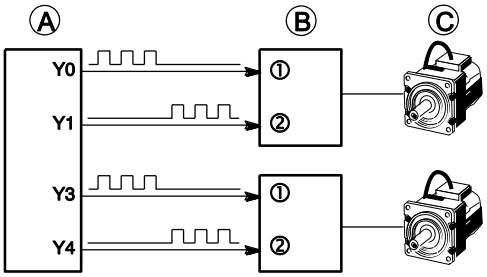
The high-speed counter function counts external inputs such as those from sensors or encoders. When the count reaches the target value, this function turns the desired output to TRUE or to FALSE.



①	PLC		
②	Encoder	ⓐ	Encoder output is input to the high-speed counter
③	Motor		
④	Roller		
⑤	Inverter	ⓑ	Start/stop signal
⑥	Cutter	ⓒ	Cutter blade control signal
⑦	Tape, lead wire		

Pulse output function

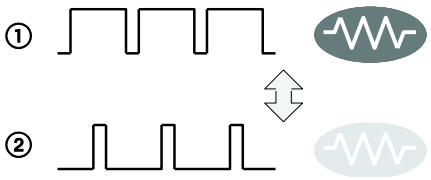
By connecting a commercially available motor driver to the PLC, positioning control can be performed with the pulse output function. Using special instructions, trapezoidal control, home return, or JOG operation is possible.



Ⓐ	PLC	①	CW pulse output
Ⓑ	Motor driver	②	CCW pulse output
Ⓒ	Stepping motor/servo motor		

PWM output function

A special instruction makes it possible to output pulses with a specified duty ratio.



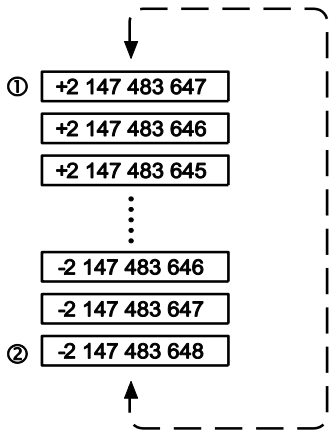
Heater control using the PWM output function

①	Increasing the pulse width increases heating
②	Decreasing the pulse width decreases heating

Counting Range

The counting range of the built-in high-speed counter is from -2 147 483 648 to 2 147 483 647 (32-bit binary number).

The high-speed counter is a ring counter. Consequently, if the counted value exceeds the maximum value, the counter returns to the minimum value. Similarly, if the counted value drops below the minimum value, the counter goes back to the maximum value and continues counting from there.



①	Maximum value
②	Minimum value

Note

Using linear interpolation instruction F175_PulseOutput_Linear or PulseOutput_Linear_FB: The target value or the amount of travel must be within the range of -8 388 608 to +8 388 607 (24-bit binary number).

7.2 Function specifications and restrictions

This section contains the specifications and restrictions of the high-speed counter, pulse output, and PWM output function.

7.2.1 High-speed counter function

For each count input mode, there are certain high-speed counter channels, inputs, and memory areas.

Input numbers

Input mode [1]	No. of phases	Channel no. [2]	Input [3]	Reset input [4]
• Incremental • Decremental	1	0	X0	X2
		1	X1	X2
		2	X3	X5
		3	X4	X5
		4	X6	–
		5	X7	–
• Two-phase • Incremental/decremental • Incremental/decremental control	2	0	X0, X1	X2
		2	X3, X4	X5
		4	X6, X7	–

[1] For details on the different input modes, see p. 176.

[2] Channel 4 and channel 5 are not available for the C10 type.

[3] X4 and X7 can also be used as home inputs of the pulse output function. Set the desired function in the system registers.

[4] Reset input X2 can be set to either channel 0 or channel 1. Reset input X5 can be set to either channel 2 or channel 3.

Performance

No. of phases	Minimum input pulse width [1]	No. of channels	Maximum counting speed [2]
1	10μs	5	50kHz
2	25μs	1	15kHz
		2	15kHz (×2 channels)
		3	10kHz (×3 channels)

[1] For information on the minimum input pulse width, see p. 178.

[2] The maximum counting speed may be lower than the values indicated in the table when the pulse output speed is changed, or when a cam control, target value match on/off, or other interrupt programs are executed simultaneously.

Control flags and memory areas

The high-speed counter operating status, counting values, and control code are stored in special internal flags and special data registers. The control code contains the counter settings. To access special data registers and special internal flags, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. See "Instructions and system variables" on p. 179.

Related instructions

- F165_HighSpeedCounter_Cam: Cam control
- F166_HighSpeedCounter_Set or Hsc_TargetValueMatch_Set: Target value match ON
- F167_HighSpeedCounter_Reset or Hsc_TargetValueMatch_Reset**:** Target value match OFF
- F178_HighSpeedCounter_Measure: Input pulse measurement

7.2.2 Pulse output function

For each pulse output mode and position control mode there are certain designated high-speed counter channels, inputs and outputs.

Note

The pulse output function is only available with the transistor output type.

Input/output numbers

Channel no.			CW pulse output	CCW pulse output	Deviation counter clear output [1]	Home input [3]	Position control trigger input [4]	Near home input
			Pulse output	Direction output				
0			Y0	Y1	Y6 (Y8)	X4	X0	Any[5]
1			Y2	Y3	Y7 (Y9)	X5	X1	
2			Y4	Y5	– (YA)	X6	X2	
3			Y6	Y7	– (YB)	X7	X3	
Linear interpolation control [2]	0	X axis	Y0	Y1	Y6 (Y8)	X4	–	
		Y axis	Y2	Y3	Y7 (Y9)	X5		
	1	X axis	Y4	Y5	– (YA)	X6		
		Y axis	Y6	Y7	– (YB)	X7		

1) The values in parentheses refer to the CPU typesC32, T32, and F32.
For CPU type C16: The deviation counter clear output is not available for channels 2 and 3 and when outputs Y6 and Y7 are used by pulse output channel 3.

2) The home return operation of the interpolation axes should be performed for every channel.

3) X4 and X7 can also be used as high-speed counter inputs. Set the desired function in the system registers.

4) The position control trigger input is used with F171_PulseOutput_Jog_Positioning and PulseOutput_Jog_Positioning0_FB/PulseOutput_Jog_Positioning1_FB. The specified number of pulses is output after the position control trigger input has turned to TRUE. A deceleration is performed before the target value is reached and pulse output stops. The position control trigger can be started by turning a position control trigger input to TRUE or by setting bit 6 of the data register storing the pulse output control code from FALSE to TRUE (e.g. `MOVE(16#140, sys_wHscOrPulseControlCode);`).

5) Any input can be specified in the global variable list. The near home input is enabled/disabled using the pulse output control code. See p. 199.

Performance

No. of channels:	Maximum output frequency:[1)]
4	50kHz
Linear interpolation control	50kHz

[1)] The maximum output frequency may be lower than the values indicated in the table when the pulse output speed is changed, when a target value match on/off instruction, another pulse I/O process or interrupt program is executed simultaneously.

Control flags and memory areas

Counter and pulse output settings as well as elapsed values are stored in special data registers. The pulse output status is stored in special internal flags. To access special data registers and special internal flags, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. See "Instructions and system variables" on p. 196.

Related instructions

- F166_PulseOutput_Set or Pulse_TargetValueMatch_Reset: Target value match ON (pulse output)
- F167_PulseOutput_Reset or Pulse_TargetValueMatch_Set: Target value match OFF (pulse output)
- F171_PulseOutput_Trapezoidal or PulseOutput_Trapezoidal_FB: Trapezoidal control
- F171_PulseOutput_Jog_Positioning or PulseOutput_Jog_Positioning0_FB/PulseOutput_Jog_Positioning1_FB: JOG operation and positioning
- F172_PulseOutput_Jog or PulseOutput_Jog_FB/PulseOutput_Jog_TargetValue_FB: JOG operation
- F174_PulseOutput_DataTable: Data table control
- F175_PulseOutput_Linear or PulseOutput_Linear_FB: Linear interpolation control
- F177_PulseOutput_Home or PulseOutput_Home_FB: Home return

7.2.3 PWM output function

For the pulse-width modulation output function there are two designated channels and outputs.

Note

The PWM output function is only available with the transistor output type.

Output numbers

Channel no.:	PWM output:
0	Y0
1	Y2
2	Y4
3	Y6

Performance

Resolution:	Output frequency (duty ratio:)
1000	6Hz–4.8kHz (0.0–99.9%)

Control flags

The PWM output status is stored in special internal flags. To access special data registers and special internal flags, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header.
See "PWM output function" on p. 216.

Related instructions

F173_PulseOutput_PWM: PWM output

7.2.4 Maximum counting speed and output frequency

The maximum counting speed of the high-speed counter function is determined by the number of channels used and the simultaneous use of the pulse output function. Use the following simplified chart as a guide.

Note

The maximum counting speed may be lower than the values indicated in the table when the pulse output speed is changed, or when a cam control, target value match on/off, or other interrupt programs are executed simultaneously.

Maximum counting speed

No.[1]	Combination of high-speed counter channels									Maximum counting speed (frequency) [kHz][2]									
										No pulse output		Pulse output, 1 channel		Pulse output, 2 channels		Pulse output, 3 channels		Pulse output, 4 channels	
	1-phase						2-phase			1-phase	2-phase	1-phase	2-phase	1-phase	2-phase	1-phase	2-phase	1-phase	2-phase
	Channel																		
	0	1	2	3	4	5	0	2	4										
1	●									50		50		50		50		30	
2	●	●								50		50		50		35		25	
3	●	●	●							50		50		50		30		20	
4	●	●	●	●						50		50		40		30		20	
5	●	●	●	●	●					50		40		35		29		20	
6	●	●	●	●	●	●				50		40		30		24		15	
7							●				15		14		10		10		10
8							●	●			15		10		9		8		8
9							●	●	●		10		10		9		8		8
10			●				●			50	15	50	10	50	10	44	10	30	10
11			●	●			●			50	15	50	10	50	10	40	10	28	10
12			●	●	●		●			50	15	44	10	44	10	30	10	25	10
13			●	●	●	●	●			50	15	35	10	35	10	25	10	20	10
14					●		●	●		50	15	50	9	50	9	35	8	28	8
15					●	●	●	●		50	15	40	9	40	9	30	8	25	8
16	●								●	50	15	50	10	50	10	50	10	40	8
17	●	●							●	50	13	50	10	50	10	45	8	35	7
18	●	●	●						●	50	12	50	9	50	9	40	8	30	7
19	●	●	●	●					●	50	12	50	8	50	8	35	8	30	7
20	●							●	●	50	13	50	10	50	10	50	8	40	8
21	●	●						●	●	50	12	50	9	50	9	45	8	35	7

● Channel is being used

1) The numbers are reference numbers for the specifications continued in the next table.

2) When combined with pulse output function: trapezoidal control, no change in speed (50kHz)

Maximum output frequency

Note

The maximum output frequency may be lower than the values indicated in the table when the pulse output speed is changed, when a target value match on/off instruction, another pulse I/O process or interrupt program is executed simultaneously.

Using channels independently: Even if all channels are used, the maximum output frequency is 50kHz for all.

1-phase				Maximum output frequency [kHz]
Channel 0	Channel 1	Channel 2	Channel 3	
●				50
●	●			50
●	●	●		50
●	●	●	●	50

● Channel is being used

Using linear interpolation control: Even if all channels are used for interpolation, the maximum output frequency is 50kHz for all.

Linear interpolation control		Maximum output frequency [kHz]
Channel 0	Channel 2	
●		50
●	●	50

● Channel is being used

7.3 High-speed counter function

The high-speed counter function counts the input signals and sets the desired output to TRUE or to FALSE when the target value is reached. The high-speed counter function can also be used for cam control and for input pulse measurement.

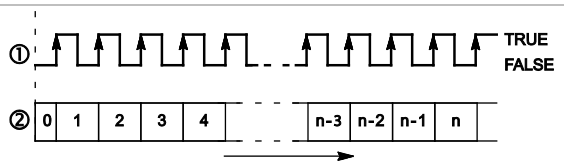
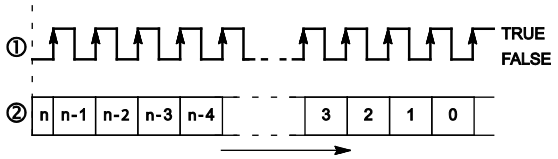
Setting the system registers

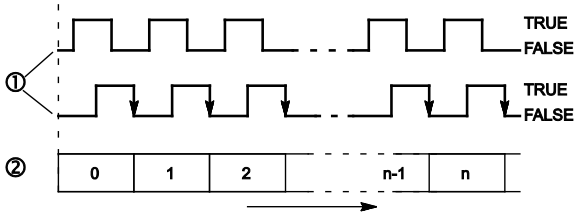
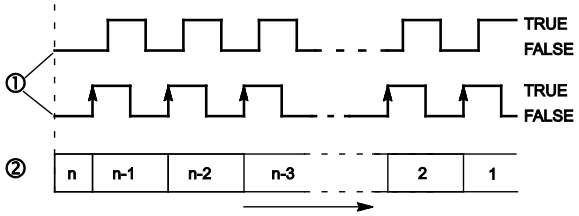
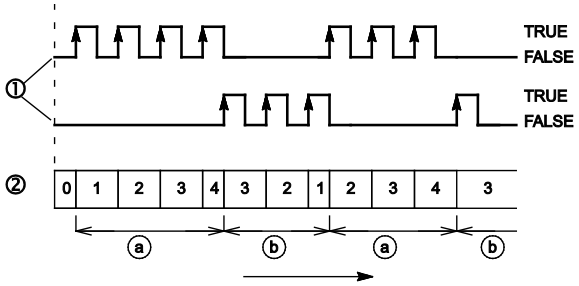
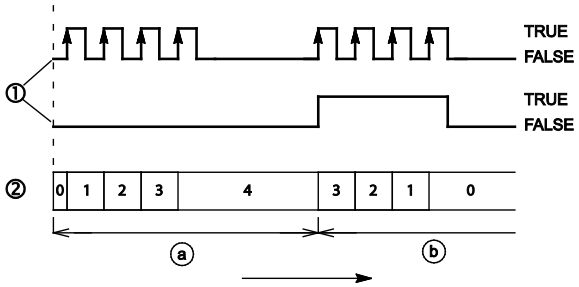
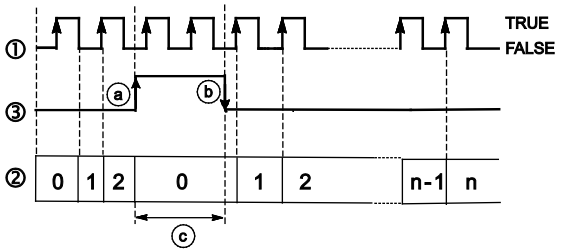
In order to use the high-speed counter function, it is necessary to set the inputs in the system registers.

Procedure

1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click "High-speed counter, pulse-catch input, interrupt input"
4. Select the desired inputs for each channel

7.3.1 Count input modes

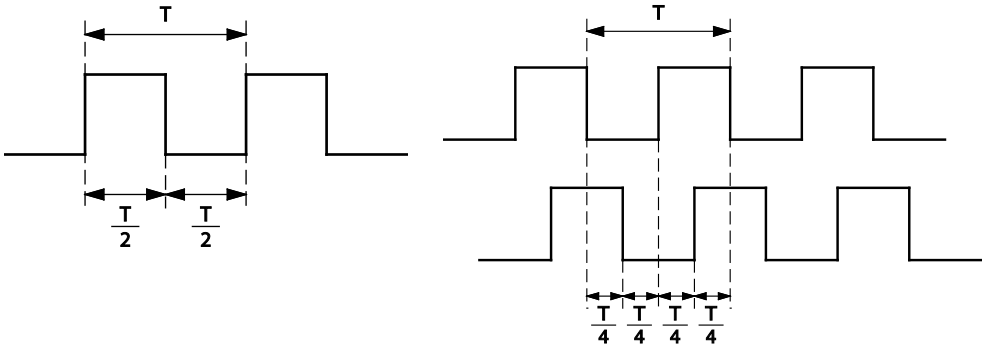
Input mode	Input signals	
Incremental		
	①	High-speed counter input: X0 (X1, X3, X4, X6, X7)
	②	Counter value
Decremental		
	①	High-speed counter input: X0 (X1, X3, X4, X6, X7)
	②	Counter value

Input mode	Input signals
Two-phase	Incremental input 
	Decremental input 
	① High-speed counter input: X0+X1 (X3+X4 or X6+X7)
	② Counter value
Incremental/ decremental	
	① High-speed counter input: X0+X1 (X3+X4 or X6+X7)
	② Counter value
	ⓐ Increasing
	ⓑ Decreasing
Incremental/ decremental control	
	① High-speed counter input: X0+X1 (X3+X4 or X6+X7)
	② Counter value
	ⓐ Increasing
	ⓑ Decreasing
Count for reset (incremental)	
	① High-speed counter input: X0+X1 (X3+X4 or X6+X7)
	② Counter value

Input mode	Input signals	
	③	Reset input: X2 (X5)
	ⓐ	Rising edge: count disabled, elapsed value cleared
	ⓑ	Falling edge: count enabled
	ⓒ	Count prohibited
	The reset at ③ is executed by the interruption at ⓐ (rising edge) and ⓑ (falling edge). The reset input can be enabled/disabled using bit 2 of sys_wHscOrPulseControlCode.	

7.3.2 Minimum input pulse width

For the period T (1/frequency), a minimum input pulse width of T/2 (single-phase input) or T/4 (two-phase input) is required.



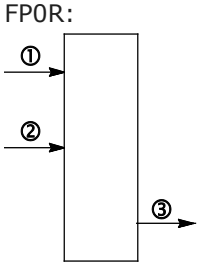
Single-phase input *Two-phase input*

7.3.3 I/O allocation

The inputs and outputs used will differ depending on the channel number being used. (See "Function specifications and restrictions" on p. 169.)

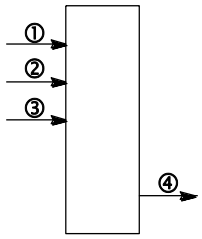
The output to be turned to TRUE or to FALSE can be specified with the instructions F166_HighSpeedCounter_Set or Hsc_TargetValueMatch_Set and F167_HighSpeedCounter_Reset or Hsc_TargetValueMatch_Reset. Outputs can be specified from Y0 to Y7.

Using channel 0 with incremental input and reset input



①	Count input X0
②	Reset input X2
③	TRUE/FALSE output at Yn
Yn	The output which is turned TRUE or FALSE when the target value is reached: Y0–Y7

Using channel 0 with two-phase input and reset input



①	Phase A input X0
②	Phase B input X1
③	Reset input X2
④	TRUE/FALSE output at Yn
Yn	The output which is turned TRUE or FALSE when the target value is reached: Y0–Y7

7.3.4 Instructions and system variables

Control FPWIN Pro offers two concepts for programming with high-speed counter instructions: the original F instructions and the advanced tool instructions. The tool instructions are universal instructions which are supported by all PLC types of the FP series. They offer new and comfortable features including information functions for evaluating status flags and settings, control functions for configuring high-speed counters and pulse outputs, PLC-independent functions and DUTs, as well as variable channel numbers.

Most of the information, which is accessible via information and control functions, is stored in special internal flags and special data registers. These flags and registers can also be accessed using PLC-independent system variables.

The instruction F165_HighSpeedCounter_Cam performs Cam control according to the parameters in the specified DUT.

Use the target value match instructions to turn the desired output to TRUE or to FALSE when the specified target value is reached. To turn the output to TRUE, use F166_HighSpeedCounter_Set or Hsc_TargetValueMatch_Set. To turn the output to FALSE, use F167_HighSpeedCounter_Reset or Hsc_TargetValueMatch_Reset.

The instruction F178_HighSpeedCounter_Measure measures the number of input pulses in a specified counting period and the pulse period.

System variables for memory areas used

Description		System variable	Address
High-speed counter: control flag for channel	0	sys_bIsHscChannel0ControlActive	R9110
	1	sys_bIsHscChannel1ControlActive	R9111
	2	sys_bIsHscChannel2ControlActive	R9112
	3	sys_bIsHscChannel3ControlActive	R9113
	4	sys_bIsHscChannel4ControlActive	R9114
	5	sys_bIsHscChannel5ControlActive	R9115
High-speed counter: elapsed value of channel	0	sys_diHscChannel0ElapsedValue	DDT90300
	1	sys_diHscChannel1ElapsedValue	DDT90304
	2	sys_diHscChannel2ElapsedValue	DDT90308
	3	sys_diHscChannel3ElapsedValue	DDT90312
	4	sys_diHscChannel4ElapsedValue	DDT90316
	5	sys_diHscChannel5ElapsedValue	DDT90320
High-speed counter: target value of channel	0	sys_diHscChannel0ControlTargetValue	DDT90302
	1	sys_diHscChannel1ControlTargetValue	DDT90306
	2	sys_diHscChannel2ControlTargetValue	DDT90310
	3	sys_diHscChannel3ControlTargetValue	DDT90314
	4	sys_diHscChannel4ControlTargetValue	DDT90318
	5	sys_diHscChannel5ControlTargetValue	DDT90322
High-speed counter: control code monitor for channel	0	sys_wHscChannel0ControlCode	DT90370
	1	sys_wHscChannel1ControlCode	DT90371
	2	sys_wHscChannel2ControlCode	DT90372
	3	sys_wHscChannel3ControlCode	DT90373
	4	sys_wHscChannel4ControlCode	DT90374
	5	sys_wHscChannel5ControlCode	DT90375
High-speed counter or pulse output control code		sys_wHscOrPulseControlCode	DT90052

7.3.4.1 Writing the high-speed counter control code

Control codes are used to perform special counter operations.

When programming with F instructions: Use a MOVE instruction to write or read the control code to or from the special data register reserved for this code (DT90052 or DT9052, depending on the PLC type). The special data register where the high-speed counter and pulse output control code are stored can be accessed with the system variable sys_wHscOrPulseControlCode.

When programming with tool instructions: Use universal high-speed counter control instructions which apply to all PLC types to make control code settings. Use the high-speed counter information instructions to monitor control code settings.

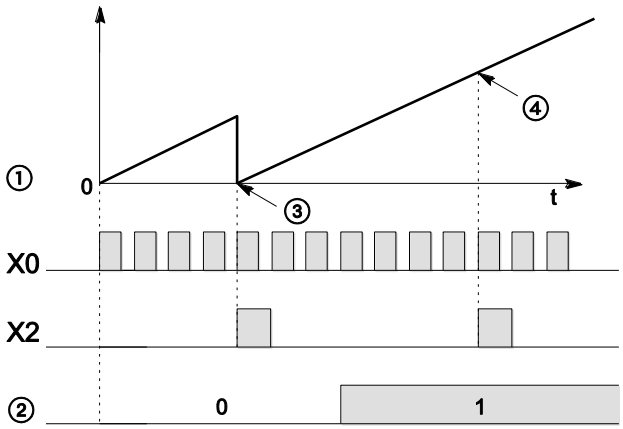
Operations performed by the high-speed counter control code:

- Cancelling high-speed counter instructions (bit 3)
- Enabling/disabling the reset input (hardware reset) of the high-speed counter (bit 2)
- Enabling/disabling counting operations (bit 1)
- Resetting the elapsed value (software reset) of the high-speed counter to 0 (bit 0)

Cancelling high-speed counter instructions (bit 3)

To cancel execution of an instruction, set bit 3 of the data register storing the high-speed counter control code (sys_wHscOrPulseControlCode) to TRUE. The high-speed counter control flag then changes to FALSE. To re-enable execution of the high-speed counter instruction, reset bit 3 to FALSE.

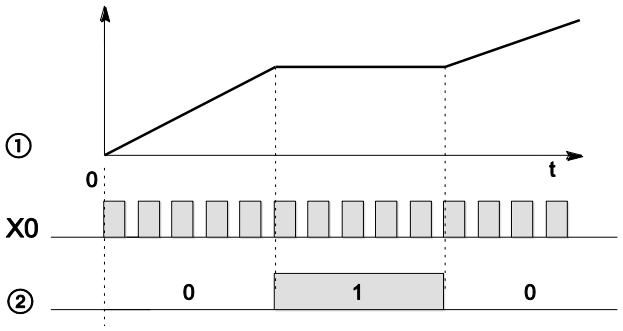
Enabling/disabling the reset input (hardware reset) of the high-speed counter (bit 2)



X0	High-speed counter input
①	Elapsed value
②	Bit 2 of high-speed counter control code (enable/disable reset input)
③	Elapsed value is reset to 0
④	Reset not possible

When bit 2 of the control code is set to TRUE, a hardware reset using the reset input specified in the system registers is not possible. Counting will continue even if the reset input has turned to TRUE. The hardware reset is disabled until bit 2 is reset to 0.

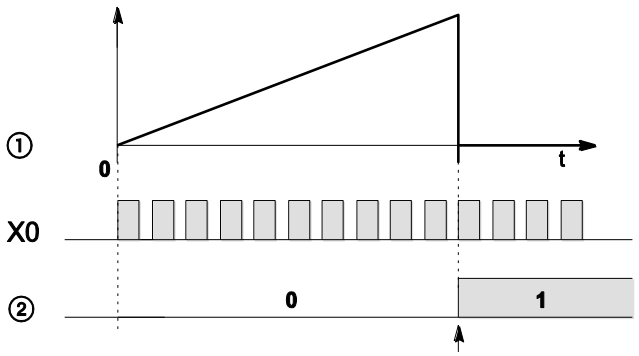
Enabling/disabling counting operations (bit 1)



X0	High-speed counter input
①	Elapsed value
②	Bit 1 of high-speed counter control code (count)

When bit 1 of the control code is set to TRUE, counting is prohibited and the elapsed value keeps its current value. Counting is continued when bit 1 is reset to FALSE.

Resetting the elapsed value (software reset) of the high-speed counter to 0 (bit 0)

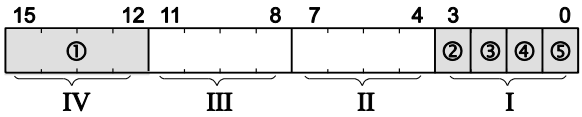


X0	High-speed counter input
①	Elapsed value
②	Bit 0 of high-speed counter control code (software reset)

When bit 0 of the control code is set to TRUE, a software reset is performed and the elapsed value is set to 0. The elapsed value keeps the value 0 until bit 0 is reset to FALSE.

Control code settings

Bits 0–15 of the control code are allocated in groups of four. The bit setting in each group is represented by a hex number (e.g. 0002 0000 0000 1001 = 16#2009).



Group IV	①	Channel number (channel n: 16#n)	
Group III		0 (fixed)	
Group II		0 (fixed)	
Group I	②	Cancel high-speed counter instruction (bit 3)	
		0: continue	1: clear
	③	Reset input (bit 2) (see note)	
		0: enabled	1: disabled
	④	Count (bit 1)	
		0: permit	1: prohibit
	⑤	Reset elapsed value to 0 (bit 0)	
		0: no	1: yes

Example: 16#2009

Group	Value	Description	
IV	2	Channel number: 2	
III	0	(fixed)	
II	0	(fixed)	
I	9	Hex 9 corresponds to binary 1001	
		Cancel high-speed counter instruction: clear (bit 3)	1
		Reset input: enabled (bit 2)	0
		Count: permit (bit 1)	0
		Reset elapsed value to 0: yes (bit 0)	1

Note

Use the reset input setting (bit 2) to disable the reset input allocated in the system registers.

Reference

Please refer to the Control FPWIN Pro online help for programming examples.

7.3.4.2 Writing and reading the elapsed value of the high-speed counter

The elapsed value is stored as a double word in the special data registers.

When programming with F instructions: Access the special data registers using the system variable sys_diHscChannelxElapsedValue (where x=channel number).

When programming with tool instructions: Use universal high-speed counter information and control instructions and pulse information and control instructions which apply to all PLC types to read and write the elapsed value.

System variables for memory areas used:

Description		System variable	Address
High-speed counter: elapsed value of channel	0	sys_diHscChannel0ElapsedValue	DDT90300
	1	sys_diHscChannel1ElapsedValue	DDT90304
	2	sys_diHscChannel2ElapsedValue	DDT90308
	3	sys_diHscChannel3ElapsedValue	DDT90312
	4	sys_diHscChannel4ElapsedValue	DDT90316
	5	sys_diHscChannel5ElapsedValue	DDT90320

Reference

Please refer to the Control FPWIN Pro online help for programming examples.

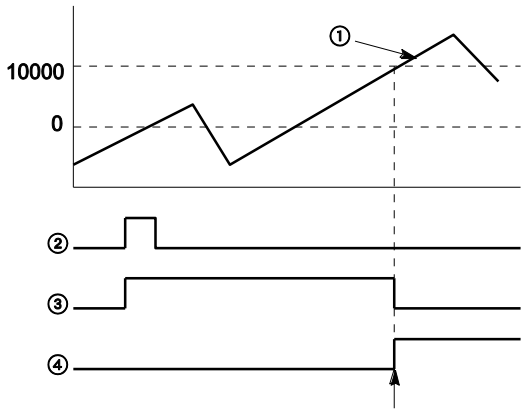
7.3.4.3 Target value match ON

If the elapsed value of the high-speed counter matches the target value, an interrupt process immediately turns the specified output to TRUE.

Tool instruction: Hsc_TargetValueMatch_Set

F instruction: F166_HighSpeedCounter_Set

Characteristics of target value match ON control



10000	Target value
①	Elapsed value of high-speed counter
②	Execution condition
③	High-speed counter control flag
④	PLC output

The PLC output turns to TRUE when the elapsed value matches the target value. In addition, the high-speed counter control flag turns to FALSE and the instruction is deactivated.

Reference

For programming examples, please refer to Example for Hsc_TargetValueMatch_Set or Example for F166_HighSpeedCounter_Set in the Control FPWIN Pro online help.

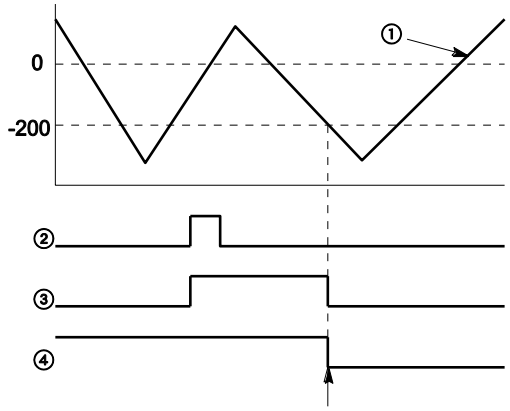
7.3.4.4 Target value match OFF

If the elapsed value of the high-speed counter matches the target value, an interrupt process immediately turns the specified output to FALSE.

Tool instruction: Hsc_TargetValueMatch_Reset

F instruction: F167_HighSpeedCounter_Reset

Characteristics of target value match OFF control



-200	Target value
①	Elapsed value of high-speed counter
②	Execution condition
③	High-speed counter control flag
④	PLC output

The PLC output turns to FALSE when the elapsed value matches the target value. In addition, the high-speed counter control flag turns to FALSE and the instruction is deactivated.

Reference

For programming examples, please refer to Example for Hsc_TargetValueMatch_Reset or Example for F167_HighSpeedCounter_Reset in the Control FPWIN Pro online help.

7.3.4.5 Input pulse measurement

This instruction measures the number of input pulses in a specified counting period and the pulse period.

Tool instruction: Not available

F instruction: F178_HighSpeedCounter_Measure

Characteristics of input pulse measurement

- For input pulse measurement, the channel number, the counting period (1ms–5s) and the number of counting periods (1–5) must be specified. These parameters are used to calculate the average number of input pulses per counting period.
- The unit of pulse period measurement ([μs], [ms] or both) can be specified.

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

7.3.5 Sample programs

The following programming examples demonstrate how to make control code settings and how to use the high-speed counter instructions.

The Control FPWIN Pro projects in LD and ST code can be downloaded from the Panasonic Web site (http://www.panasonic-electric-works.com/eu/downloadcenter.htm).

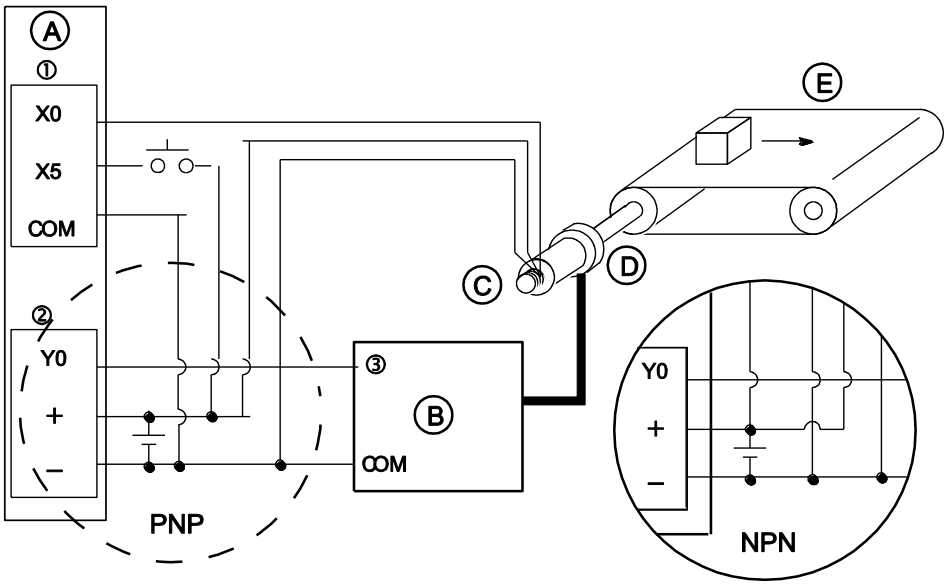
The programming examples for this chapter can be found in pe_63403_0001_sample_high_speed.zip.

These examples can be used with different PLC types. Therefore you have to adapt the PLC type in the Control FPWIN Pro Navigator.

After you have changed the PLC type, a message appears: "Adapt System Registers and Compile Options?" Select [Keep current settings], so that you do not lose the system register settings set in the programming example.

7.3.5.1 Positioning operations with a single-speed inverter

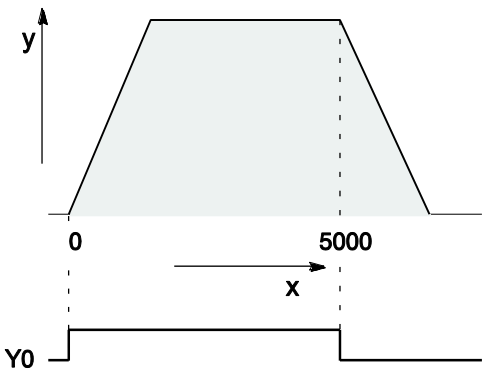
Wiring example



Ⓐ	PLC	①	Input terminal	X0	Encoder input	X5	Operation start
		②	Output terminal	Y0	Inverter operation		
Ⓑ	Inverter	③	Operation stop				
Ⓒ	Encoder						
Ⓓ	Motor						
Ⓔ	Conveyor						

When X5 turns to TRUE, Y0 turns to TRUE and the conveyor starts to move. When the elapsed value (sys_diHscChannel0ElapsedValue) reaches 5000, Y0 turns to FALSE and the conveyor stops.

Operation chart



x	Number of pulses
y	Speed

System register settings

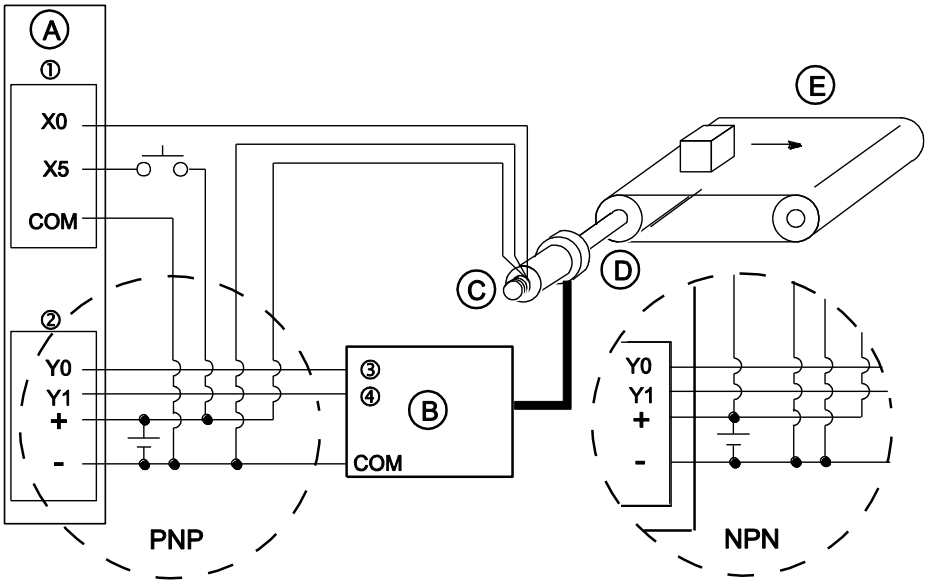
No	Item Name	Data
400	High-speed counter: Channel 0	Incremental input (X0)

Reference

For POU Header and POU Body, please see the programming examples in Panasonic's download area.

7.3.5.2 Positioning operations with a double-speed inverter

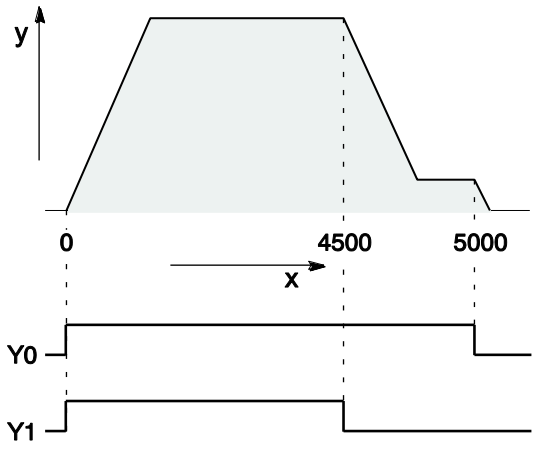
Wiring example



Ⓐ	PLC	①	Input terminal	X0	Encoder input	X5	Operation start
		②	Output terminal	Y0	Inverter operation	Y1	Inverter high-speed
Ⓑ	Inverter	③	Operation stop				
		④	Fast/slow				
Ⓒ	Encoder						
Ⓓ	Motor						
Ⓔ	Conveyor						

When X5 turns to TRUE, Y0 and Y1 turn to TRUE and the conveyor begins to move. When the elapsed value (sys_diHscChannel0ElapsedValue) reaches 4500, Y1 turns to FALSE and the conveyor begins to decelerate. When the elapsed value reaches 5000, Y0 turns to FALSE and the conveyor stops.

Operation chart



x	Number of pulses
y	Speed

System register settings

No	Item Name	Data
400	High-speed counter: Channel 0	Incremental input (X0)

Reference

For POU Header and POU Body, please see the programming examples in Panasonic's download area.

7.4 Pulse output function

Together with a commercially available pulse-string input type motor driver, the pulse output function can be used for positioning control.

Note

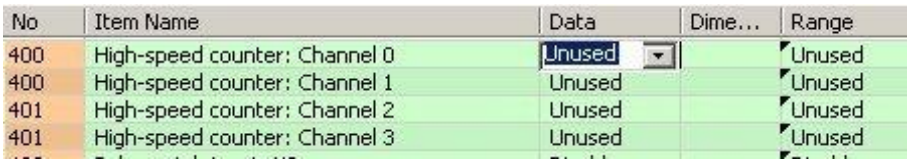
The pulse output function is only available with the transistor output type.

Setting system registers

When using the pulse output function, make sure the high-speed counter function is not allocated to the channel selected for pulse output.

Procedure

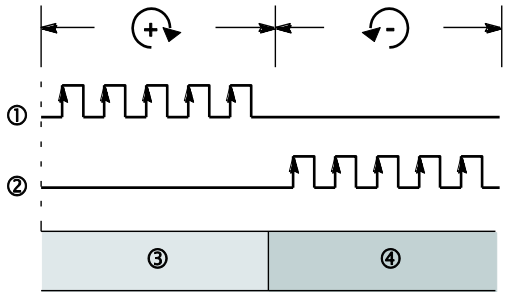
1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click "High-speed counter, pulse-catch input, interrupt input"
4. Set any high-speed counter allocated to a pulse output channel to "Unused"



7.4.1 Pulse output methods and position control modes

The pulse output method and position control mode are specified by means of the variables used with the positioning command.

CW/CCW

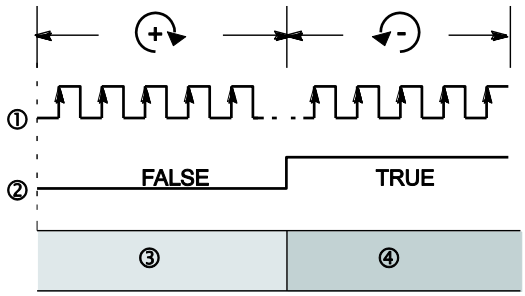


①	CW pulse output: Y0 (Y2)
②	CCW pulse output Y1 (Y3)
③	Incremental counting
④	Decremental counting

Control is carried out using two pulses: a positive or clockwise rotation pulse (CW) and a negative or counterclockwise rotation pulse (CCW pulse).

Pulse/direction

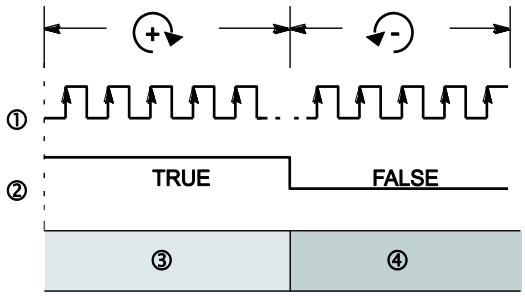
Forward FALSE



①	Pulse output: Y0 (Y2)
②	Direction output Y1 (Y3)
③	Incremental counting
④	Decremental counting

Control is carried out using one pulse output to specify the speed and another to specify the direction of rotation with TRUE/FALSE signals. In this mode, forward rotation is carried out when the rotation direction signal is FALSE.

Forward TRUE



①	Pulse output: Y0 (Y2)
②	Direction output Y1 (Y3)
③	Incremental counting
④	Decremental counting

Control is carried out using one pulse output to specify the speed and another to specify the direction of rotation with TRUE/FALSE signals. In this mode, forward rotation is carried out when the rotation direction signal is TRUE.

Relative value control

The number of pulses set with the target value is output. Positive values result in a positive rotation, negative values in a negative rotation.

Example

With a current position of 5000 and a target value of +1000, 1000 pulses are output from CW to reach the new position at 6000.

Absolute value control

A number of pulses equal to the difference between the set target value and the current value are output. Values greater than the current value result in a positive rotation, values smaller than the current value result in a negative rotation.

Example

With a current position of 5000 and a target value of +1000, 4000 pulses are output from CCW to reach the new position at 1000.

The following outputs are TRUE or FALSE depending on the selected pulse output method and position control mode:

Pulse output method		Pulse output	Target value	
			Positive value/ > current value	Negative value/ < current value
CW/CCW		CW	TRUE	FALSE
		CCW	FALSE	TRUE
Pulse/direction	Forward FALSE	Pulse	TRUE	TRUE
		Direction	FALSE	TRUE
	Forward TRUE	Pulse	TRUE	TRUE
		Direction	TRUE	FALSE
Count mode			Incremental counting	Decremental counting

Home return

After a drive system has been switched on, there is a difference between the internal position value (elapsed value) and the mechanical position of the axis; this difference cannot be predetermined. The internal value must be synchronized with the actual position value of the axis. This is done by means of a home return, during which a position value is registered at a known reference point (home).

During execution of a home return instruction, pulses are continuously output until the home input is enabled. The I/O allocation is determined by the channel used. See "I/O allocation" on p. 195.

To decelerate movement when near the home position, designate a near home input and set bit 4 of the special data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE and back to FALSE again.

The deviation counter clear output can be set to TRUE when home return has been completed.

JOG operation

Pulses are output from the specified channel while the trigger for the jog operation instruction is TRUE. Direction output and output frequency are specified with an instruction.

7.4.2 I/O allocation

The I/O allocation of pulse output terminals, direction output terminal, and home input is determined by the channel used.

For the near home input, the desired contact must be allocated and bit 4 of the special data register storing the pulse output control code (sys_wHscOrPulseControlCode) must be set to TRUE and back to FALSE again.

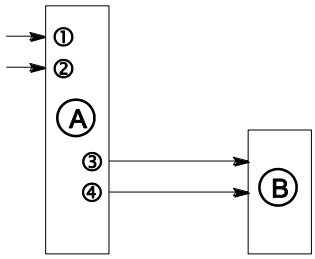
Reference

The input/output numbers are indicated by channel in the specifications.
See "Pulse output function" on p. 171.

Double pulse input driver (CW/CCW pulse output method)

Two output contacts are used as a pulse output for CW/CCW.

Set the control code for the trapezoidal control instruction to CW/CCW.



Using channel		0	2
Ⓐ	PLC		
Ⓑ	Motor driver		
①	Home input	X4	X6
②	Near home input (see note)	e.g. X0	e.g. X1
③	CW pulse output	Y0	Y4
④	CCW pulse output	Y1	Y5

Note

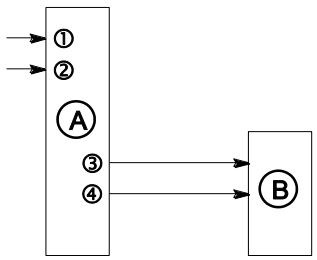
Any input that is not used for other applications can be used as the near home input.

Single pulse input driver (pulse and direction output method)

One output point is used as the pulse output and the other output is used as the direction output.

Set the control code for the trapezoidal control instruction to pulse and direction.

Up to two driver systems can be connected.



Using channel		0	2
Ⓐ	PLC		
Ⓑ	Motor driver		
①	Home input	X2	X6
②	Near home input (see note)	e.g. X0	e.g. X1
③	Pulse output	Y0	Y4
④	Direction output	Y1	Y5

Note

Any input that is not used for other applications can be used as the near home input.

7.4.3 Instructions and system variables

Control FPWIN Pro offers two concepts for programming with pulse output instructions: the original F instructions (e.g. F171_PulseOutput_Trapezoidal) and the advanced tool instructions. The tool instructions are universal instructions which are supported by all PLC types of the FP series. They offer new and comfortable features including information functions for evaluating status flags and settings, control functions for configuring high-speed counters and pulse outputs, PLC-independent functions and DUTs, as well as variable channel numbers.

Most of the information, which is accessible via information and control functions, is stored in special internal flags and special data registers. These flags and registers can also be accessed using PLC-independent system variables.

Use the following instructions to perform various positioning tasks:

Type of control	Instruction
Target value match ON (pulse output) If the elapsed value matches the target value of the selected pulse output channel, the specified output immediately turns to TRUE.	F166_PulseOutput_Set Tool instruction: Pulse_TargetValueMatch_Reset
Target value match OFF (pulse output) If the elapsed value matches the target value of the pulse output channel, the specified output immediately turns to FALSE.	F167_PulseOutput_Reset Tool instruction: Pulse_TargetValueMatch_Reset
Trapezoidal control This instruction automatically performs trapezoidal control according to the parameters in the specified DUT.	F171_PulseOutput_Trapezoidal Tool instruction: PulseOutput_Trapezoidal_FB
Home return This instruction performs a home return according to the parameters in the specified DUT.	F177_PulseOutput_Home Tool instruction: PulseOutput_Home_FB
JOG operation This instruction is used for JOG operation.	F172_PulseOutput_Jog Tool instructions: PulseOutput_Jog_FB PulseOutput_Jog_TargetValue_FB
JOG operation (input controlled) The specified number of pulses is output after the position control trigger input has turned to TRUE. A deceleration is performed before the target value is reached and pulse output stops.	F171_PulseOutput_Jog_Positioning Tool instructions: PulseOutput_Jog_Positioning0_FB PulseOutput_Jog_Positioning1_FB
Data table control This instruction performs rectangular control according to the parameters in the specified DUT with an arbitrary number of different speeds and target values.	F174_PulseOutput_DataTable
Linear interpolation control Pulses are output from two channels in accordance with the parameters in the specified DUT, so that the path to the target position forms a straight line.	F175_PulseOutput_Linear Tool instruction: PulseOutput_Linear_FB

Using the pulse output control flag

The flag is TRUE if a pulse output instruction is being executed. Use this flag to prohibit the simultaneous execution of other pulse output instructions on the specified channel, and to verify completion of the execution.

Note

The status of the high-speed counter control flag or pulse output control flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal flag should be copied to a variable at the beginning of the program.

Channel and pulse output numbers

Channel no.	Interpolation axis [1)]	Pulse output	Pulse output method	
			CW/CCW	Pulse/direction
0	x	Y0	CW	Pulse
		Y1	CCW	Direction
1	y	Y2	CW	Pulse
		Y3	CCW	Direction
2	x	Y4	CW	Pulse
		Y5	CCW	Direction
3	y	Y6	CW	Pulse
		Y7	CCW	Direction

1) For F175_PulseOutput_Linear

Note

For interpolation, channel 0 and 1 or channel 2 and 3 are used as pairs. You may only specify 0 or 2 (for C14T: 0 only).

System variables for memory areas used

Description		System variable	Address
Pulse output: control flag for channel	0	sys_bIsPulseChannel0Active	R9120
	1	sys_bIsPulseChannel1Active	R9121
	2	sys_bIsPulseChannel2Active	R9122
	3	sys_bIsPulseChannel3Active	R9123
Pulse output: elapsed value for channel	0	sys_diPulseChannel0ElapsedValue	DDT90400
	1	sys_diPulseChannel1ElapsedValue	DDT90410
	2	sys_diPulseChannel2ElapsedValue	DDT90420
	3	sys_diPulseChannel3ElapsedValue	DDT90430
Pulse output: target value for channel	0	sys_diPulseChannel0TargetValue	DDT90402
	1	sys_diPulseChannel1TargetValue	DDT90412
	2	sys_diPulseChannel2TargetValue	DDT90422
	3	sys_diPulseChannel3TargetValue	DDT90432
Corrected initial speed for channel[1)]	0	sys_iPulseChannel0CorrectedInitialSpeed	DT90406
	1	sys_iPulseChannel1CorrectedInitialSpeed	DT90416
	2	sys_iPulseChannel2CorrectedInitialSpeed	DT90426
	3	sys_iPulseChannel3CorrectedInitialSpeed	DT90436

Description		System variable	Address
Corrected final speed for channel[1)]	0	sys_iPulseChannel0CorrectedFinalSpeed	DT90407
	1	sys_iPulseChannel1CorrectedFinalSpeed	DT90417
	2	sys_iPulseChannel2CorrectedFinalSpeed	DT90427
	3	sys_iPulseChannel3CorrectedFinalSpeed	DT90437
Acceleration forbidden area starting position for channel[1)]	0	sys_diPulseChannel0AccelerationForbidden-AreaStartingPosition	DDT90408
	1	sys_diPulseChannel1AccelerationForbidden-AreaStartingPosition	DDT90418
	2	sys_diPulseChannel2AccelerationForbidden-AreaStartingPosition	DDT90428
	3	sys_diPulseChannel3AccelerationForbidden-AreaStartingPosition	DDT90438
Pulse output: control code monitor for channel	0	sys_wPulseChannel0ControlCode	DT90380
	1	sys_wPulseChannel1ControlCode	DT90381
	2	sys_wPulseChannel2ControlCode	DT90382
	3	sys_wPulseChannel3ControlCode	DT90383
High-speed counter or pulse output control code		sys_wHscOrPulseControlCode	DT90052

[1)] For F171_PulseOutput_Jog_Positioning, F171_PulseOutput_Trapezoidal, F172_PulseOutput_Jog

7.4.3.1 Writing the pulse output control code

Writing control codes

Control codes are used to perform special counter operations.

When programming with F instructions: Use a MOVE instruction to write or read the control code to or from the special data register reserved for this code (DT90052 or DT9052, depending on the PLC type). The special data register where the high-speed counter and pulse output control code are stored can be accessed with the system variable sys_wHscOrPulseControlCode.

When programming with tool instructions: Use universal pulse control instructions which apply to all PLC types to make control code settings. Use the pulse information instructions to monitor control code settings.

See also:

"Writing the pulse output control code" in the FPWIN Pro online help

Operations performed by the pulse output control code:

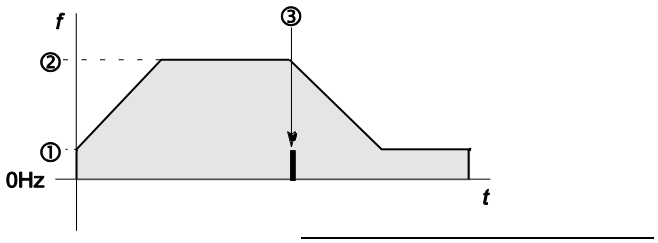
- Setting/resetting near home input
- Continuing/stopping pulse output (forced stop)

- Enabling/disabling counting operations
- Resetting the elapsed value (software reset) of the high-speed counter
- Cancelling high-speed counter and position control instructions (FP0R only)

Setting/resetting near home input

To decelerate movement when near the home position, designate a near home input and set bit 4 of the special data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE and back to FALSE again.

The near home bit is retained. Set this bit to FALSE right after setting it to TRUE to be able to set the near home input a second time during a home return.

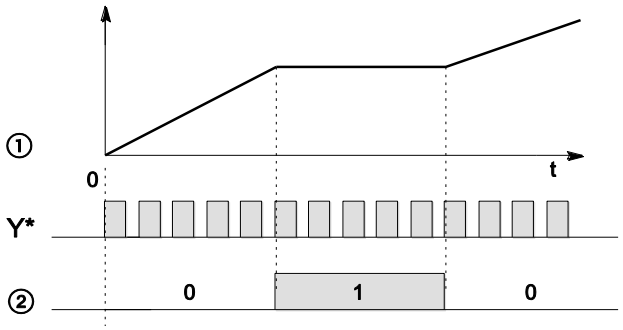


①	Initial and final speed	③	Near home input: TRUE
②	Target speed	④	Home input: TRUE
⑤	Home input is effective at any time.		

Continuing/stopping pulse output (forced stop)

By setting bit 3 of the data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE pulse output is stopped. The possibility of a forced stop should be provided for in every program using pulse output instructions. Reset bit 3 to FALSE to continue pulse output.

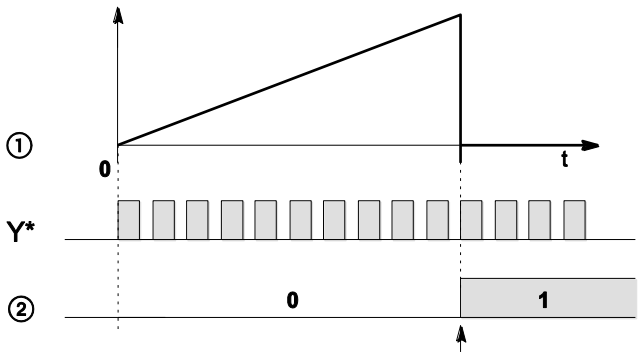
Enabling/disabling counting operations



Y*	Pulse output
①	Elapsed value
②	Bit 1 of pulse output control code (count)

When bit 1 of the control code is set to TRUE, counting is prohibited and the elapsed value keeps its current value. Counting is continued when bit 1 is reset to FALSE.

Resetting the elapsed value (software reset) of the high-speed counter to 0



Y*	Pulse output
①	Elapsed value
②	Bit 0 of pulse output control code (software reset)

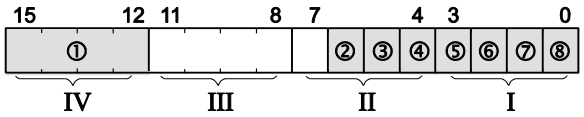
When bit 0 of the control code is set to TRUE, a software reset is performed and the elapsed value is set to 0. The elapsed value keeps the value 0 until bit 0 is reset to FALSE.

Cancelling high-speed counter and position control instructions

To cancel execution of a pulse output instruction, set bit 2 of the data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE. The pulse output control flag will then change to FALSE. To reenable execution of the instruction, reset bit 2 to FALSE.

Control code settings

Bits 0–15 of the control code are allocated in groups of four. The bit setting in each group is represented by a hex number (e.g. 0002 0001 0000 1001 = 16#2109).



Group IV	①	Channel number (channel n: 16#n)	
Group III		1 (fixed)	
Group II	②	Position control start request	
		0: disabled	1: enabled
	③	Decelerated stop request	
		0: disabled	1: enabled
	④	Near home input (bit 4) (see note)	
		0: FALSE	1: TRUE
Group I	⑤	Pulse output (bit 3)	
		0: continue	1: stop
	⑥	Cancel pulse output control (bit 2)	
		0: continue	1: stop
	⑦	Count (bit 1)	
		0: permit	1: prohibit
	⑧	Reset elapsed value to 0 (bit 0)	
		0: no	1: yes

Example: 16#2109

Group	Value	Description	
IV	2	Channel number: 2	
III	1	(fixed)	
II	0	Position control start request: disabled	
		Decelerated stop request: disabled	
		Near home input: FALSE	
I	9	Hex 9 corresponds to binary 1001	
		Pulse output: stop (bit 3)	1
		Cancel pulse output control (bit 2)	0
		Count: permit (bit 1)	0
		Reset elapsed value to 0: yes (bit 0)	1

Note

- Performing a forced stop may cause the elapsed value at the PLC output side to differ from the elapsed value at the motor input side. Therefore, you must execute a home return after pulse output has stopped.
- Setting the near home input is not possible if counting is prohibited or if a software reset is performed.

Reference

Please refer to the Control FPWIN Pro online help for programming examples.

7.4.3.2 Writing and reading the elapsed value of the pulse output

The elapsed value is stored as a double word in the special data registers.

When programming with F instructions: Access the special data registers using the system variable sys_diHscChannelxElapsedValue (where x=channel number).

When programming with tool instructions: Use universal high-speed counter information and control instructions and pulse information and control instructions which apply to all PLC types to read and write the elapsed value.

System variables for memory areas used:

Description		System variable	Address
Pulse output: elapsed value for channel	0	sys_diPulseChannel0ElapsedValue	DDT90400
	1	sys_diPulseChannel1ElapsedValue	DDT90410
	2	sys_diPulseChannel2ElapsedValue	DDT90420
	3	sys_diPulseChannel3ElapsedValue	DDT90430

Reference

Please refer to the Control FPWIN Pro online help for programming examples.

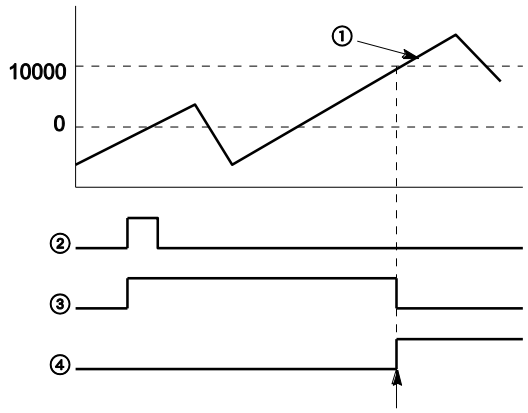
7.4.3.3 Target value match ON (pulse output)

If the elapsed value matches the target value of the selected pulse output channel, the specified output immediately turns to TRUE.

Tool instruction: Pulse_TargetValueMatch_Set

F instruction: F166_PulseOutput_Set

Pulse output characteristics



10000	Target value
①	Elapsed value of pulse output
②	Execution condition
③	"Output control active" flag
④	PLC output

The PLC output turns to TRUE when the elapsed value matches the target value. In addition, the "Output control active" flag turns to FALSE and the instruction is deactivated.

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

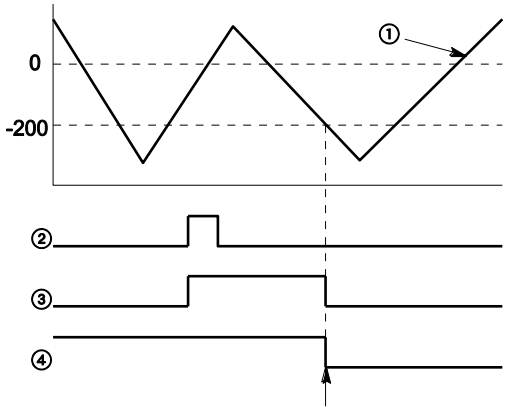
7.4.3.4 Target value match OFF

If the elapsed value matches the target value of the pulse output channel, the specified output immediately turns to FALSE.

Tool instruction: Pulse_TargetValueMatch_Reset

F instruction: F167_PulseOutput_Reset

Pulse output characteristics



10000	Target value
①	Elapsed value of pulse output
②	Execution condition
③	"Output control active" flag
④	PLC output

The PLC output turns to FALSE when the elapsed value matches the target value. In addition, the "Output control active" flag turns to FALSE and the instruction is deactivated.

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

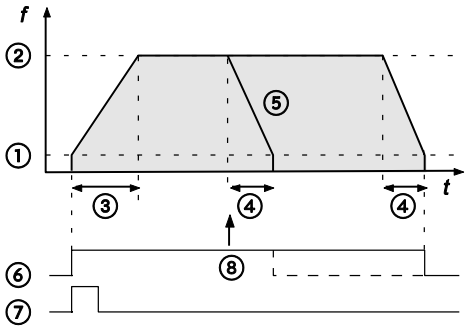
7.4.3.5 Trapezoidal control

This instruction automatically performs trapezoidal control according to the parameters in the specified DUT. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Tool instruction: PulseOutput_Trapezoidal_FB

F instruction: F171_PulseOutput_Trapezoidal

Pulse output characteristics

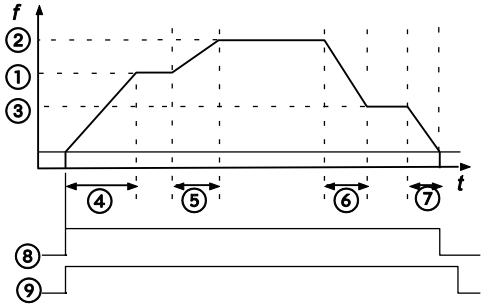


①	Initial and final speed	⑤	Target value
②	Target speed	⑥	Pulse output control flag
③	Acceleration time	⑦	Execution condition
④	Deceleration time	⑧	Decelerated stop request

Type 0: The difference between target speed and initial speed determines the slope of the acceleration ramp. The difference between target speed and final speed determines the slope of the deceleration ramp.

Type 1: The difference between the maximum speed of 50kHz and the final speed determines the slope of the deceleration ramp. The difference between the maximum speed of 50kHz and the initial speed determines the slope of the acceleration ramp.

Changing the target speed during pulse output



Type 1: The speed can be changed within the range of the maximum speed (50kHz).

①	Target speed	⑥	Deceleration
②	1st change of target speed	⑦	Deceleration time
③	2nd change of target speed	⑧	Pulse output control flag
④	Acceleration time	⑨	Execution condition
⑤	Acceleration		

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

7.4.3.6 JOG operation and positioning

The specified number of pulses is output after the position control trigger input has turned to TRUE. A deceleration is performed before the target value is reached and pulse output stops. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Select one of two different operation modes:

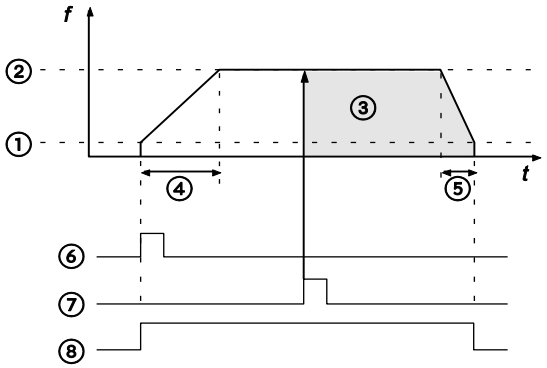
Type 0: The speed can be changed within the range of the specified target speed.

Type 1: The target speed can be changed once when the position control trigger input turns to TRUE.

Tool instruction: PulseOutput_Jog_Positioning0_FB, PulseOutput_Jog_Positioning1_FB

F instruction: F171_PulseOutput_Jog_Positioning

Pulse output characteristics

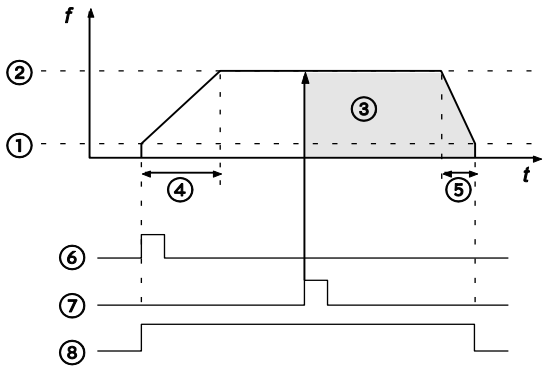


①	Initial and final speed	⑤	Deceleration time
②	Target speed	⑥	Execution condition
③	Target value	⑦	Position control trigger input
④	Acceleration time	⑧	Pulse output control flag

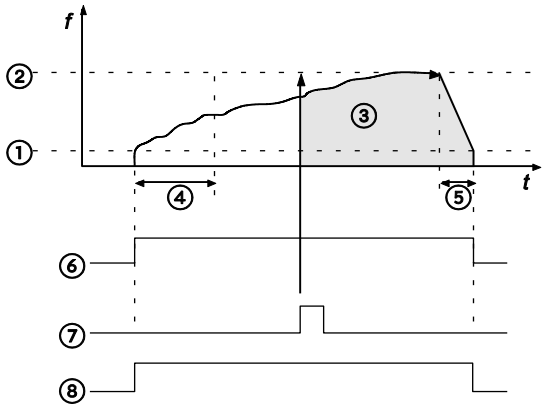
JOG Operation Type 0

The target speed can be changed during pulse output. The speed can be changed within the range of the specified target speed.

Without changing the target speed:



With changing the target speed:

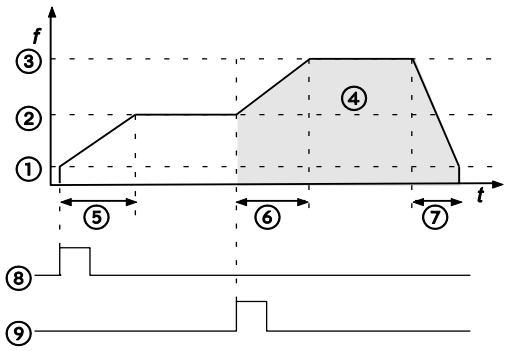


①	Initial and final speed	⑤	Deceleration time
②	Target speed	⑥	Execution condition
③	Target value	⑦	Position control trigger input
④	Acceleration time	⑧	Pulse output control flag

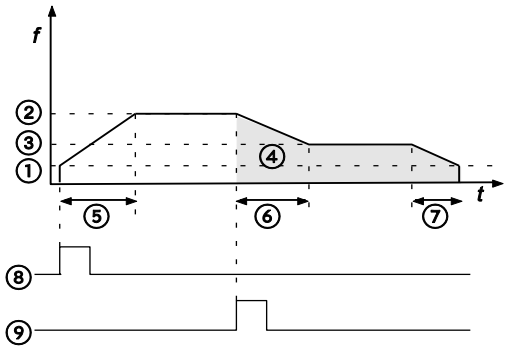
JOG Operation Type 1

The target speed can be changed once when the position control trigger input turns to TRUE.

Target speed 1 < target speed 2:



Target speed 1 > target speed 2:



①	Initial and final speed	⑥	Change time
②	Target speed 1	⑦	Deceleration time
③	Target speed 2	⑧	Execution condition
④	Target value	⑨	Position control trigger input
⑤	Acceleration time		

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

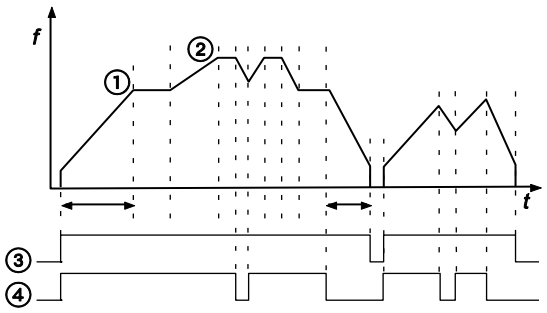
7.4.3.7 JOG operation

This instruction is used for JOG operation. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Tool instruction: PulseOutput_Jog_FB, PulseOutput_Jog_TargetValue_FB

F instruction: F172_PulseOutput_Jog

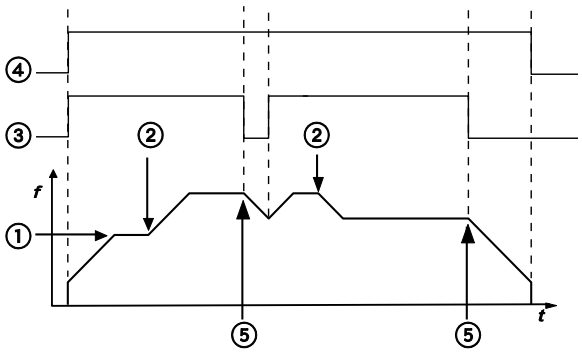
Pulse output characteristics



①	Target speed 1	③	Pulse output control flag
②	Target speed 2	④	Execution condition

Select one of two different operation modes:

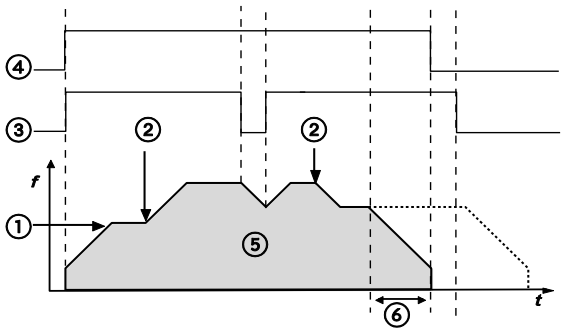
- Mode with no target value (type 0): Pulses are output in accordance with the conditions set in the DUT as long as the execution condition is TRUE. A decelerated stop begins whenever the execution condition is FALSE.



①	Initial and final speed	④	Pulse output control flag
②	Change of target speed	⑤	Decelerated stop
③	Execution condition		

- Target value match stop mode (type 1): Output stops when the target value is reached. Set this mode in the control code, and specify the

target value (an absolute value) in the DUT. A decelerated stop is performed when the target value has been reached. Deceleration is performed within the specified deceleration time.



①	Initial and final speed	④	Pulse output control flag
②	Change of target speed	⑤	Target value
③	Execution condition	⑥	Deceleration time

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

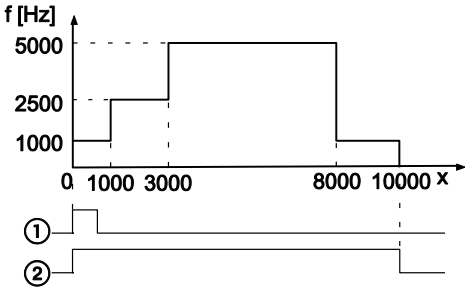
7.4.3.8 Data table control

This instruction performs rectangular control according to the parameters in the specified DUT with an arbitrary number of different speeds and target values. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Tool instruction: not available

F instruction: F174_PulseOutput_DataTable

Pulse output characteristics



x	Elapsed value of pulse output
①	Execution condition
②	Pulse output control flag

- Pulses are output at the specified frequency until the target value is reached. Then the frequency changes to the second frequency value and pulse output continues until the second target value is reached, and so forth.
- Pulse output stops when the last target value is reached.

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

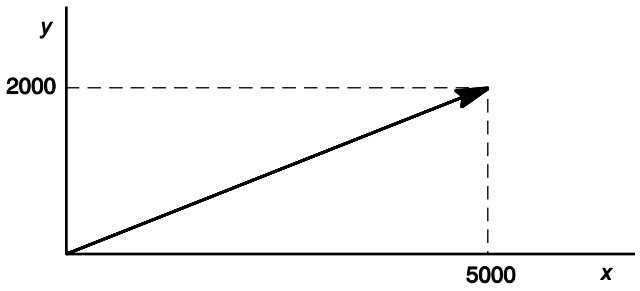
7.4.3.9 Linear interpolation

Pulses are output from two channels in accordance with the parameters in the specified DUT, so that the path to the target position forms a straight line. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Tool instruction: PulseOutput_Linear_FB

F instruction: F175_PulseOutput_Linear

Pulse output characteristics



5000	X-axis target value (channel 0)
2000	Y-axis target value (channel 1)

The two axes are controlled so that a linear path is followed to the target position.

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

7.4.3.10 Home return

This instruction performs a home return according to the parameters in the specified DUT.

After a drive system has been switched on, there is a difference between the internal position value (elapsed value) and the mechanical position of the axis; this difference cannot be predetermined. The internal value must be synchronized with the actual position value of the axis. This is done by means of a home return, during which a position value is registered at a known reference point (home).

Tool instruction: PulseOutput_Home_FB

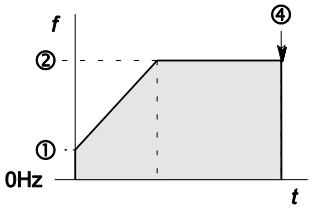
F instruction: F177_PulseOutput_Home

During execution of a home return instruction, pulses are continuously output until the home input is enabled. The I/O allocation is determined by the channel used.

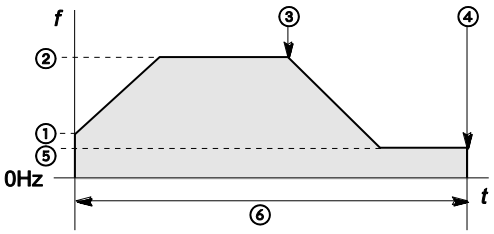
Select one of two different operation modes:

- Type 0: The home input is effective regardless of whether or not there is a near home input, whether deceleration is taking place, or whether deceleration has been completed.

Without near home input:

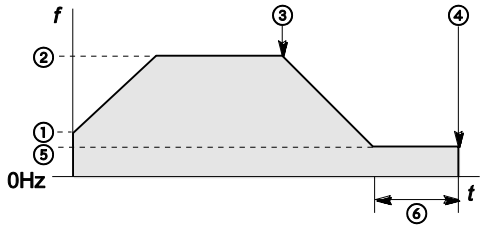


With near home input:



①	Initial speed	④	Home input: TRUE
②	Target speed	⑤	Creep speed
③	Near home input: TRUE	⑥	Home input is effective at any time.

- Type 1: The home input is effective only after deceleration (started by near home input) has been completed.



①	Initial speed	④	Home input: TRUE
②	Target speed	⑤	Creep speed
③	Near home input: TRUE	⑥	Home input is effective only after deceleration

Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

7.5 PWM output function

Use the instruction F173_PulseOutput_PWM. This instruction delivers a pulse width modulated output signal according to the specified DUT.

The PWM output status is stored in special internal flags. To access special data registers and special internal flags, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header.

Reference

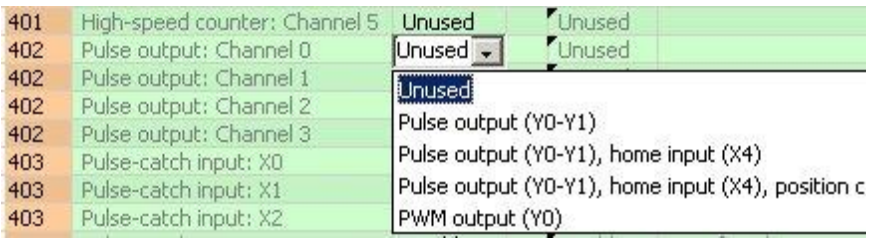
For detailed information, please refer to using system variables in the FPWIN Pro online help.

Setting system registers

When using the PWM output function, specify the desired PWM output in the system registers.

Procedure

1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click "High-speed counter, pulse-catch input, interrupt input"
4. Specify the PWM output for the channel used



Reference

Please refer to the Control FPWIN Pro online help for details and a programming example.

Channel and pulse output numbers

Channel no.	Pulse output
0	Y0
1	Y2
2	Y4
3	Y6

System variables for memory areas used

Description		System variable	Address
Pulse output: control flag for channel	0	sys_bIsPulseChannel0Active	R9120
	1	sys_bIsPulseChannel1Active	R9121
	2	sys_bIsPulseChannel2Active	R9122
	3	sys_bIsPulseChannel3Active	R9123

Chapter 8

Security functions

8.1 Security function types

The following security settings are available:

- PLC program upload protection
- password protection
- security settings for FP Memory Loader

8.2 Security settings in Control FPWIN Pro

If FPWIN Pro is in online mode, **Online** → **Security Settings** opens a dialog that displays the current security settings and enables you to protect your PLC.

The LEDs in the dialog display the PLC's current protection status. To display a tool tip, hold the cursor over the LED for approximately 2s.

Reference

For a detailed description of the options, please refer to Security settings in the FPWIN Pro online help.

8.2.1 Upload protection

When upload protection is enabled, you cannot:

- upload projects or program code to a PC
- upload system registers to a PC

NOTICE

Data can be lost permanently - even if you know the password!

When using this function, be sure to back up your programs! The program on your PLC will not be recoverable, either by a person knowing the password or by customer support.

You can cancel the setting for this function using FPWIN Pro. However, all programs, system registers and password information will be deleted!

If upload protection is enabled, you can edit files on the PLC while online using FPWIN Pro. However, programs will be corrupted if the program in FPWIN Pro and the program on the PLC are not identical.

Note

Even if upload protection has been set, upload to the FP Memory Loader is possible. If you are using FP Memory Loader version 2 or higher, you can enable or disable program upload to the FP Memory Loader or program transfer between two PLCs using the FP Memory Loader. For details, see "FP Memory Loader" on p. 220.

8.2.2 PLC protection (password protection)

You can set a new password with up to 8 characters, or change an existing one.

To access a PLC for which a password has been set, a login is required whenever the power is turned on.

To set a password, you can use:

- the programming tool
- the SYS1 instruction

NOTICE

- Be sure to memorize your password. Without the password, you will not be able to read programs on password-protected PLCs.
- If you have forgotten your password, our support team will not be able to reset it for you.
- If you are not logged in, [Clear Password] will erase not only the password but also the program and parameters stored in the PLC's comment memory.

Reference

For details, please refer to the description of the SYS1 instruction in your Programming Manual or in the FPWIN Pro online help.

8.3 FP Memory Loader

The FP Memory Loader V2.0 or higher (AFP8670/AFP8671) can be used to transfer a program from one PLC to another.

To prevent unauthorized copying of user programs, you must enable upload protection. This function is recommended for users who manage original programs on a PC.

In FPWIN Pro, **Online** → **Security Settings** opens the Security Settings dialog box, which offers two security settings for the FP Memory Loader:

- Upload protection
- Download protection

8.3.1 Upload protection

Upload protection prevents programs from being uploaded to an FP Memory Loader.

Procedure

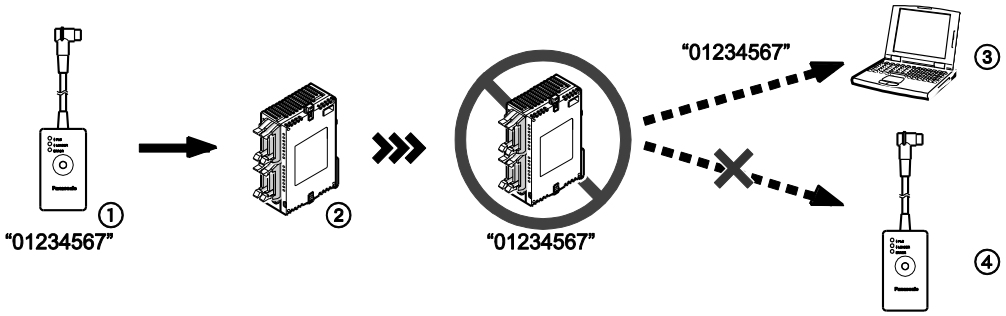
1. **Online** → **Security settings**

   The Security settings dialog box opens.

2. Select "Enable upload protection"
3. Enter the password
4. Choose [Set Protection] or [Change Protection]

   When defining security settings for the first time, choose [Set Protection].
   To change existing security settings, choose [Change Protection].

5. Download program from source PLC to FP Memory Loader
6. Transfer program to target PLC
7. After program download from the FP Memory Loader to the target PLC, this PLC is now upload-protected.



Program upload can be disabled in the Security Settings dialog box (see table below)

①	The FP Memory Loader contains a password- and upload-protected program. Password: 01234567 Upload protection: enabled
②	The security settings are transferred together with the program to the target PLC. The target PLC is now doubly protected.
③	Program upload to a PC requires password entry.
④	Upload to an FP Memory Loader is not possible, even if the source PLC and target PLC are protected by identical passwords ("01234567").

8.3.2 Download protection

Using download protection, you can transfer a program from one PLC to another with the FP Memory Loader, provided the passwords of the two PLCs are identical.

Procedure

1. **Online → Security settings**

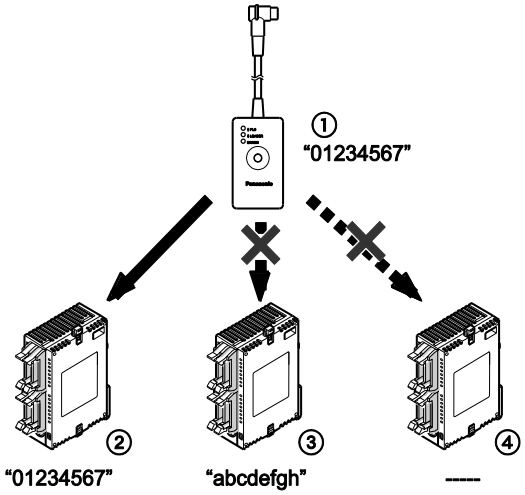
   The Security settings dialog box opens.

2. Select "Allow download to PLC only if password in the PLC is the same"
3. Enter the password
4. Choose [Set Protection] or [Change Protection]

   When defining security settings for the first time, choose [Set Protection].
   To change existing security settings, choose [Change Protection].

5. Download program from source PLC to FP Memory Loader

6. Transfer program to target PLC



Programs can only be downloaded to PLCs that are protected by identical passwords (see table below)

①	The FP Memory Loader contains a password protected program. Password: 01234567
②	Download is possible only if the target PLC is protected by the same password ("01234587").
③	Download to a target PLC that is protected by a different password ("abcdefgh") is not possible.
④	Download to a target PLC that is not password-protected (-----) is not possible.

NOTICE

During program download from the FP Memory Loader to a target PLC, the password set on the source PLC may be changed.

The password on the source PLC may be changed under the following conditions:

Security setting on FP Memory Loader	Password setting on target PLC after download
No password set	Password will be cleared
8-digit password set, "Allow download to PLC only if password in the PLC is the same" setting disabled	Password will be overwritten with new 8-digit password
8-digit password set, "Allow download to PLC only if password in the PLC is the same" setting enabled	Password will not be changed (no download possible)

Chapter 9

Other functions

9.1 F-ROM backup (P13_EPWT)

Data registers of 32765 words can be written to the built-in F-ROM of the FP0R CPU using the instruction P13_EPWT.

Writing can be performed up to 10000 times. After that, the correct operation cannot be guaranteed.

If the power supply turns off while the instruction **P13_EPWT** is being executed or during online editing, data in the hold area may be lost.

Reference

For details, please refer to the online help of Control FPWIN Pro.

9.2 Sampling trace

Using the sampling trace function, the current contact conditions and/or the variable values can be displayed on a time axis. After data recording in the PLC has been completed, the data is loaded into FPWIN Pro. Sampling parameters such as the sampling time and trigger conditions can be set in FPWIN Pro.

A maximum of 16 Boolean variables and three 16-bit variables can be traced per sampling.

Reference

For details, please refer to the online help of Control FPWIN Pro.

9.3 Input time constants

You can specify input time constants in order to negate the effects of noise or bouncing, e.g. for a switching device.

To set the time constants, use the system registers or the instruction F182_FILTER.

Time constant settings are invalid if the input is used as a high-speed counter, pulse catch, or interrupt input.

Reference

For details, please refer to the online help of Control FPWIN Pro.

Time constants can be set for the following inputs, depending on the CPU type:

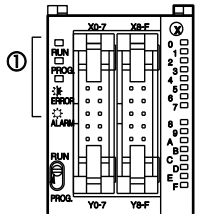
Input	CPU type	
	C10/C14/C16	C32/T32/F32
X0–X3	●	●
X4–X7	●	●
X8–XB	–	●
XC–XF	–	●

Chapter 10

Troubleshooting

10.1 LED display for status condition

When an error occurs, the status of the operation status LEDs on the CPU changes as shown in the table below.



① Status indicator LEDs

Status indicator LEDs on CPU

	LED status			Description	Operation status
	RUN	PROG.	ERROR/ ALARM		
Normal condition	On	Off	Off	Normal operation	Continue
	Off	On	Off	PROG mode	Stop
	Flashes	Flashes	Off	Forcing on/off in RUN mode	Continue
Abnormal condition	On	Off	Flashes	A self-diagnostic error has occurred	Continue
	Off	On	Flashes	A self-diagnostic error has occurred	Stop
	Varies	Varies	On	System watchdog timeout has occurred	Stop

10.2 Operation on error

The CPU has a self-diagnostic function which identifies errors and stops operation if necessary. For some errors, the user may select whether operation shall continue or stop when the error occurs.

Procedure

1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. Double-click “Act on Error”

   Select the desired setting for each type of error.

Example

Operation is to continue even though a calculation error has occurred: Set the system register "Operation error" to "Continue". Operation errors will be handled as an error, but operation will continue.

10.3 ERROR/ALARM LED is flashing

Check the error code using the programming software.

Procedure

- In online mode: **Monitor** → **PLC status** or 

  The error code is displayed in the "Self-diagnostic error" section.

For self-diagnostic errors 20 and higher other than a syntax error

There are three ways to clear the error:

- Choose [Clear] in the "PLC status" dialog while in PROG mode
- Turn the power supply off/on while in PROG mode (this clears all of the contents of the operation memory except hold type data)
- Execute the self-diagnostic error set instruction F148_ERR

Note

- If the operation mode selector has been set to RUN, the error is cleared and at the same time operation is enabled. However, the error continues to be displayed unless the cause of the error has been eliminated.
- When an operation error (error code 45) occurs, the address at which the error occurred is stored in special data registers DT90017 (sys_iOperationErrorStepHold) and DT90018 (sys_iOperationErrorNonHold). If this happens, monitor the address at which the error occurred before cancelling the error.

10.4 ERROR/ALARM LED is ON

If the ERROR/ALARM LED is on, the system watchdog timer has been activated and the operation of the PLC has been stopped. There are two ways to remedy the problem:

- Set the mode selector of the PLC from RUN to PROG mode and turn the power off and then on.
  - If the ERROR/ALARM LED turns on again, there is probably an abnormality in the CPU. Please contact your dealer.
  - Check if the ERROR/ALARM LED is flashing. See "ERROR/ALARM LED is flashing" on p. 226.
- Set the mode selector from PROG to RUN mode. If the ERROR/ALARM LED turns on, the program execution time is too long.
  - Check if instructions such as JP or LOOP are programmed in such a way that a scan can never finish.
  - Make sure that interrupt instructions are executed in succession.

10.5 All LEDs are OFF

If all LEDs are OFF, try the following:

- Check the power supply wiring.
- Check if the power supplied to the CPU is in the range of the rating. Be sure to check the fluctuation in the power supply.
- Disconnect the power supply wiring to the other devices if the power supplied to the CPU is shared with them.
  - If the LEDs on the CPU turn on at this moment, increase the capacity of the power supply or prepare another power supply for other devices.
  - Please contact your dealer for further information.

10.6 Diagnosing output malfunction

If the outputs do not function correctly, both software (e.g. program, I/O allocation) and hardware (e.g. wiring, power supply) may be responsible. Check the output side first and then the input side.

If the output status LEDs are ON:

- Check the wiring of the loads.
- Check if the power is properly supplied to the loads.
  - If the power is properly supplied to the load, there is probably an abnormality in the load.
  - If the power is not supplied to the load, there is probably an abnormality with the outputs.

If the output status LEDs are OFF:

- Monitor the output condition using Control FPWIN Pro.
  - If the output monitored is TRUE, there is probably a duplicate output error.
- Set the output to TRUE by force using Control FPWIN Pro.
  - If the output status LED is turned ON, you must check the input side.
  - If the output status LED remains OFF, there is probably an abnormality with the outputs.

If the input status LEDs are OFF:

- Check the wiring of the input devices.
- Check that the power is properly supplied to the input terminals.
  - If the power is properly supplied to the input terminal, there is probably an abnormality with the inputs.
  - If the power is not supplied to the input terminal, there is probably an abnormality in the input device or input power supply. Check the input device and input power supply.

If the input status LEDs are ON:

Monitor the input condition using Control FPWIN Pro.

- If the input monitored is FALSE, there is probably an abnormality with the inputs.
- If the input monitored is TRUE, check the leakage current at the input devices (e.g. two-wire type sensor) and check the program again, referring to the following:
  - Check for duplicate output errors and for outputs having been rewritten by high-level instructions.
  - Check the program flow when instructions such as MC or JP are used.
  - Check if the I/O map agrees with the actual mounting status.

10.7 Password protection error message

If a protection error message appears, a password has been set.

To access a PLC for which a password has been set, a login is required whenever the power is turned on.

Procedure

1. **Online** → **Security settings**
2. Enter your password under "PLC Access"
3. Choose [Login]

NOTICE

If you are not logged in, [Clear Password] will erase not only the password but also the program and parameters stored in the PLC's comment memory.

10.8 PROG mode does not change to RUN

If PROG mode does not change to RUN, a syntax error or a self-diagnostic error that caused operation to stop has occurred.

- Check to see if the ERROR/ALARM LED is flashing. See "ERROR/ALARM LED is flashing" on p. 226.
- Locate the syntax error by executing **Monitor** → **PLC status** or .

Chapter 11

Appendix

11.1 Specifications

11.1.1 General specifications

Item		Description		
Rated operating voltage		24V DC		
Operating voltage range		20.4–28.8V DC		
Momentary power off time	C10, C14, C16	5ms at 20.4V, 10ms at 21.6V		
	C32, T32, F32	10ms at 20.4V		
Fuse		Built-in (cannot be replaced)		
Ambient temperature		0–+55°C		
Storage temperature		-40–+70°C (T32: -20–+70°C)		
Ambient humidity		10%–95% RH (at 25°C, non-condensing)		
Storage humidity		10%–95% RH (at 25°C, non-condensing)		
Breakdown voltage (Cutoff current: 5mA)			Transistor types	Relay types
		Input terminals ↔ Output terminals	500V AC for 1min	1500V AC for 1min
		Output terminals ↔ Output terminals (of different COM terminals)	–	1500V AC for 1min
		Input terminals ↔ Power supply terminal/Function earth	500V AC for 1min	500V AC for 1min
		Output terminals ↔ Power supply terminal/Function earth	500V AC for 1min	1500V AC for 1min
		Function earth ↔ Power supply terminal/	500V AC for 1min	500V AC for 1min
Insulation resistance (measured with a 500V DC megger)		Input terminals ↔ Output terminals	Min. 100MΩ	Min. 100MΩ
		Output terminals ↔ Output terminals (of different COM terminals)	–	Min. 100MΩ
		Input terminals ↔ Power supply terminal/Function earth	Min. 100MΩ	Min. 100MΩ
		Output terminals ↔ Power supply terminal/Function earth	Min. 100MΩ	Min. 100MΩ
		Function earth ↔ Power supply terminal/	Min. 100MΩ	Min. 100MΩ
Vibration resistance		5–9Hz, 1 sweep/min, amplitude of 3.5mm 9–150Hz, 1 sweep/min, constant acceleration of 9.3m/s^2, 10min on 3 axes (in X, Y, and Z direction)		
Shock resistance		147m/s^2, 4 times on 3 axes (in X, Y, and Z direction)		

Item	Description
Noise immunity (Power supply terminal)	1000Vp-p, with pulse widths 50ns and 1μs (based on in-house measurements)
Operation conditions	Free from corrosive gases and excessive dust
Overvoltage category	II
Pollution degree	2
Weight	C10: 100g, C14: 105g, C16: 85g, C32: 115g, T32: 115g, F32: 120g

11.1.2 Performance specifications

Item		C10, C14, C16	C32, T32, F32
Programming method/control method		Relay symbol/cyclic operation	
Program memory	Internal memory	F-ROM	
	Program capacity (steps)	16000	32000
	Online edit mode	Available (entire program)	
	Security function	Password protection (8-digit), upload protection	
Comment memory	Memory capacity	328kbyte	
	Online edit mode	Available (project information)	
I/O refresh time		≤0.2ms With expansion units: ≤0.2ms + (1 × no. of expansion units)ms	
Operation speed	≤3000 steps	Basic instructions: 0.08μs, timer instruction: 2.2μs High-level instructions: 0.32μs (MV instruction)	
	>3000 steps	Basic instructions: 0.58μs, timer instruction: 3.66μs High-level instructions: 1.62μs (MV instruction)	
Basic instructions		Approx. 110	
High-level instructions		Approx. 210	
Operation memory: bits	Inputs (X)	1760	
	Outputs (Y)	1760	
	Internal flags (R)	4096	
	Special internal flags (R)	224	
	Timer flags/Counter flags (T/C)	1024 Factory setting timers: 1008 points (T0–T1007) Factory setting counters: 16 points (C1008–C1023) Timer: 1–32767 (in units of 1ms, 10ms, 100ms, or 1s). Counter: 1–32767	
	Link flags (L)	2048	

Item		C10, C14, C16	C32, T32, F32
Operation memory: words	Data registers (DT)	12315 words	32765 words
	Special data registers (DT)	440 words (DT90000–DT90443)	
	Link registers (LD)	256 words	
	Index registers (I)	14 words (I0–ID)	
Differential points		Unlimited	
Master control flags (MCR)		256	
Number of labels (JP and LOOP)		256	
Number of SFC steps		1000	
Number of subroutines		500	
Sampling trace		300 samples	1000 samples
		Per scan or per time interval Max. 16 Boolean variables and 3 16-bit variables per sampling	
High-speed counter [1]		1-phase: 6 channels (max. 50kHz) 2-phase: 3 channels (max. 15kHz)	
Pulse output (not available for C10, C14)[1] [2]		4 channels (max. 50kHz)	
PWM output (not available for C10, C14)[1] [2]		4 channels (max. 4.8kHz)	
Pulse catch inputs		8 (including high-speed counter and interrupt input)	
Number of interrupt programs		8 external inputs (C10: 6) 1 periodical interrupt 4 target value match interrupts	
Periodical interrupt		0.5ms–1.5s (unit: 0.5ms), 10ms–30s (unit: 10ms)	
Constant scan time		0.5ms–600ms (unit: 0.5ms)	
F-ROM backup [3]	Using instructions F12 and P13	All areas (32765 words)	
	Automatically when power is cut off	Counter flags: 16 (C1008–C1023) Internal flags: 128 (R2480–R255F) Data registers: 315 words	
		DT12000–DT12314	DT32450–DT32764
RAM backup (T32 and F32 only) [4]		T32: All areas (built-in backup battery)[5] F32: All areas	
Clock/calendar function [6]		Available for T32 only.	
Communication ports		TOOL port, USB port, COM port	
Self-diagnostic function		E. g. watchdog timer, program syntax check (Watchdog timer: approx. 690ms)	

[1] These are the specifications when the rated input voltage is 24V DC at 25°C. The frequency will decrease depending on voltage, temperature or usage condition.

[2] A total of 4 channels is available for pulse output and PWM output.
Pulse output can be specified up to 50kHz. PWM output can be specified up to 4.8kHz. A maximum error on the pulse width of 40μs may occur for the setting value depending on voltage, temperature or operating condition.

[3] Writing is possible up to 10000 times.

[4] All memory areas including timers/counters, internal flags, link flags, link registers and data registers can be backed up. Areas to be held and not to be held can be specified in the system registers.

[5] The built-in backup battery is not charged when the unit is shipped. Charge the battery sufficiently before use.

The battery does not have an alarm function when it is running low. If the battery is empty the data values in the hold area will become indefinite during power-off. They are cleared to 0 the next time the power is turned on. We recommend adding a program for checking if the data is set to 0 when the power is turned on the next time.

[6] Precision: at 0°C: error <104s/month; at 25°C: error <51s/month; at 55°C: error <155s/month

11.1.3 Communication specifications

TOOL port

Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Slave Modem connection Program controlled (in RUN mode only)

USB port

Item	Description
Standard (baud rate)	USB 2.0 Full Speed
Communication mode	MEWTOCOL-COM Slave

COM port (RS232C)

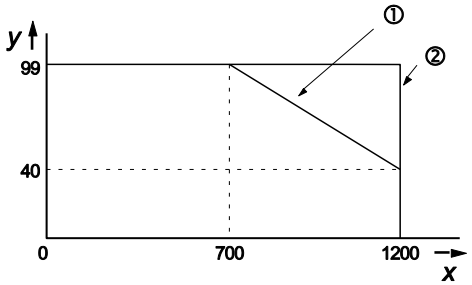
Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Master/Slave Modem connection Program controlled Modbus RTU Master/Slave PLC Link

COM port (RS485)

Item		Description
Interface		RS485
Connection mode		1:N
Transmission distance		1200m [1)2)]
Baud rate		19200, 115200bit/s [2)3)]
Communication method		2-wire, half-duplex
Synchronous method		Start stop synchronous system
Transmission line		Shielded twisted-pair cable or VCTF
Transmission code	MEWTOCOL-COM	ASCII
	Program controlled	ASCII, Binary
	Modbus RTU	Binary
Communication format (set in system registers) [4)]		Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
No. of connected stations [2) 5)]		≤99 (≤32 with C-NET adapter)
Communication mode		MEWTOCOL-COM Master/Slave Modem connection Program controlled Modbus RTU Master/Slave PLC Link

[1)] The number of stations, transmission distance, and baud rate may vary depending on the connected RS485 device.

2) The values for the transmission distance, baud rate and number of stations should be within the values noted in the following graph.



x Transmission distance [m]

y Number of stations

① For a baud rate of 115200bit/s

② For a baud rate of 19200bit/s

3) Set the baud rate in the system registers and set the DIP switch on the bottom of the unit to the same setting. When a C-NET adapter is connected to the RS485 interface, you can only specify a baud rate of 19200bit/s.

4) The start and end code can only be used in program controlled communication.

5) Station numbers should be registered via the system registers.

Note

If the potential difference between the power supplies of RS485 devices exceeds 4V, communication may fail because the RS485 port is non-isolated. The large potential difference will damage the connected devices.

Default settings

Port	Baud rate	Data length	Parity	Stop bit
TOOL port	9600bit/s	8 bits	Odd	1 bit
COM port (RS232C)	9600bit/s	8 bits	Odd	1 bit
COM port (RS485)	115200bit/s	8 bits	Odd	1 bit

11.1.4 Power supply specifications

Item		FP-PS24-024E	FP-PS24-060E	FP-PS24-120E
Primary side	Rated input voltage	100–240V AC/DC, 50–60Hz		
	Operating voltage range	85–264V AC, 47–63Hz (DC 100–375V)		
	Input current	Fulfills the requirements of EN 61000-3-2 (limits for harmonic current emissions)		
	Fuse	Built-in (cannot be replaced), T4AH/250V		
Secondary side	Rated output voltage	24V DC		
	Accuracy of output voltage	±1% over the complete load and input voltage range		
	Adjustable range (with potentiometer)	23V–29V		
	Max. output capacity	1A continuous at 24V	2.5A continuous at 24V	5.0A continuous at 24V
	Min. output capacity	0A		
	Current limiting (typ.)	2A continuous, 2A dynamic	2.7A continuous, 5A dynamic	5.3 continuous, 9.5A dynamic
	Ripple voltage	40mVSS measured at 20MHz, 50Ω terminated		
	Overvoltage protection	Yes, U1<35V		
	Overcurrent protection function	In case of overload the output voltage will be reduced down to approx. 17V. Below this the power supply will enter hicc-up mode to protect power supply and load from over temperature or burning.		
Lifetime of capacitors		Min. 50000h at an airflow temperature of Tu=50°C		

11.1.5 Current consumption

Type of unit		CPU [1]	Expansion unit [2]	Input circuit [3]	Output circuit [4]
FP0R CPU	FP0R-C10	≤100mA	–	≤15.9mA	–
	FP0R-C14	≤120mA	–	≤21.1mA	–
	FP0R-C16	≤70mA	–		≤20mA
	FP0R-C32 FP0R-T32 FP0R-F32	≤90mA	–	≤42.2mA	≤40mA
FP0/FP0R I/O expansion unit	FP0R-E8X	≤10mA	--	≤37.6mA	–
	FP0R-E8R		≤50mA	≤18.8mA	–
	FP0R-E8YR		≤100mA	–	–
	FP0R-E8YT/P	≤15mA	–	–	≤26mA
	FP0R-E16X	≤10mA	–	≤75.2mA	–
	FP0R-E16R	≤20mA	≤100mA	≤37.6mA	–
	FP0R-E16T/P		–	≤37.6mA	≤26mA
	FP0R-E16YT/P	≤25mA	–	–	≤52mA
	FP0R-E32T/P	≤35mA	–	≤75.2mA	
	FP0R-E32RS	≤40mA	≤200mA	≤69mA	–
FP0 analog unit	FP0-A04V	≤20mA ≤20mA	≤100mA	–	–
	FP0-A04I		≤130mA	–	–
	FP0-A21		≤100mA	–	–
	FP0-A80		≤60mA	–	–
	FP0-TC4/TC8/RTD6	≤25mA	–	–	–
FP0 intelligent unit	FP0-IOL	≤30mA	≤40mA	–	–
	FP0-CCLS	≤40mA		–	–
	FP0-DPS2	≤30mA	≤100mA	–	–
Communication cassette	FPG-COM1 FPG-COM2	≤20mA	–	–	–
	FPG-COM3 FPG-COM4	≤25mA	–	–	–
GT series touch panel (5V type)	AIGT0030B1 AIGT0030H1 AIGT0230B1 AIGT0230H1	≤80mA	–	–	–
C-NET adapter S2	AFP15402	≤50mA	–	–	–

[1] The current consumed by the CPU power supply connector. If expansion units or intelligent units are added, the current is increased by the value indicated in the table.

[2] The current consumed by the expansion unit power supply connector. If a unit is not listed in the table, it means that it has no power supply connector.

[3] The current consumed by the input circuits of the various units. The value indicates the current that flows into the input circuit.

[4] The current consumed by the output circuits of the various units. The value indicates the current used to drive the output circuits. The value does not include the load current value.

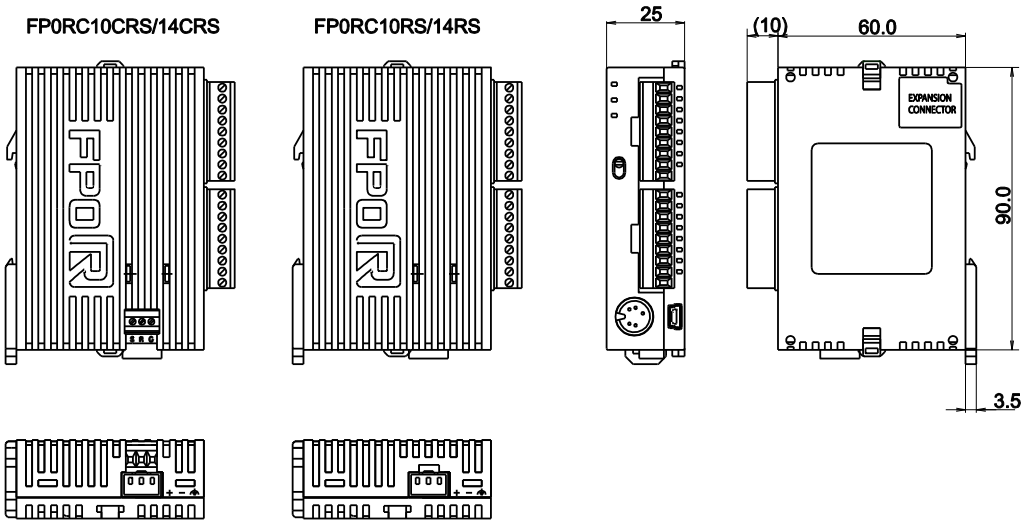
11.2 Dimensions

11.2.1 C10/C14 CPU (terminal block)

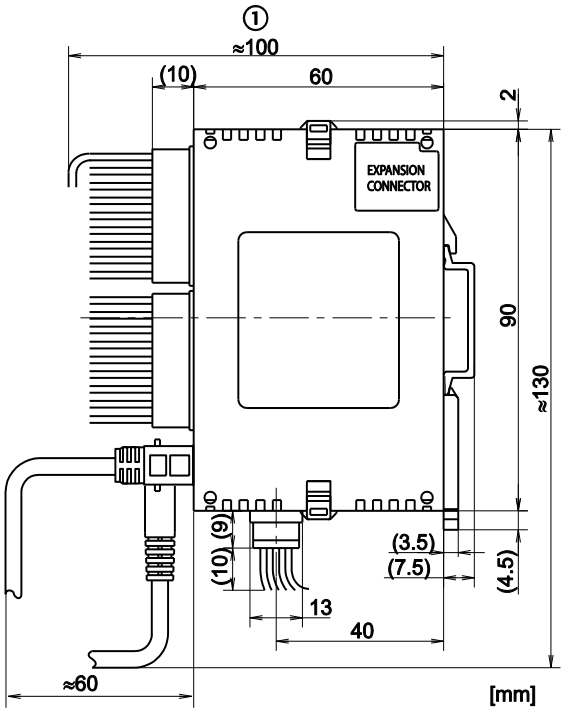
FP0RC10CRS/14CRS, FP0RC10RS/14RS

The same dimensions apply to the following FP0/FP0R expansion units:

- FP0R-E8RS
- FP0R-E16RS.



When mounting terminal block and power supply cable



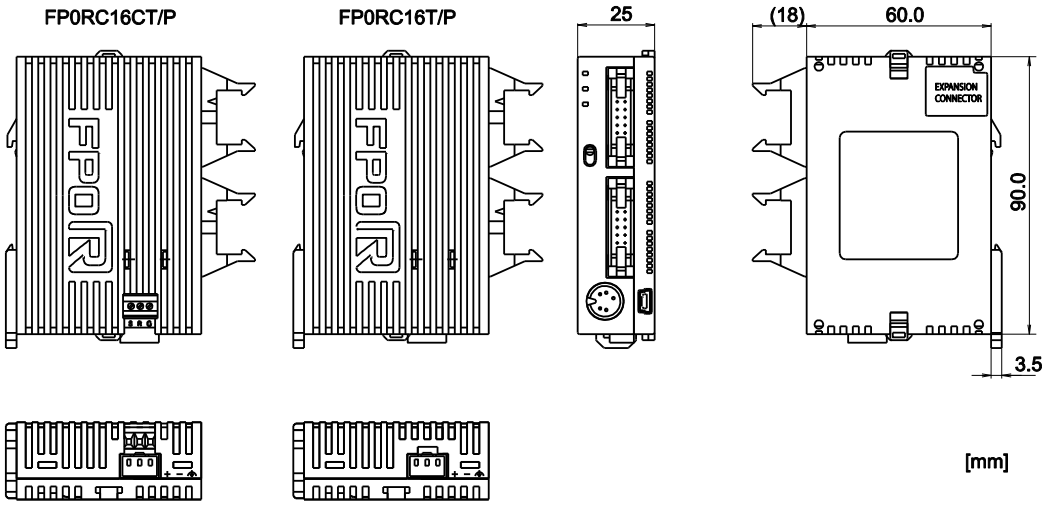
① Maximum installation dimension

11.2.2 C16 CPU (MIL connector)

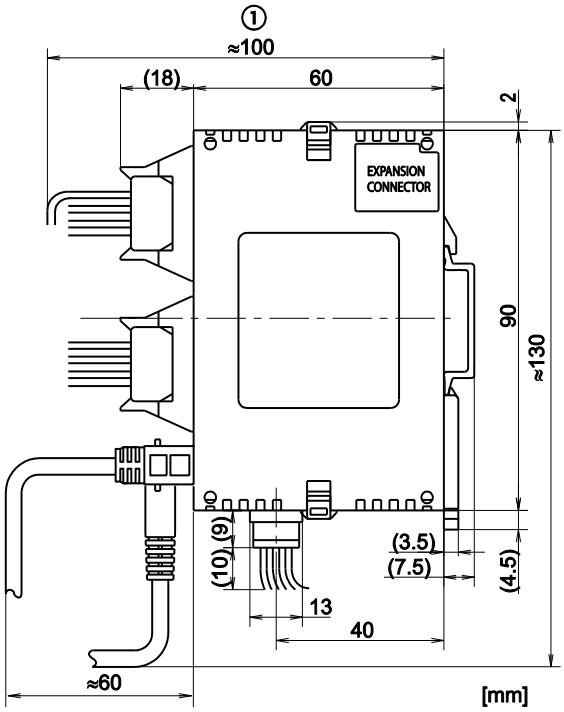
FP0RC16CT/P, FP0RC16T/P

The same dimensions apply to the following FP0/FP0R expansion units:

- FP0R-E32T, FP0R-E32P
- FP0R-E16X, FP0R-E16YT, FP0R-E16YP, FP0R-E16T, FP0R-E16P
- FP0R-E8X, FP0R-E8YT, FP0R-E8YP



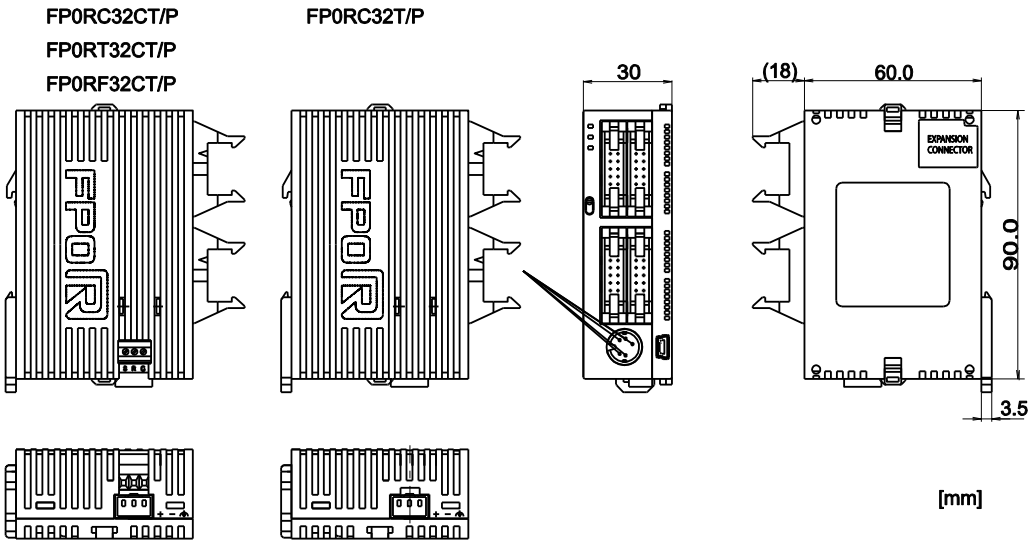
When mounting MIL connector and power supply cable



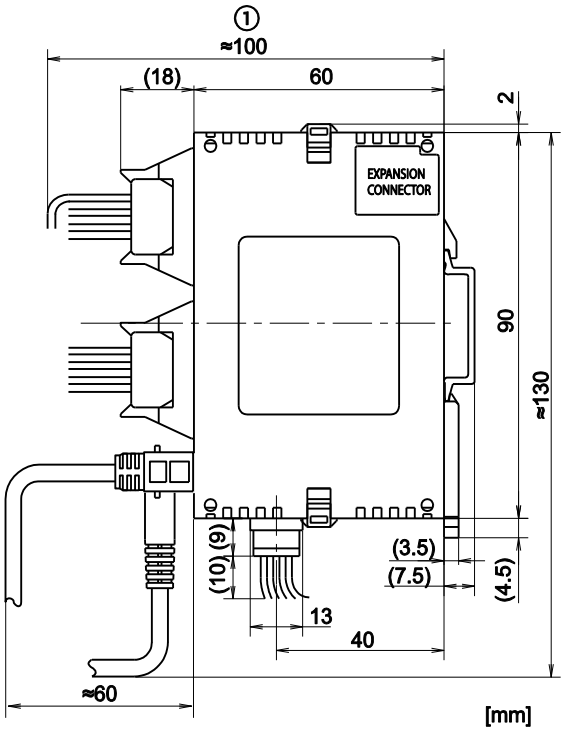
① Maximum installation dimension

11.2.3 C32 CPU (MIL connector)

FP0RC32CT/P, FP0RT32CT/P, FP0RF32CT/P, FP0RT32T/P



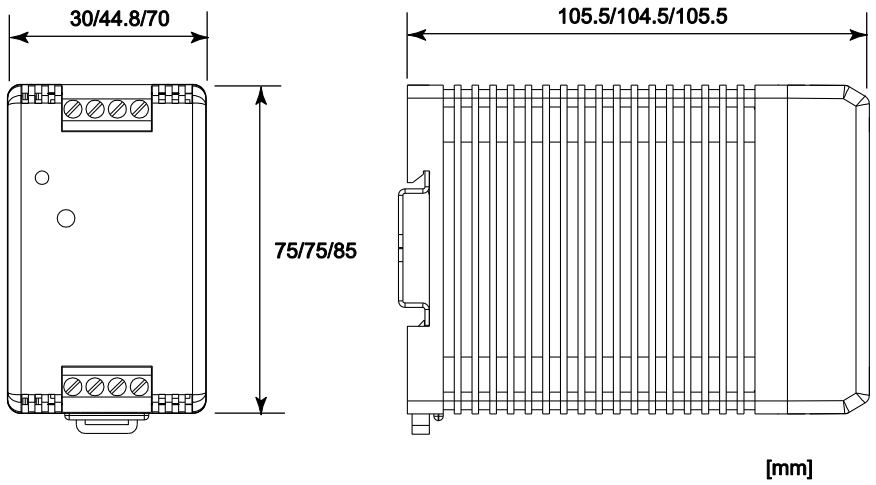
When mounting MIL connector and power supply cable



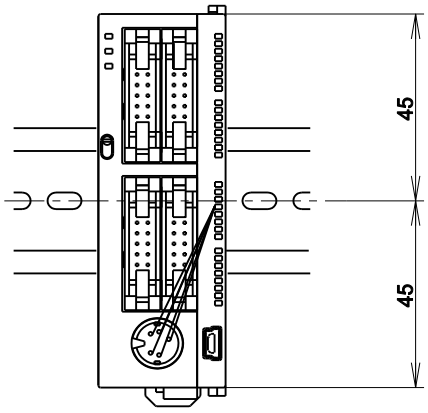
① Maximum installation dimension

11.2.4 Power supply unit

FP-PS24-024E/FP-PS24-060E/FP-PS24-120E



11.2.5 Using DIN rails



11.3 I/O allocation

FP0R CPUs

CPU type		I/Os	I/O addresses
C10	Input	6	X0–X5
	Output	4	Y0–Y3
C14	Input	8	X0–X7
	Output	6	Y0–Y5
C16	Input	8	X0–X7
	Output	8	Y0–Y7
C32/T32/F32	Input	16	X0–XF
	Output	16	Y0–YF

FP0/FP0R expansion units

I/O allocation is performed automatically when an expansion unit is added and is determined by the installation location.

Type of unit		I/Os	Channel	Unit number (installation location)		
				1	2	3
FP0/FP0R I/O expansion unit						
FP0R-E8X	Input	8	–	X20–X27	X40–X47	X60–X67
FP0R-E8R	Input	4	–	X20–X23	X40–X43	X60–X63
	Output	4	–	Y20–Y23	Y40–Y43	Y60–Y63
FP0R-E8YR, E8YT, E8YP	Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
FP0R-E16X	Input	16	–	X20–X2F	X40–X4F	X60–X6F
FP0R-E16R, E16T, E16P	Input	8	–	X20–X27	X40–X47	X60–X67
	Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
FP0R-E16YT, E16YP	Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
FP0R-E32T, E32P, E32RS	Input	16	–	X20–X2F	X40–X4F	X60–X6F
	Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
FP0 analog I/O unit FP0-A21	Input	16	0	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
	Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 A/D conversion unit FP0-A80 and FP0 thermocouple unit FP0-TC4, FP0-TC8	Input	16	0, 2, 4, 6	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1, 3, 5, 7	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
FP0 D/A conversion unit FP0-A04V, FP0-A04I	Input	16	–	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Output	16	0, 2	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
	Output	16	1, 3	WY3 (Y30–Y3F)	WY5 (Y50–Y5F)	WY7 (Y70–Y7F)
FP0 RTD unit FP0-RTD6	Input	16	0, 2, 4	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1, 3, 5	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
	Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 I/O link unit FP0-IOL	Input	32	–	X20–X3F	X40–X5F	X60–X7F
	Output	32	–	Y20–Y3F	Y40–Y5F	Y60–Y7F

Note

The data for each channel of the A/D and D/A conversion units FP0-A80, FP0-TC4/TC8, FP0-A04V/I, and FP0-RTD6 is converted and loaded with a user program that includes a switching flag to convert the data in 16-bit words (see corresponding manuals).

11.4 Flags and memory areas for FP0R

Flags [bits]

Type	Memory size	Available address area		Function
		FP	IEC	
Inputs [1]	1760	X0–X109F	%IX0.0–%IX109.15	Turn on or off based on external input.
Outputs [1]	1760	Y0–Y109F	%QX0.0–%QX109.15	Turn on or off connected devices based on the operation result.
Internal flags [2]	4096	R0–R255F	%MX0.0.0–%MX0.255.15	Used internally by the PLC program to store bit information.
Link flags [2]	2048	L0–L127F	%MX7.0.0–%MX7.127.15	Shared by multiple PLCs connected using PLC link.
Timer flags [2] [3]	1024	T0–T1007/C1008-C1023	%MX1.0–%MX1.1007/%MX2.1008–%MX2.1023	Turn on when the value set with a TM instruction for the timer with the same number has reached 0.
Counter flags [2] [3]	1024	C1008–C1023/T0–T1007	%MX2.1008–%MX2.1023/%MX1.0–%MX1.1007	Turn on when the value set with a CT instruction for the counter with the same number has reached 0.
Special internal flags	224	R9000–R913F	%MX0.900.0–%MX0.913.15	Turn on or off based on specific conditions. Used internally as a flag.

Memory area [words]

Type	Memory size	Available address area		Function
		FP	IEC	
Inputs [1]	110	WX0–WX109	%IW0–%IW109	Code for specifying 16 inputs as one word (16 bits) of data.
Outputs [1]	110	WY0–WY109	%QW0–%QW109	Code for specifying 16 outputs as one word (16 bits) of data.
Internal flags [2]	256	WR0–WR255	%MW0.0–%MW0.255	Code for specifying 16 internal flags as one word (16 bits) of data.

Type		Memory size	Available address area		Function
			FP	IEC	
Link flags		128	WL0–WL127	%MW7.0–%MW7.127	Code for specifying 16 link flags as one word (16 bits) of data.
Data registers [2]	C10, C14, C16	12315	DT0–DT12312	%MW5.0–%MW5.12312	Data memory used in a program. Data is handled in 16-bit units (one word).
	C32, T32, F32	32763	DT0–DT32762	%MW5.0–%MW5.32762	
Link registers [2]		256	LD0–LD255	%MW8.0–%MW8.255	Data memory shared by multiple PLCs connected using PLC link. Data is handled in 16-bit units (one word).
Timer/counter set value area [2]		1024	SV0–SV1023	%MW3.0–%MW3.1023	Data memory for storing the set values of timers or counters. The values are stored by timer/counter number.
Timer/counter elapsed value area [2]		1024	EV0–EV1023	%MW4.0–%MW4.1023	Data memory for storing the elapsed values during operation of timers or counters. The values are stored by timer/counter number.
Special data registers		440	DT90000–DT90439	%MW5.90000–%MW5.90439	Data memory for storing settings and error codes.

Memory area [double words]

Type	Memory size	Available address area		Function
		FP	IEC	
Inputs [1]	55	DWX0–DWX108	%ID0–%ID108	Code for specifying 32 inputs as a double word (32 bits) of data.
Outputs [1]	55	DWY0–DWY108	%QD0–%QD108	Code for specifying 32 outputs as a double word (32 bits) of data.
Internal flags [2]	128	DWR0–DWR254	%MD0.0–%MD0.254	Code for specifying 32 internal flag points as a double word (32 bits) of data.
Link flags	64	DWL0–DWL126	%MD7.0–%MD7.126	Code for specifying 32 link flag points as a double word (32 bits) of data.

Type		Memory size	Available address area		Function
			FP	IEC	
Data registers [2]	C10, C14, C16	6157	DDT0–DDT12311	%MD5.0–%MD5.12311	Data memory used in a program. Data is handled in 32-bit units (double word).
	C32, T32, F32	16382	DDT0–DDT32761	%MD5.0–%MD5.32761	
Link registers [2]		128	DLD0–DLD126	%MD8.0–%MD8.126	Data memory shared by multiple PLCs connected using PLC link. Data is handled in 32-bit units (double word).
Timer/counter set value area [2]		512	DSV0–DSV1022	%MD3.0–%MD3.1022	Data memory for storing the set values of timers or counters. The values are stored by timer/counter number.
Timer/counter elapsed value area [2]		512	DEV0–DEV1022	%MD4.0–%MD4.1022	Data memory for storing the elapsed values during operation of timers or counters. The values are stored by timer/counter number.
Special data registers		220	DDT90000–DDT90438	%MD5.90000–%MD5.90438	Data memory for storing settings and error codes.

[1] The number of points noted above is the number reserved as the calculation memory. The actual number of points available for use is determined by the hardware configuration.

[2] There are hold and non-hold type memory areas. When the power supply turns off or the mode is changed from RUN to PROG mode, hold type areas are stored and non-hold type areas are reset.

C10/C14/C16/C32:
The hold type and non-hold type areas are fixed. For information on the size of each area, refer to the performance specifications.

T32/F32:
The settings of the hold type areas and non-hold type areas can be changed using the system registers.

T32:
If the battery is empty the data values in the hold area will become indefinite during power-off. They are cleared to 0 the next time the power is turned on.
See "Backup and clock/calendar functions" on p. 38.

[3] The number of points for timer and counter flags can be changed using system register 5. The numbers in the table are the default settings.

11.5 System registers

System registers are used to set values (parameters) which determine operation ranges and functions used. Set values based on the use and specifications of your program. There is no need to set system registers for functions which will not be used.

11.5.1 Precautions when setting system registers

System register settings are effective from the time they are set.

However, MEWNET-W0 PLC link settings, input settings, TOOL and COM port communication settings become effective when the mode is changed from PROG to RUN. With regard to the modem connection setting, when the power is turned off and on or when the mode is changed from PROG to RUN, the PLC sends a command to the modem which enables it for reception.

After an initialization with **Online** → **Clear PLC**, all system register values (parameters) set will be reset to their default values.

11.5.2 Types of system registers

Memory size (system register 0)

Size of the memory area for the user program.

Hold on/off (system registers 5–8, 10–14)

Use these system registers to specify the hold area start addresses for flags and registers. Hold areas are not cleared to 0 when the PLC is switched to PROG mode or when the power is turned off.

The memory area for timer flags and counter flags is partitioned using system register no. 5. Specify the start address for the counter flags.

Act on error (system registers 4, 20, 23, 26)

Set the operation mode to be chosen after errors such as an operation error, a battery error, or an I/O verification error.

Time-Out (system registers 30–32, 34)

Set the waiting time before an error is output. You can also specify a constant scan time.

PLC Link (system registers 40–47, 50–55, 57)

These settings are for using link flags and link registers in MEWNET-W0 PLC link communication. Note that PLC Link is not the default setting.

High-Speed Counter, Pulse-Catch Input, Interrupt Input (system registers 400–405)

When using the high-speed counter function, pulse catch function or interrupt function, set the operation mode and the input number to be used for the function.

Time Constants (system registers 430–433)

Set a time constant for the CPU inputs. These time constants can be useful to negate the effects of noise or bouncing, e.g. for a switching device.

TOOL Port, COM Port (system registers 410–421)

Set these registers when the TOOL port and COM ports 1 and 2 ports are to be used for MEWTOCOL-COM Master/Slave connections, program controlled communication, PLC link, and modem communication. Note that the default setting is MEWTOCOL-COM Master/Slave.

11.5.3 Checking and changing system registers

Procedure

Downloading project and system registers

1. Double-click "PLC" in the navigator
2. Double-click "System registers"
3. To change a value, write the new value into the system register table
4. **Online** → **Online mode** or 
5. **Online** → **Download program code and PLC configuration**

   This downloads the project and system registers.

Procedure

Downloading system registers only

1. **Online → PLC configuration**
2. Double-click "System registers"
3. Choose [Download to PLC]

11.5.4 Table of system registers

Memory size

Name	Name	Default	Values
0	Sequence program area size	12/16/32 kwords[1)]	Fixed

[1)] Depending on PLC type (12k, 16k, or 32k type)

Hold on/off [1)]

Name	Name	Default	Values
5	Counter start address	1008	0–1024
6	Timer/Counter hold area start address	1008	Fixed/0–1024 [3)]
7	Internal flag hold area start address (in word units)	248	Fixed/0–256 [3)]
8	Data register hold area start address	12000/ 32450 [2)]	Fixed/0–32763 [3)]
10	Link flag hold area start address for PLC Link 0 (in word units)	64	Fixed/0–64 [3)]
11	Link flag hold area start address for PLC Link 1 (in word units)	128	Fixed/64–128 [3)]
12	Link register hold area start address for PLC Link 0	128	Fixed/0–128 [3)]
13	Link register hold area start address for PLC Link 1	256	Fixed/128–256 [3)]
14	Step ladder hold/non-hold	Non-hold	Fixed or Hold/Non-hold [3)]

[1)] FP0R-T32: If the battery is empty the data values in the hold area will become indefinite during power-off. They are cleared to 0 the next time the power is turned on.

[2)] Depending on PLC type (16k/32k type)

[3)] Depending on PLC type (Fixed for C10, C14, C16, C32, variable for T32, F32)

Act on error

Name	Name	Default	Values
4	DF-, P-function leading/falling edge detection	Holds result	Holds result/disregards result
20	Duplicate output	Enable	Fixed
23	I/O verification error	Stop	Stop/Continue
26	Operation error	Stop	Stop/Continue

Time-Out

Name	Name	Default	Values
30	Watchdog timer time-out	699.1ms	Fixed
31	Multi-frame communication time	6500.0ms	10.0–81900.0ms
32	Timeout value for the communication functions based on F145, F146	10000.0ms	10.0–81900.0ms
34	Constant scan time	0.0ms	0.0–600.0ms 0.0: Normal scan (non-constant)

PLC Link

Name	Name	Default	Values
46	PLC link 0 and 1 allocation setting	Use PLC link 0	Use PLC link 0/Use PLC link 1
47	PLC link 0 - Highest station number in network	16	1–16
40	PLC link 0 - Link flags - Send/receive area - Number of words shared by all linked PLCs	0	0–64 words
42	PLC link 0 - Link flags - Send area - Start sending from this word address	0	0–63
43	PLC link 0 - Link flags - Send area - Number of words to send	0	0–64 words
41	PLC link 0 - Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0–128 words
44	PLC link 0 - Link registers - Send area - Start sending from this word address	0	0–127
45	PLC link 0 - Link registers - Send area - Number of words to send	0	0–127 words
57	PLC link 1 - Highest station number in network	16	1–16
50	PLC link 1 - Link flags - Send/receive area - Number of words shared by all linked PLCs	0	0–64 words
52	PLC link 1 - Link flags - Send area - Start sending from this word address	64	64–127
53	PLC link 1 - Link flags - Send area - Number of words to send	0	0–64 words
51	PLC link 1 - Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0–128 words
54	PLC link 1 - Link registers - Send area - Start sending from this word address	128	128–255
55	PLC link 1 - Link registers - Send area - Number of words to send	0	0–127 words

High-Speed Counter, Pulse-Catch Input, Interrupt Input

Name	Name	Default	Values
400	High-speed counter: Channel 0	Unused	• Two-phase input (X0, X1) • Two-phase input (X0, X1), Reset input (X2) • Incremental input (X0) • Incremental input (X0), Reset input (X2) • Decremental input (X0) • Decremental input (X0), Reset input (X2) • Incremental input (X0), Decremental input (X1) • Incremental input (X0), Decremental input (X1), Reset input (X2) • Counter input (X0), Incremental/decremental control input (X1) • Counter input (X0), Incremental/decremental control input (X1), Reset input (X2)
400	High-speed counter: Channel 1	Unused	• Incremental input (X1) • Incremental input (X1), Reset input (X2) • Decremental input (X1) • Decremental input (X1), Reset input (X2)
400	High-speed counter: Channel 2	Unused	• Two-phase input (X3, X4) • Two-phase input (X3, X4), Reset input (X5) • Incremental input (X3) • Incremental input (X3), Reset input (X5) • Decremental input (X3) • Decremental input (X3), Reset input (X5) • Incremental input (X3), Decremental input (X4) • Incremental input (X3), Decremental input (X4), Reset input (X5) • Counter input (X3), Incremental/decremental control input (X4) • Counter input (X3), Incremental/decremental control input (X4), Reset input X5)
400	High-speed counter: Channel 3	Unused	• Incremental input (X4) • Incremental input (X4), Reset input (X5) • Decremental input (X4) • Decremental input (X4), Reset input (X5)
401	High-speed counter: Channel 4	Unused	• Two-phase input (X6, X7) • Incremental input (X6) • Decremental input (X6) • Incremental input (X6), Decremental input (X7) • Counter input (X6), Incremental/decremental control input (X7)
401	High-speed counter: Channel 5	Unused	• Incremental input (X7) • Decremental input (X7)
402	Pulse output: Channel 0 (transistor types only)	Unused	• Pulse output (Y0, Y1) • Pulse output (Y0, Y1), Home input (X4) • Pulse output (Y0, Y1), Home input (X4), Position control trigger input (X0) • PWM output (Y0)

Name	Name	Default	Values
402	Pulse output: Channel 1 (transistor types only)	Unused	• Pulse output (Y2, Y3) • Pulse output (Y2, Y3), Home input (X5) • Pulse output (Y2, Y3), Home input (X5), Position control trigger input (X1) • PWM output (Y2)
402	Pulse output: Channel 2 (transistor types only)	Unused	• Pulse output (Y4, Y5) • Pulse output (Y4, Y5), Home input (X6) • Pulse output (Y4, Y5), Home input (X6), Position control trigger input (X2) • PWM output (Y4)
402	Pulse output: Channel 3 (transistor types only)	Unused	• Pulse output (Y6, Y7) • Pulse output (Y6, Y7), Home input (X7) • Pulse output (Y6, Y7), Home input (X7), Position control trigger input (X3) • PWM output (Y6)
403	Pulse catch input: X0	Disable	Disable/Enable
403	Pulse catch input: X1	Disable	Disable/Enable
403	Pulse catch input: X2	Disable	Disable/Enable
403	Pulse catch input: X3	Disable	Disable/Enable
403	Pulse catch input: X4	Disable	Disable/Enable
403	Pulse catch input: X5	Disable	Disable/Enable
403	Pulse catch input: X6	Disable	Disable/Enable
403	Pulse catch input: X7	Disable	Disable/Enable
404/405	Interrupt input: X0→Interrupt 0	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X1→Interrupt 1	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X2→Interrupt 2	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X3→Interrupt 3	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X4→Interrupt 4	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X5→Interrupt 5	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X6→Interrupt 6	Unused	Rising edge/Falling edge/Rising and falling edge
404/405	Interrupt input: X7→Interrupt 7	Unused	Rising edge/Falling edge/Rising and falling edge

Note

If the same input has been set as high-speed counter input, pulse catch input or interrupt input, the following order of precedence is effective: High-speed counter → Pulse catch → Interrupt.

If reset input settings overlap for channel 0 and channel 1, the channel 1 setting takes precedence. If reset input settings overlap for channel 2 and channel 3, the channel 3 setting takes precedence.

The input modes two-phase, incremental/decremental, or incremental/decremental control require a second channel. If channel 0, 2, or channel 4 has been set to one of these modes, the settings for channel 1, 3, and 5, respectively, will be invalid.

The settings for pulse catch inputs and interrupt inputs can only be specified in the system registers.

Transistor types (C16 and higher)

Note

CPU outputs which have been specified as pulse output or PWM output cannot be used as normal outputs.

Input numbers X4 to X7 can be used as home input of pulse output channels 0 to 3. When using the home return function, always set the home input. In this case, X4 to X7 cannot be used as high-speed counter inputs.

The output numbers for the deviation counter clear signal, which can be used with the home return function, are fixed for each channel.
For C16: Channel 0 = Y6, channel 1 = Y7
For C32/T32/F32: Channel 0 = Y8, channel 1 = Y9, channel 2 = YA, channel 3 = YB
If used for the deviation counter clear signal, these outputs are not available as pulse outputs.

Time Constants

Name	Name	Default	Values
430	Time constant of input X0	Unused	0.1ms 0.5ms 1.0ms 2.0ms 4.0ms 8.0ms 16.0ms 32.0ms 64.0ms
430	Time constant of input X1		
430	Time constant of input X2		
430	Time constant of input X3		
431	Time constant of input X4		
431	Time constant of input X5		
431	Time constant of input X6		
431	Time constant of input X7		
432[1)]	Time constant of input X8		
432[1)]	Time constant of input X9		
432[1)]	Time constant of input XA		
432[1)]	Time constant of input XB		
433[1)]	Time constant of input XC		
433[1)]	Time constant of input XD		
433[1)]	Time constant of input XE		
433[1)]	Time constant of input XF		

[1)] 32k types only

TOOL Port

Name	Name	Default	Values
412	TOOL port - communication mode	MEWTOCOL-COM Slave	MEWTOCOL-COM Slave/Program controlled
410	TOOL port -station number	1	1–99
415	TOOL port - baud rate	115200 baud	115200/57600/38400/19200/9600/4800/2400 baud
413	TOOL port - sending data length	8 bits	7 bits/8 bits
413	TOOL port -sending parity check	Odd	None/Odd/Even
413	TOOL port - sending stop bit	1 bit	1 bit/2 bits
413	TOOL port - sending start code	No-STX	No-STX/STX
413	TOOL port - sending end code/reception done condition	CR	CR/CR+LF/ETX/None
420	TOOL port- -receive buffer starting address	0	0–12312 (16k type) 0–32762 (32k type)
421	TOOL port - receive buffer capacity	0	0-2048
412	TOOL port - modem connection	Disable	Disable/Enable

COM Port

Name	Name	Default	Values
412	COM port 1 - communication mode	MEWTOCOL-COM Master/Slave	MEWTOCOL-COM Master/Slave/Program controlled/PLC Link/Modbus RTU Master/Slave
410	COM port 1 -station number	1	1–99
415	COM port 1 - baud rate[1)]	9600 baud	115200/57600/38400/19200/9600/4800/2400 baud
413	COM port 1 - sending data length	8 bits	7 bits/8 bits
413	COM port 1 -sending parity check[1)]	Odd	None/Odd/Even
413	COM port 1 - sending stop bit	1 bit	1 bit/2 bits
413	COM port 1 - sending start code[1)]	No-STX	No-STX/STX
413	COM port 1 - sending end code/reception done condition[1)]	CR	CR/CR+LF/ETX/None
416	COM port 1- -receive buffer starting address	0	0–12312 (16k type) 0–32762 (32k type)
417	COM port 1 - receive buffer capacity	0	0-2048
412	COM port 1 - modem connection	Disable	Disable/Enable

[1)] For PLC Link, the communication format and baud rate settings are fixed:
Data length: 8 bits
Parity: Odd
Stop bit: 1 bit
End code: CR
Start code: No STX
Other system register settings will be ignored.

11.6 Error codes

11.6.1 Error codes E1 to E8

Error code	Name of error	Operation status of PLC	Description and steps to take
E1 (see note)	Syntax error	Stops	A program with a syntax error has been written. Change to PROG mode and correct the error.
E2 (see note)	Duplicated output error	Stops	Two or more operation results are output to the same flag. (This error also occurs if the same timer/counter number is being used.) Change to PROG mode and correct the error. This error is also detected during online editing. No changes will be downloaded and operation will continue.
E3	Not paired error	Stops	For instructions which must be used in a pair such as jump (JP and LBL), one instruction is either missing or in an incorrect position. Change to PROG mode and correct the error.
E4 (see note)	Parameter mismatch error	Stops	An instruction has been written which does not agree with system register settings. For example, the timer/counter number setting in a program does not agree with the timer/counter range setting. Change to PROG mode and correct the error.
E5 (see note)	Program area error	Stops	An instruction was written to the wrong program area (main program area or subprogram area). Change to PROG mode and correct the error. This error is also detected during online editing. No changes will be downloaded and operation will continue.
E6 (see note)	Compile memory full error	Stops	The program stored in the PLC is too large to compile in the program memory. Change to PROG mode and correct the error.
E7 (see note)	High-level instruction type error	Stops	In the program, high-level F and P instructions are triggered by the same operation result. (While the execution condition is TRUE, F instructions are executed in every scan. P instructions are executed only once, at the rising edge of the execution condition.) Correct the program so that the high-level instructions executed in every scan and at the rising edge are triggered separately.
E8	High-level instruction operand combination error	Stops	There is an incorrect operand in an instruction which requires a specific combination of operands (for example, the operands must all be of a certain type). Change to PROG mode and correct the error.

Note

In Control FPWIN Pro, these errors are detected by the compiler. Therefore, they are not critical.

11.6.2 Self-Diagnostic error codes

Error code	Name of error		Operation status of PLC	Description and steps to take
E26	User's ROM error		Stops	Probably a hardware problem. Please contact your dealer.
E27	Unit installation error		Stops	The number of installed units exceeds the limit. Turn off the power supply and check the restrictions on unit combinations.
E28	System register error		Stops	Probably an error in the system registers. Check the system register settings.
E30	Interrupt error 0		Stops	Probably a hardware problem. Please contact your dealer.
E31	Interrupt error 1		Stops	An interrupt occurred without an interrupt request. A hardware problem or error due to noise is possible. Turn off the power and check the noise conditions.
E32	Interrupt error 2		Stops	An interrupt occurred without an interrupt request. A hardware problem or error due to noise is possible. Turn off the power and check the noise conditions.
				There is no interrupt program for an interrupt which occurred. Check the number of the interrupt program and change it to agree with the interrupt request.
E34	I/O status error		Stops	A faulty unit is installed. Replace the unit with a new one.
E42	I/O unit verify error		Selectable	The connection condition of an I/O unit has changed compared to that at the time of power-up. Check the error using sys_wVerifyErrorUnit_0_15 and locate the faulty I/O unit. Set the operation status using system register 23 to continue operation.
E45	Operation error		Selectable	Operation has become impossible after a high-level instruction was executed and a calculation error occurred. The causes of operation errors vary depending on the instruction. Set the operation status using system register 23 to continue operation.
E100–E299	Self-diagnostic error set by F148_ERR	E100–E199	Stops	The self-diagnostic error specified by the F148_ERR instruction has occurred. Use **Monitor** → **PLC status** or  to check the error code.
		E200–E299	Continues	

11.6.3 MEWTOCOL-COM error codes

Error code	Name	Description
!21	NACK error	Link system error
!22	WACK error	
!23	Unit no. overlap	
!24	Transmission format error	
!25	Link unit hardware error	
!26	Unit no. setting error	
!27	No support error	
!28	No response error	
!29	Buffer closed error	
!30	Time-out error	
!32	Transmission impossible error	
!33	Communication stop	
!36	No destination error	
!38	Other communication error	
!40	BCC error	A transfer error occurred in the data received.
!41	Format error	A formatting error in the command received was detected.
!42	No support error	A non-supported command was received.
!43	Multiple frames procedure error	A different command was received when processing multiple frames.
!50	Link setting error	A non-existing route number was specified. Verify the route number by designating the transmission station.
!51	Transmission time-out error	Transmission to another device is not possible because the transmission buffer is full.
!52	Transmit disable error	Transmission processing to another device is not possible (link unit runaway, etc.).
!53	Busy error	Processing of command received is not possible because of multiple frame processing or because command being processed is congested.
!60	Parameter error	Content of specified parameter does not exist or cannot be used.
!61	Data error	There was a mistake in the contact, data area, data number designation, size designation, range, or format designation.
!62	Registration over error	Operation was done when number of registrations was exceeded or when there was no registration.
!63	PC mode error	PC command that cannot be processed was executed during RUN mode.
!64	External memory error	An abnormality occurred when loading RAM to ROM/IC memory card. There may be a problem with the ROM or IC memory card. When loading, the specified contents exceeded the capacity. Write error occurs. • ROM or IC memory card is not installed. • ROM or IC memory card does not conform to specifications
!65	Protection error	A program or system register write operation was executed when the protect mode (password setting or DIP switch, etc.) or ROM operation mode was being used.

Error code	Name	Description
!66	Address error	There was an error in the code format of the address data. Also, when exceeded or insufficient address data, there was a mistake in the range designation.
!67	No program error and no data error	Cannot be read because there is no program in the program area or the memory contains an error. Or, reading of non-registered data was attempted.
!68	Rewrite during RUN error	When inputting with programming tool software, editing of an instruction (ED, SUB, RET, INT, IRET, SSTP, and STPE) that cannot perform a rewrite during RUN is being attempted. Nothing is written to the CPU.
!70	SIM over error	Program area was exceeded during a program write process.
!71	Exclusive access control error	A command that cannot be processed was executed at the same time as a command being processed.

11.7 MEWTOCOL-COM communication commands

Command name	Code	Description
Read contact area	RC (RCS) (RCP) (RCC)	Reads the on and off status of contacts. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Write contact area	WC (WCS) (WCP) (WCC)	Turns contacts on and off. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Read data area	RD	Reads the contents of a data area.
Write data area	WD	Writes data to a data area.
Read timer/counter set value area	RS	Reads the value set for a timer/counter.
Write timer/counter set value area	WS	Writes a timer/counter setting value.
Read timer/counter elapsed value area	RK	Reads the timer/counter elapsed value.
Write timer/counter elapsed value area	WK	Writes the timer/counter elapsed value.
Register or Reset contacts monitored	MC	Registers the contact to be monitored.
Register or Reset data monitored	MD	Registers the data to be monitored.
Monitoring start	MG	Monitors a registered contact or data using MD and MC.
Preset contact area (fill command)	SC	Embeds the area of a specified range in a 16-point on and off pattern.
Preset data area (fill command)	SD	Writes the same contents to the data area of a specified range.
Read system register	RR	Reads the contents of a system register.
Write system register	WR	Specifies the contents of a system register.
Read the status of PLC	RT	Reads the specifications of the PLC and error codes if an error occurs.
Remote control	RM	Switches the operation mode of the PLC.
Abort	AB	Aborts communication.

11.8 Data types

In Control FPWIN Pro, variable declarations require a data type. All data types conform to IEC61131-3.

Reference

For details, please refer to the online help of Control FPWIN Pro.

11.8.1 Elementary data types

Keyword	Data type	Range	Reserved memory	Initial value
BOOL	Boolean	0 (FALSE) 1 (TRUE)	1 bit	0
WORD	Bit string of length 16	0–65535	16 bits	0
DWORD	Bit string of length 32	0–4294967295	32 bits	0
INT	Integer	-32768–32,767	16 bits	0
DINT	Double integer	-2147483648– 2147483647	32 bits	0
UINT	Unsigned integer	0–65,535	16 bits	0
UDINT	Unsigned double integer	0–4294967295	32 bits	0
REAL	Real number	-3.402823466*E38– -1.175494351*E-38 0.0 +1.175494351*E-38– +3.402823466*E38	32 bits	0.0
TIME	Duration	T#0s–T#327.67s	16 bits[1)]	T#0s
		T#0s–T#21474836.47s	32 bits[1)]	
DATE_AND_TIME	Date and time	DT#2001-01-01-00:00:00– DT#2099-12-31-23:59:59	32 bits	DT#2001-01-01-00:00:00
DATE	Date	D#2001-01-01–D#2099-12-31	32 bits	D#2001-01-01
TIME_OF_DAY	Time of day	TOD#00:00:00–TOD#23:59:59	32 bits	TOD#00:00:00
STRING	Variable-length character string	1–32767 bytes (ASCII) depending on PLC memory size	2 words for the head + (n+1)/2 words for the characters	''

1) Depending on PLC type

11.8.2 Generic data types

Generic data types are used internally by system functions and system function blocks and cannot be selected in user-defined POUs. Generic data types are identified by the prefix ANY.

Note

Generic data types are not available in user-defined POUs.

Hierarchy of generic data types

			ANY16 (WX, WY)	ANY32 (DWX, DWY)	
ANY		BOOL	INT, UINT, WORD	DINT, UDINT, DWORD, REAL, DATE, TOD, DT	STRING
	ANY_NOT_BOOL		INT, UINT, WORD	DINT, UDINT, DWORD, REAL, DATE, TOD, DT	
	ANY_NUM		INT, UINT	DINT, UDINT, REAL	
	ANY_INT		INT, UINT	DINT, UDINT	
	ANY_BIT	BOOL	WORD	DWORD	
	ANY_BIT_NOT_BOOL		WORD	DWORD	
	ANY_DATE			DATE, TOD, DT	

11.9 Hexadecimal/binary/BCD

Decimal	Hexadecimal	Binary data	BCD data (Binary Coded Decimal)
0	0000	0000 0000 0000 0000	0000 0000 0000 0000
1	0001	0000 0000 0000 0001	0000 0000 0000 0001
2	0002	0000 0000 0000 0010	0000 0000 0000 0010
3	0003	0000 0000 0000 0011	0000 0000 0000 0011
4	0004	0000 0000 0000 0100	0000 0000 0000 0100
5	0005	0000 0000 0000 0101	0000 0000 0000 0101
6	0006	0000 0000 0000 0110	0000 0000 0000 0110
7	0007	0000 0000 0000 0111	0000 0000 0000 0111
8	0008	0000 0000 0000 1000	0000 0000 0000 1000
9	0009	0000 0000 0000 1001	0000 0000 0000 1001
10	000A	0000 0000 0000 1010	0000 0000 0001 0000
11	000B	0000 0000 0000 1011	0000 0000 0001 0001
12	000C	0000 0000 0000 1100	0000 0000 0001 0010
13	000D	0000 0000 0000 1101	0000 0000 0001 0011
14	000E	0000 0000 0000 1110	0000 0000 0001 0100
15	000F	0000 0000 0000 1111	0000 0000 0001 0101
16	0010	0000 0000 0001 0000	0000 0000 0001 0110
17	0011	0000 0000 0001 0001	0000 0000 0001 0111
18	0012	0000 0000 0001 0010	0000 0000 0001 1000
19	0013	0000 0000 0001 0011	0000 0000 0001 1001
20	0014	0000 0000 0001 0100	0000 0000 0010 0000
21	0015	0000 0000 0001 0101	0000 0000 0010 0001
22	0016	0000 0000 0001 0110	0000 0000 0010 0010
23	0017	0000 0000 0001 0111	0000 0000 0010 0011
24	0018	0000 0000 0001 1000	0000 0000 0010 0100
25	0019	0000 0000 0001 1001	0000 0000 0010 0101
26	001A	0000 0000 0001 1010	0000 0000 0010 0110
27	001B	0000 0000 0001 1011	0000 0000 0010 0111
28	001C	0000 0000 0001 1100	0000 0000 0010 1000
29	001D	0000 0000 0001 1101	0000 0000 0010 1001
30	001E	0000 0000 0001 1110	0000 0000 0011 0000
31	001F	0000 0000 0001 1111	0000 0000 0011 0001
.	.	.	.
.	.	.	.
.	.	.	.
63	003F	0000 0000 0011 1111	0000 0000 0110 0011
.	.	.	.
.	.	.	.
.	.	.	.
255	00FF	0000 0000 1111 1111	0000 0010 0101 0101
.	.	.	.
.	.	.	.
.	.	.	.
9999	270F	0010 0111 0000 1111	1001 1001 1001 1001

11.10 ASCII codes

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	ASCII HEX code		b7							
										b6	0	0	0	0	1	1	1	1
										b5	0	0	1	1	0	0	1	1
										b4	0	1	0	1	0	1	0	1
											Most significant digit							
											0	1	2	3	4	5	6	7
				0	0	0	0	Least significant digit	0		NUL	DEL	SP AC E	0	@	P		p
				0	0	0	1		1		SOH	DC1	!	1	A	Q	a	q
				0	0	1	0		2		STX	DC2	"	2	B	R	b	r
				0	0	1	1		3		ETX	DC3	#	3	C	S	c	s
				0	1	0	0		4		EOT	DC4	$	4	D	T	d	t
				0	1	0	1		5		ENQ	NAK	%	5	E	U	e	u
				0	1	1	0		6		ACK	SYN	&	6	F	V	f	v
				0	1	1	1		7		BEL	ETB	'	7	G	W	g	w
				1	0	0	0		8		BS	CAN	(	8	H	X	h	x
				1	0	0	1		9		HT	EM	)	9	I	Y	i	y
				1	0	1	0		A		LF	SUB	*	:	J	Z	j	z
				1	0	1	1		B		VT	ESC	+	;	K	[	k	{
				1	1	0	0		C		FF	FS	,	<	L	\	l	?
				1	1	0	1		D		CR	GS	-	=	M	]	m	}
				1	1	1	0		E		SO	RS	.	>	N	^	n	~
				1	1	1	1		F		SI	US	/	?	O	_	o	DEL

Record of changes

Manual no.	Date	Description of changes
ART1F475E	05/2009	First edition
ACGM0475V1EN	04/2014	First European edition Addition of Control FPWIN Pro examples
ACGM0475V1.1EN	02/2011	Correction of errors
ACGM0475V2EN	01/2012	• Addition of RS485 type CPUs, including RS485 specifications and wiring information • Change of FP0 expansion units to FP0R expansion units • Change of pressure connection tool product no. from AXY5200 to AXY5200FP • Modification of FP0 program compatibility mode description • Modification of CPU input and output specifications • Removal of AFP0811 mounting plate • Addition of Windows 7 support • Addition of input time constants • Modification of data type description
ACGM0475V3EN	10/2014	Additions: • Tool instructions • New communication instructions Modifications: • Table for accessories, link units, power supply unit • Clock/calendar function: added information on SET_RTC instruction; deleted programming example (2.5.2.2) • Layout change Error corrections: • Behavior of hold areas when battery errors occur (2.5.1) • Wiring diagrams (5.5.1, 6.5.5) • Formula for resistance (5.5.1) • Deleted note on bridging CS and RS (6.6.2) • Description of programming examples (6.6.3) • Flags in program controlled communications (6.6.5.2) • Count input modes (7.3.1)



North America | Europe | Asia Pacific | China | Japan

Panasonic Electric Works

Please contact our Global Sales Companies in:

Europe

Headquarters	Panasonic Electric Works Europe AG	Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49 (0) 8024 648-0, Fax +49 (0) 8024 648-111, www.panasonic-electric-works.com
Austria	Panasonic Electric Works Austria GmbH	Josef Madersperger Str. 2, 2362 Biedermannsdorf, Tel. +43 (0) 2236-26846, Fax +43 (0) 2236-46133 www.panasonic-electric-works.at
	Panasonic Industrial Devices Materials Europe GmbH	Ennshafenstraße 30, 4470 Enns, Tel. +43 (0) 7223 883, Fax +43 (0) 7223 88333, www.panasonic-electronic-materials.com
Benelux	Panasonic Electric Works Sales Western Europe B.V.	De Rijn 4, (Postbus 211), 5684 PJ Best, (5680 AE Best), Netherlands, Tel. +31 (0) 499 372727, Fax +31 (0) 499 372185, www.panasonic-electric-works.nl
Czech Republic	Panasonic Electric Works Europe AG	Administrative centre PLATINIUM, Veveří 3163/111, 616 00 Brno, Tel. +420 541 217 001, Fax +420 541 217 101, www.panasonic-electric-works.cz
France	Panasonic Electric Works Sales Western Europe B.V.	Succursale française, 10, rue des petits ruisseaux, 91370 Verrières Le Buisson, Tél. +33 (0) 1 6013 5757, Fax +33 (0) 1 6013 5758, www.panasonic-electric-works.fr
Germany	Panasonic Electric Works Europe AG	Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49 (0) 8024 648-0, Fax +49 (0) 8024 648-111, www.panasonic-electric-works.de
Hungary	Panasonic Electric Works Europe AG	Magyarországi Közvetlen Kereskedelmi Képviselet, 1117 Budapest, Neumann János u. 1., Tel. +36 1 999 89 26 www.panasonic-electric-works.hu
Ireland	Panasonic Electric Works UK Ltd.	Irish Branch Office, Dublin, Tel. +353 (0) 14600969, Fax +353 (0) 14601131, www.panasonic-electric-works.co.uk
Italy	Panasonic Electric Works Italia srl	Via del Commercio 3-5 (Z.I. Ferlina), 37012 Bussolengo (VR), Tel. +39 0456752711, Fax +39 0456700444, www.panasonic-electric-works.it
Nordic Countries	Panasonic Electric Works Europe AG	Filial Nordic, Knarrarnbergatan 15, 164 40 Kista, Sweden, Tel. +46 859476690, Fax +46 859476699, www.panasonic-electric-works.se
	Panasonic Eco Solutions Nordic AB	Jungmansgatan 12, 21119 Malmö, Tel. +46 40 697 7000, Fax +46 40 697 7099, www.panasonic-fire-security.com
Poland	Panasonic Electric Works Polska sp. z o.o	ul. Wołoska 9A, 02-583 Warszawa, Tel. +48 22 338-11-33, Fax +48 22 338-12-00, www.panasonic-electric-works.pl
Spain	Panasonic Electric Works España S.A.	Barajas Park, San Severo 20, 28042 Madrid, Tel. +34 913293875, Fax +34 913292976, www.panasonic-electric-works.es
Switzerland	Panasonic Electric Works Schweiz AG	Grundstrasse 8, 6343 Rotkreuz, Tel. +41 (0) 41 7997050, Fax +41 (0) 41 7997055, www.panasonic-electric-works.ch
United Kingdom	Panasonic Electric Works UK Ltd.	Sunrise Parkway, Linford Wood, Milton Keynes, MK14 6LF, Tel. +44 (0) 1908 231555, Fax +44 (0) 1908 231599, www.panasonic-electric-works.co.uk

North & South America

USA	Panasonic Industrial Devices Sales Company of America	629 Central Avenue, New Providence, N.J. 07974, Tel. 1-908-464-3550, Fax 1-908-464-8513, www.pewa.panasonic.com

Asia Pacific/China/Japan

China	Panasonic Electric Works Sales (China) Co. Ltd.	Level 2, Tower W3, The Towers Oriental Plaza, No. 2, East Chang An Ave., Dong Cheng District, Beijing 100738, Tel. +86-10-5925-5988, Fax +86-10-5925-5973
Hong Kong	Panasonic Industrial Devices Automation Controls Sales (Hong Kong) Co., Ltd.	RM1205-9, 12/F, Tower 2, The Gateway, 25 Canton Road, Tsimshatsui, Kowloon, Hong Kong, Tel. +852-2956-3118, Fax +852-2956-0398
Japan	Panasonic Corporation	1048 Kadoma, Kadoma-shi, Osaka 571-8686, Japan, Tel. +81-6-6908-1050, Fax +81-6-6908-5781, www.panasonic.net
Singapore	Panasonic Industrial Devices Automation Controls Sales Asia Pacific	300 Beach Road, #16-01 The Concourse, Singapore 199555, Tel. +65-6390-3811, Fax +65-6390-3810

Panasonic®

ACGM0475V3EN 10/2014

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP